Sample TCAS Intent Specification

Nancy G. Leveson and Jon Damon Reese
Safeware Engineering Corporation

This sample intent specification for TCAS was produced by Nancy Leveson (Levels 1 and 2) and Jon Reese (Levels 3 and 4). We have built on the RSML specification produced by ourselves, Mats Heimdahl, and Holly Hildreth for the FAA, but we have changed the notation and modeling language and have expanded the specification to include the entire MOPS Volume 1 and 2 (published by the Radio Technical Commission for Aeronautics) along with additional information. This specification is an example of an Intent Specification and should not be used to provide information about any real implementation of TCAS or any current version of the official FAA specification of TCAS.

Preface

The following is an example of an intent specification using TCAS as an example. An intent specification differs from a standard specification primarily in its structure: Hierarchical abstraction is based on *intent* ("why") rather than simply the more usual *what* and *how*. Because each level is mapped to the appropriate parts of the intent levels above and below it, traceability of design rationale and design decisions is provided from high-level system requirements and constraints down to code (or physical form if the function is implemented in hardware) and vice versa.

There are five levels in an intent specification. Each level supports a different type of reasoning about the system and represents a different model of the same system. The model at each level is described in terms of a different set of attributes or language.

The highest level of an intent specification assists system engineers in their reasoning about system-level properties such as goals, constraints, hazards, priorities, and tradeoffs among them. The second level, System Design Principles, allows engineers to reason about the system in terms of the physical principles and laws upon which the design is based. The third or Blackbox Behavior level, enhances reasoning about the logical design of the system as a whole and the interactions between the components as well as the functional state without being distracted by implementation issues. The lowest two levels provide the information necessary to reason about individual component design and implementation issues. The mappings between levels provide the relational information that allows reasoning across the hierarchical levels and tracing from high-level requirements down to implementation and vice versa.

The intent information represents the design rationale upon which the specification is based and thus design rationale is integrated directly into the specification. Each level also contains information about underlying *assumptions* upon which the design and validation is based. Assumptions are especially important in operational safety analyses. When conditions change such that the assumptions are no longer true, then a new safety analysis should be triggered. These assumptions may be included in a safety analysis document (or at least should be), but are not usually traced to the parts of the implementation they affect. Thus the system safety engineer may know that a safety analysis assumption has changed (e.g., the pacemakers are now being used on children rather than the adults for which the device was originally designed and validated), but it is a very difficult and resource-intensive process to figure out what parts of the design used that assumption.

Each of the five intent levels is also organized in terms of the more common part-whole abstractions, i.e., parallel decomposition and refinement. Because the separation of human factors and the design of the human-computer interface from the main system and component design can lead to serious deficiencies in each, we have attempted to integrate both types of specifications at each level and across levels. Interface specifications and specification of important aspects of environmental components are also integrated into the intent specification. Finally, each level of the intent specification includes a specification of the requirements and results of verification and validation activities of the information at that specification level.

The specification as a whole allows a seamless transition from system to component (including software) specifications and the integration of formal and informal aspects of system and software development. Because the structuring is based on what is known about human problem solving, we believe that this type of specification will enhance human processing and use of specifications and will also enhance our ability to engineer for quality and to build evolvable and changeable systems without degrading quality. The structure is designed to facilitate the tracing of system level requirements and constraints into the design and the assurance of various system properties (such as safety) in the initial design and implementation as well as reduce the costs of implementing changes and reanalysis when the system is changed, as it inevitably will be.

In this document, we try to use industry standard terminology where "shall" denotes a requirement, "should" denotes an option, "must" represents a constraint, and "will" denotes an assumption about the environment. Again, we had to guess at some of these in the following because the documents on which we based this intent specification were sometimes ambiguous in this respect. Mappings are indicated by pointers, but an electronic version of this type of specification could use sophisticated hyper-text links including multiple windows to denote these relationships.

The first number of letters of a link tells you where it is located:

Number 1-5: Requirement on Levels 1 to 5
G: Goal (Level 1)
EA: Environmental Assumption (Level 1)

EC: Environment Constraint (Level 1)
OP: Operator behavioral requirement, assumption, or constraint (Level 1)
L: Limitation (Level 1)
C: Non-safety-related design constraint (Level 1)
SC: Safety-relatd design constraint (Level 1)
FTA-x: Line x of the Fault Tree Analysis

Caveats

This specification is only an example. It does not correspond to the current specification of TCAS and should not be used to answer questions about any TCAS specification or implementation. It is based on a specification we did of an earlier version of TCAS than the current operational version and is incomplete in some sections where we either could not get the necessary information or felt it was too tedious to enter it all when an example of the type of information that would be included was sufficient to illustrate the point. The information concerning human factors is most incomplete due to our inability to find the necessary information. We tried to maintain consistency with the MOPS (DO-185) but we added information in the Level 1 specification of the safety-related design constraints in order to provide a more complete example of this critical type of information.

In addition, we generated the mappings between levels ourselves, and they have not been reviewed by any TCAS experts (nor are they complete). They should also be taken as examples and not as accurate reflections of the original designer's intent: The limitations in our understanding of TCAS required many assumptions on our part that may be unjustified. This is, of course, exactly why it is important to record intent from the beginning.

Contents

Level 1

System-Level Goals, Requirements, Constraints

... intent spe... (1) ... the ...
... (2) ... (3) ... fimitations o...
... (4) ... (5) ... ization ...
... system ... qualities su... system ...

Introduction

TCAS is an airborne collision avoidance system. It interrogates air traffic control transponders on aircraft in its vicinity and listens for the transponder replies. By analyzing these replies with respect to slant range and relative altitude, TCAS determines which aircraft represent potential collision threats and provides appropriate display indications (or advisories) to the flight crew to assure separation.

An aircraft is declared to be a collision threat to the TCAS aircraft if its current position, or its projected position, simultaneously violate range and relative altitude criteria. Generally, an aircraft will be declared to be a collision threat 20-30 seconds before closest approach, providing time for an escape maneuver by the pilot. Two classes of advisories can be provided: Resolution Advisories (RAs) and Traffic Advisories (TAs).

Resolution advisories (RAs) indicate vertical maneuvers that are predicted to increase vertical separation from threatening aircraft. RAs are chosen to provide a specific margin of separation with a minimum change in the existing flight path of the TCAS II aircraft.

Two types of resolution advisories are issued. A *preventive resolution advisory* requires no immediate action but warns the crew not to climb, descend, or adjust vertical speed due to nearby traffic. A *corrective resolution advisory* directs the pilot to alter the vertical speed of the aircraft to ensure safe separation from nearby traffic in the vertical plane.

Traffic advisories (TAs) indicate the positions of intruding aircraft that may later cause resolution advisories to be displayed. In potential collision encounters, the system is designed to ensure that the TA normally occurs approximately 15 seconds before the RA. For intruders not equipped with altitude-reporting equipment, it is not possible to determine if the aircraft is a potential collision threat. For these aircraft, TCAS may therefore generate TAs but will not generate RAs.

Besides the RAs and TAs representing threats or potential threats, TCAS may also provide traffic advisories that represent proximity warnings. These warnings indicate the positions of aircraft that have entered the protected airspace around our aircraft but that do not currently represent a threat or a potential threat.

TCAS is meant to provide back-up separation assurance for the existing conventional air traffic control system without producing unwanted alarms in encounters for which the collision risk does not warrant escape maneuvers. The operation of TCAS is not dependent on any ground-based systems.

TCAS is not designed to supplant either air traffic control's system of separation assurance or the time-honored "see and avoid" system of traffic separation. Rather, it provides an independent backup to these systems, since it makes use solely of airborne transponders as its data source.

Historical Perspective

The development of an effective airborne collision avoidance system has been the goal of the aviation community for many years. Spurred by the collision of two airliners over the Grand Canyon in 1956, the airlines initiated a study of collision avoidance concepts.

During the late 1950s and early 1960s, efforts included an emphasis on passive and noncooperating systems. These proved to be impractical. One reason was the need for nonconflicting, complementary avoidance maneuvers, which requires high-integrity communications between the aircraft in conflict. Another reason was the need to acquire more precise altitude information from intruding aircraft.

One of the most important developments was the derivation of the range rate (τ) concept, the first accurate study of the basic physics of collision avoidance, by Dr. John S. Morrell of Bendix. This concept is based on time-to-go, rather than on distance-to-go, to the closest point of approach.

During the late 1960s and early 1970s, aircraft collision avoidance systems (ACAS) were developed by several manufacturers based on interrogator transponder techniques. Although these systems functioned properly during staged aircraft encounter testing, the FAA and the airlines jointly concluded that in normal airline operations, they would generate a high rate of unnecessary alarms in dense terminal areas. This would undermine their credibility with flight crews. In addition, each target aircraft would have to be similarly equipped in order for an ACAS aircraft to receive protection.

In the mid-1970s, the Beacon Collision Avoidance System (BCAS), which used reply data from Air Traffic Control Radar Beacon System (ATCRBS) transponders to determine intruder range and altitude, was developed. ATCRBS transponders were already installed in all airline, military, and most general aviation aircraft. Thus any BCAS-equipped aircraft would be protected against the majority of other aircraft, without imposing additional equipment requirements for those aircraft. In addition, the discrete address communications techniques used in the Mode S transponders being developed permitted two conflicting BCAS aircraft to achieve coordinated escape maneuvers with a high degree of reliability and allowed a collision avoidance capability on the flightdeck that is independent of the ground system,

In 1981, the FAA made the decision to develop and implement TCAS, using the BCAS design but providing additional capabilities. After FAA and industry sponsored studies, simulations, and flight tests (see *lgl esMin* *idtion* in Level 2 on Page 111), the Radio Technical Commission for Aeronautics (RTCA) completed the initial version of the TCAS II MOPS, labeled DO-185, in September 1984. The MOPS provides specific guidance for avionics manufacturers in designing and testing TCAS II equipment. The MOPS has had extensive changes since its first version was issued. A consolidated version with all the changes included was issued September 6, 1990. The information in this sample document (intent specification) is based on and includes (or is intended to include) all of the information in the 1990 consolidated MOPS.

The FAA has issued Technical Standard Order C-119 and Advisory Circular 20-131 for use by FAA airworthiness authorities to certify the installation of TCAS II in aircraft. Advisory Circular 120-TCAS addresses TCAS II operational requirements. Because TCAS II will also be used by foreign carriers operating inU S. airspace, the International Civil Aviation Organization (ICAO) has prepared Standards and Recommended Practices (SARPS) that are the worldwide standard governing TCAS II design and operation. Finally, the Airlines Electronic 735, wEngineering Committee (AEEC) has completed ARINC Characteristic standardizes the form and function of TCAS II units installed in airline aircraft.

Based on a Congressional mandate (Public Law 100-223), the FAA issued a , 9 rule effective February 1989 that required the equipage of TCAS II on airline aircraft with more than 30 seats by December 30, 1991. Although not in the law, the FAA rule required airline aircraft with 10 to 30 passenger seats to have TCAS , 9 I installed by February 1995. The installation of TCAS III is permitted by the rule but is not required. Public Law 100-236, which was signed on December 15, 1989, amended PL 100-2223 to (1) permit the FAA administrator to extend the deadline for TCAS II fleetwide implementation to December 30, 1993, and (2) require the FAA to conduct a 1-year operational evaluation of TCAS II beginning no later than December 30, 1990. The installation requirement is now in effect.

Environment

[illegible]

TCAS can be combined with other aircraft systems and multiple antennas, displays, and controls can be used. However, any such combination must not degrade the general performance requirements of TCAS equipment or of the equipment with which TCAS is combined.

Figure 1.1 shows the links between TCAS and other components of the air space system related to the achievement of the general collision avoidance goal. See
[illegible] ARINC Characteristic [illegible] (TCAS)
for detailed information about interfaces with other aircraft equipment.

Control Panel

Any needed manual control for the TCAS unit should be provided on the control panel of its associated Mode S Transponder unit. As there is no direct link between this control panel and the TCAS unit, control information will be delivered to the TCAS unit via the transponder. The control panel should provide the needed interface to the Mode S transponder required for the TCAS system. Communications links from the Mode S transponder unit to the TCAS unit should
9'Mark 33 be the 2-wire serial digital system described in ARINC Specification 42
Digital Information Transfer System (DITS)". Details of the Mode S control panel
7 are contained in ARIS Characteristic 18.

Antennas

The installation should provide two L-band transmit-receive antennas for each TCAS on the aircraft, one located on the underside of the fuselage and the other on top. The upper antenna will provide the TCAS computer unit with signals

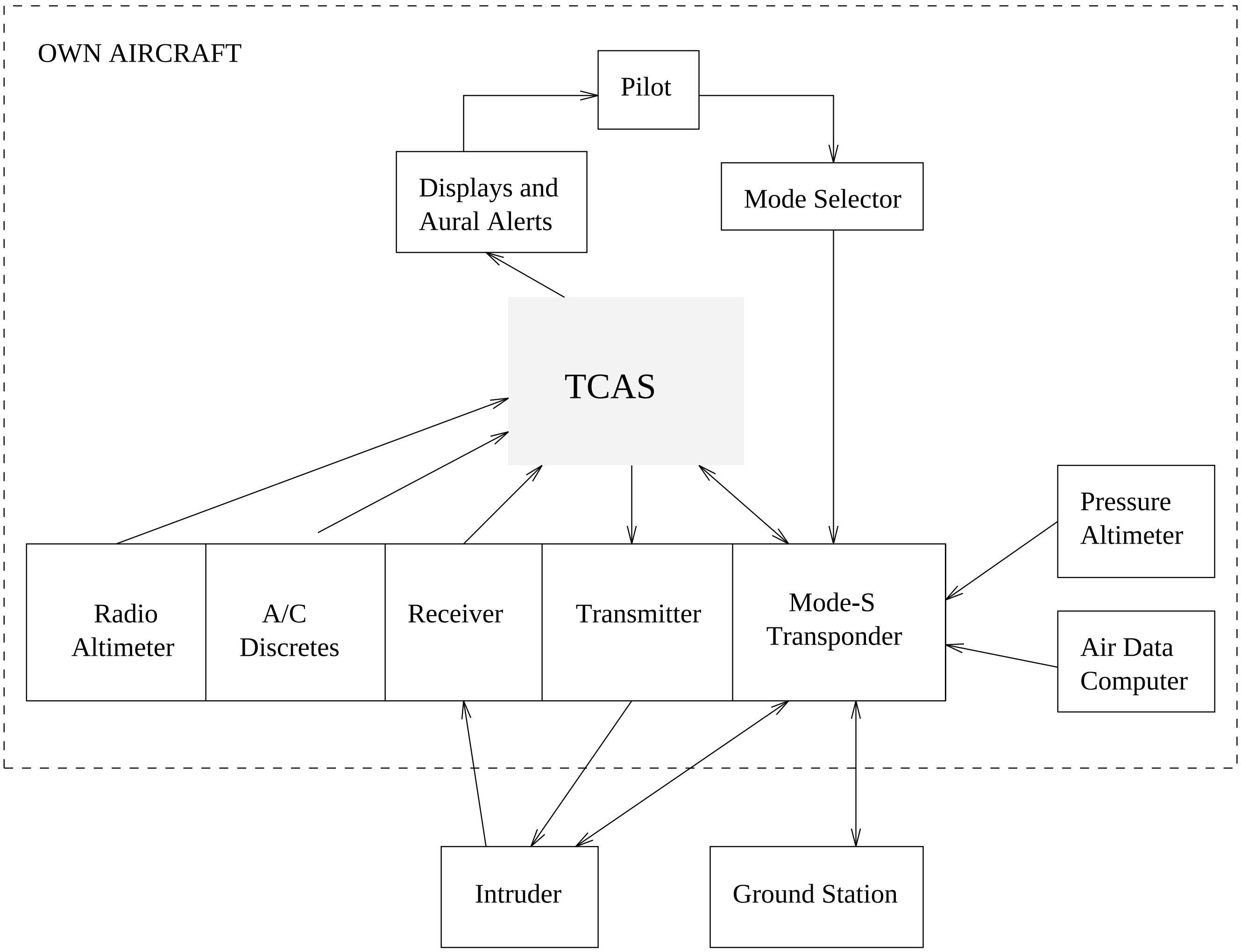


Figure 1.1: TCAS and its Environment

for an estimation of the signal angle-of-arrival. The lower antenna may be an omnidirectional blade antenna or directional antenna at the option of the user.

Altitude Reporting Data

Barometric altitude data is not provided directly to the TCAS unit. Instead, altitude data will be provided to the TCAS unit through the Mode S Transponder from appropriate aircraft computing devices, such as Air Data System, Altitude Computer System, or other parts of the aircraft altimeter system.

The pressure altitude data should be obtained from the most accurate source available in the aircraft and shall correspond to that being transmitted by the associated Mode S transponder. The accuracy of the altitude data shall be at least that specified in Appendix A of RTCA DO-185. It shall be shown that the resolution of the altimetry source is compatible with TCAS. The altitude source with the finest compatible resolution should be used. When available, the resolution should be in increments of 10 feet or less. Information should also be provided to indicate when the pressure altitude information is invalid (→FTA-340, FTA-870, 1.23.1, ↓2.68).

Radio altitude information is provided to TCAS to inhibit advisories at some low levels of flight (↓2.42), to allow automatic sensitivity level selection when close to the ground (↓2.3, 2.42), and to determine that individual targets are on the ground (↓2.19, 2.42). Information shall also be provided to indicate when the radio altitude information is invalid (→FTA-330, FTA-340, FTA-870, 1.23.3, ↓2.68).

Aircraft Discretes

Discrete information from aircraft configuration sensors such as flaps, slats, landing gear, etc. should be provided to ensure that TCAS appropriately inhibits climb advisories to the airplane performance limits (↓2.38, 2.39, 2.40, 2.41, 2.41).

Mode S Transponder

The TCAS equipment will operate in conjunction with and in close proximity to a Mode S transponder. This imposes design requirements on the Mode S transponder that are not covered in RTCA DO-181 (Minimum Operational Performance Standards for the Mode S Beacon System, March 1983) so that TCAS requirements specified in this document can be met. Similarly,

[1.1]

The functions of TCAS equipment shall not in any way degrade the perfor-
DO-mance of the Mode S transponder beyond that specified in RTCA 181.

It is the responsibility of the manufacturers to identify all areas where incompatibilities may arise and to design and construct their equipment so that the requirements of DO-181 are met when both the Mode S Transponder and TCAS are operating.

Aircraft Identification

Discrete information shall be provided to the Mode S transponder for the unique aircraft Mode S identification code.

Attitude

Pitch and roll attitude information may be provided to assist with stabilization of the directional antenna function to assure surveillance and TA display data remain unaffected by aircraft normal maneuvers. If attitude information is used by TCAS, the information shall also be provided to indicate when the attitude →data are invalid (1.23.1, 2. 68).

Heading

Aircraft heading information may be provided for use in surveillance reference and for the TA display reference presentation. If heading information is provided, information shall also be provided to indicate when the heading data are invalid (→1.23.1).

Relation to Other Alerts and Warnings on the Aircraft

TCAS is an "Environmental" System, similar to GPWS and the Windshear Alerting System. For this reason and the anticipated high alert rates for TAs and RAs compared to any other alert on the aircraft, the TCAS alerts and advisories should remain independent of those using the master caution and warning system. This does not preclude other methods of annunciation for failures of the TCAS equipment that are consistent with flight deck specific design philosophy.

[1.2]

Among the environmental alerts, the hierarchy shall be: Windshear has first priority, then GPWS, then TCAS.

[1.3]
During the period of time when TCAS is inhibited by either the windshear system or GPWS, TCAS shall revert to the TA-Only mode and the aural alerts shall be inhibited
.[1]

Assumptions about the Environment

The correct operation of TCAS as well as the hazard analyses are based on some assumptions about the environment in which TCAS operates:

[EA.1]
High-integrity communications exist between aircraft (i.e., transponders).

[EA.2]
The TCAS-equipped aircraft carries a Mode-S air traffic control transponder, whose replies include encoded altitude when appropriately interrogated.

[EA.3]
The other aircraft have operating transponders.

[EA.4]
All aircraft have legal identification numbers.

[EA.5]
Altitude information is available from intruding targets with a minimum precision of 100 feet.

[EA.6]
The altimetry system that provides own pressure altitude to the TCAS equipment will satisfy the requirements in
...

[EA.7]
Threat aircraft will not make an abrupt maneuver that thwarts the TCAS escape maneuver.

Environment Constraints

[EC.1]
The behavior or interaction of non-TCAS equipment with TCAS must not degrade the performance of the TCAS equipment.

---

[1] Although [illegible] req [illegible], [illegible] seems imp [illegible] e

[EC.2]
The interaction with TCAS must not degrade the performance of the equipment with which TCAS interacts.

Op[illegible]

[illegible] assumptions [illegible] ments [illegible] is use [illegible] TCAS l [illegible] [illegible] an

[OP.1]
Compliance with a TCAS maneuver is expected unless, in the judgement of the pilot-in-command, doing so would present a great hazard to the safety of the flight (FTA-530, 2. 72.4).

[OP.2]
While ultimate responsibility for safety of flight rests with the pilot in command, the need to follow TCAS advisories shall be emphasized in TCAS training (FTA-5 10, 5.training).

[OP.3]
TCAS advisories shall be executed in such a way as to minimize the aircraft's deviation from its ATC clearance (3.3).

[OP.4]
After the threat has been resolved, the pilot shall return promptly and smoothly to his/her previously assigned flight path (FTA-5 60, 3.3).

[OP.5]
Special procedures may be necessary for using TCAS during airport approach (2. 72.2).

[OP.6]
Controls should be set to inhibit resolution advisories when operating in intentional close proximity to other aircraft (72. 1.5).

[OP.7]
TCAS must not be used as a substitute for normal crew vigilance outside the cockpit (→FTA- 143, ↓1.3).

[OP.8]
Pilots must learn to use the system the way the designers and its logic intend it to be used, although they also must remember to use their training and experience to evaluate situations and take appropriate action to ensure safety of flight (↓4.4 AEM, ↓5. training).

> **Assumption:** It is assumed that pilots will use all of the information available to them (not just TCAS-provided information) to ensure safety of flight.

[OP.9]
The pilot must not maneuver on the basis of the TA only (→FTA- 1045, ↓2. 1.3).

[OP.10]
Pilots must not delay execution of a resolution advisory, stop before the RA is removed, or continue beyond the point the RA is removed (↓2. 1.1 →FTA-210, FTA-550, FTA-555, FTA-560, FTA-565, FTA-865).

[OP.11]
TCAS evasive maneuvers must be complied with in a timely and gentle fashion (↓3.3).

[OP.12]
TCAS must be capable of being operated by a single person (↓Page 63).

[OP.13]
Each pilot must be able to read and operate TCAS independently from the other crew members (↓Page 63).

Human Interface Requirements

Aural Alerts (→FTA-520,72).

[1.4]
Aural alerts shall be presented by voice announcements only. These announcements are of a prescribed duration.

[1.5]
Aural alerts shall be announced in a high-fidelity distinguishable voice. Automatic volume adjustment for ambient conditions is desired.

[1.6]
RA messages consisting of a single word shall be spoken three times; longer messages shall be spoken twice. If a logic change occurs before the aural alert is complete and a new alert is warranted, the original alert shall be terminated immediately and the new alert started.

[1.7]
Corrective RA messages that indicate that a previously announced corrective RA must be increased in strength or reversed shall be spoken with a sense of urgency.

Visual Alerts (↓2.81)

[1.8]
A red visual alert shall be provided in the primary field of view for each pilot for resolution advisories (↓2.80).

[1.9]
If a written message is shown on the display, it should flash or otherwise be highlighted. It must be consistent with the aural RA annunciation. Whether

or not this is done could depend upon how conspicuous other RA information is on display.

[1.10]
The written message “TCAS” may be used to indicate the source of the alert.

[1.1]1
The RA display shall be located in the primary field of view for each pilot.

[1.12]
An autopilot and autothrottle interface may be used provided the autopilot performance is consistent with TCAS performance requirements, and if its activation is pilot selectable.

[1.13]
A switch position shall be provided to inhibit RAs for parallel approach operations or other flight in intentional close proximity to other aircraft (6, ↓2.274,)2. ←OP.

[1.14]
Presentation of RA commands shall be independent of whether or not the aircraft is in a turn.

Display (↓Page 16)

Collision avoidance maneuver advisories and traffic advisories may be displayed to the cockpit crew on one or more dedicated displays, on displays integrated with other instruments such as instantaneous Vertical Speed Indicator (IVSI), or on a CRT flight instrument.

Traffic displays may take several forms: Independent, stand-alone, integrated and time-shared with digital color radar, integrated with the flightcrew’s Instantaneous Vertical Speed Indicators (IVSI), or integrated with other displays such as Electronic Horizontal Situation Indicators (EHSI), navigation, or other multifunction displays.

[1.15]
If the traffic display uses a multi-function display that is shared with other services such as ACARS, the traffic display function shall be immediately available for display by a single selection accessible to both pilots.

Safety-related display design constraints can be found on Page 36.

Controls (↓ 7.4)

[1.16]
A means to select the following modes of operation shall be provided.

1. Operation of the Mode S transponder only.
2. Operation of the Mode S transponder and TCAS II in the TA/RA mode simultaneously.
3. Operation of the ATCRBS transponder only, if installed. It must not be possible to operate the TCAS II and ATCRBS transponder or the Mode S and ATCRBS transponder simultaneously.
4. Operation of TCAS II in the TA mode and Mode S transponder simultaneously.
5. Operation of TCAS II in the standby mode.
6. A means to select the assigned ATCRBS code.
7. A means to initiate the transponder IDENT function.
8. A means to initiate the TCAS II self-test (flight deck location optional).
9. A means to suppress transponder altitude reporting.

[1.17]
The following optional controls may be provided:

1. Selection of weather radar only.
2. Control to select the display of traffic within selected altitude bands.
3. Selection of the weather radar and traffic display simultaneously.
4. Selection of relative or actual/flight-level altitude of traffic.

Safety-related control design constraints can be found on Page 35.

TCAS System Functional Goals

To provide a collision avoidance capability that can be carried by aircraft that require the highest level of protection from midair collision (for a broad spectrum of aircraft types) and that can operate independently of the ground-based system while coordinating with it if necessary.

[G.1]
Provide affordable and compatible collision avoidance system options for a broad spectrum of National Airspace System users.

[G.2]
Detect potential midair collisions with other aircraft in all meteorological conditions.

[G.3]
Provide protection throughout navigable airspace, including airspace not covered by ATC primary or secondary radar systems.

[G.4]
Provide collision protection in the absence of ground equipment.

[G.5]
Coordinate with ATC ground stations, to the extent necessary, where such ground stations are provided.

High-Level Functional Requirements

[1.18]
TCAS shall provide collision avoidance protection for any two aircraft closing horizontally at any rate up to 1200 knots and vertically up to 10,000 feet per minute.

> **Assumption:** This requirement is derived from the assumption that commercial aircraft can operate up to 600 knots and 5000 fpm during vertical climb or controlled descent (and therefore two planes can close horizontally up to 1200 knots and vertically up to 10,000 fpm).

[1.19]
TCAS shall handle encounters involving multiple aircraft in areas with large numbers of aircraft within a selected range (without saturation of the operating frequencies).

> [1.19.1]
> TCAS shall operate in en-route and terminal areas with traffic densities up to 0.3 aircraft per square nautical miles (nmi) (i.e., 24 aircraft within 5 nmi) (13, Page 2 02).
>
> > **Assumption:** Traffic density may increase to this level by 1990, and this will be the maximum density over the next 20 years.
>
> [1.19.2]
> TCAS shall operate out to 14 nautical miles (Page 188).

[1.20]
TCAS shall provide timely advisories to the pilot:

[1.20.1]
Resolution advisories shall indicate vertical maneuvers that are predicted to increase vertical separation from threatening aircraft. Depending on the selected protection range, the nominal advisory time before closest approach shall vary from 20 to 30 seconds (2.3).

[1.20.2]
Traffic advisories shall indicate the positions of intruding aircraft that may up to 15 seconds later cause resolution advisories to be displayed (2. 23.1.1).

Assumption: Traffic advisories assist the flight crew of the TCAS aircraft in the following two ways:

a. Alert the crew to the presence of potentially threatening traffic and provide intruder position data that can aid visual acquisition.
b. Provide the crew with a graphic depiction of the conflict situation prior to the time the resolution advisory is displayed, thereby facilitating a reduction in the time taken by a pilot to respond to the resolution advisory.

[1.20.2.1]
Traffic advisories shall be provided on Mode A (no altitude data available), Mode C, and Mode S transponder equipped aircraft.

[1.20.2.2]
Traffic advisories shall indicate the range, range rate, altitude, altitude rate, and bearing of the intruder aircraft relative to own aircraft.

[1.20.2.3]
Traffic advisories without altitude (proximity advisories) shall be provided on non-altitude reporting, transponder-equipped aircraft that describe their ranges and bearing.

[1.20.3]
Proximity advisories shall indicate the positions of other aircraft that are within the protected airspace around own aircraft and one or more of the following conditions holds: (1) the other aircraft does not represent a collision threat with own aircraft, (2) there is insufficient data to determine whether the aircraft represents a threat, or (3) this version of TCAS does not evaluate collision threats (see 1.22.1).

[1.21]
TCAS shall provide direction-finding capability to support the display of traffic.

> **Assumption:** TCAS is intended for operation in low to medium transponder densities (see 1.19.1).

[1.21.1]
The surveillance equipment shall be omnidirectional (Page 61).

[1.21.2]
TCAS shall be capable of resolving encounters with aircraft equipped with Mode S transponders as well as with aircraft equipped with the internationally standardized ATCRBS (Air Traffic Control Radar Beacon System) altitude reporting transponders.

> **Assumption:** Although the population of transponders will slowly change from ATCRBS to Mode S as TCAS installations are made, some ATCRBS transponders will be a part of the environment throughout the life of the TCAS equipment. Sixteen percent of the transponder-equipped aircraft in U.S. airspace by 1990 will be Mode S equipped.

[1.22]
TCAS shall support three options or levels of service.

[1.22.1]
TCAS I shall provide proximity warning only.[2]

> **Assumption:** TCAS I is intended for use by smaller commuter aircraft and by general aviation aircraft.

[1.22.2]
TCAS II shall provide traffic advisories and resolution advisories (recommended escape maneuvers) in a vertical direction to avoid conflicting traffic.

> **Assumption:** TCAS II will be used by airline aircraft and larger commuter and business aircraft.

[1.22.3]
TCAS III shall provide traffic advisories and resolution advisories in the horizontal as well as the vertical direction to avoid conflicting traffic.

---

[2] [illegible] 4 (p. 19).

Note: TCAS III has not been implemented yet.

[1.22.4]
The advisories provided will depend on what TCAS equipment the target aircraft is carrying, as specified in the following table. (TA = traffic advisory, VRA = vertical resolution advisory, HRA = horizontal resolution advisory, TTC = TCAS coordination)

Table 1. TCAS Levels of Protection

Target Aircraft Equipment	Own Aircraft Equipment	Advisories
Mode A only	TCAS I	TA
	TCAS II	TA
	TCAS III	TA
Mode C or Mode S with altitude encoding	TCAS I	TA
	TCAS II	TA, VRA
	TCAS III	TA, VRA, HRA
TCAS I, II, or III	TCAS I	TA
TCAS I	TCAS II	TA, VRA
	TCAS III	TA, VRA, HRA
TCAS II or III	TCAS II	TA, VRA, TTC
	TCAS III	TA, VRA, HRA, TTC

[1.23]
TCAS shall have performance monitoring capability.

[1.23.1]
TCAS shall inform the pilot about the operational status of TCAS (i.e., whether it is operating properly) and when resolution advisories are not possible due to failure of the TCAS equipment or any of its sensors or displays. The pilot shall be able to validate the performance of the TCAS computer (↓72. 0).

[1.23.2]
The monitor shall declare a TCAS failure in the event that own Mode S address is all 0's or 1's.

[1.23.3]
The monitor shall declare a TCAS failure after 10 seconds in the event that, at the minimum, valid radio altitude input is not present (←Page 92, ↓ 68).

[1.23.4]
When a failure is detected, the monitor shall:

[1.23.4.1]
Indicate to the flight crew that an abnormal condition exists.

[1.23.4.2]
Discontinue TCAS operation. [*Ho* [3] *...*]

[1.23.4.3]
Cause any Mode S transmissions that report own aircraft status to show that own aircraft has no on-board resolution capability.

[1.23.4.4]
Prevent interrogations by own aircraft TCAS.

[1.23.4.5]
Deactivate the normal TCAS display functions.

[1.23.5]
The monitor functions shall be accomplished by positive means within one second.

---

[3] [illegible] ...ecifica ... uded ...ecifica ... TRM ana [illegible]

System Limitations

[illegible] limitations ma [illegible] limitations ma [illegible] AFM [illegible] the AFM. [illegible]

[L.1]
TCAS provides no protection against aircraft that do not have an operating transponder.

[L.2]
TCAS is dependent on the accuracy of the threat aircraft's reported altitude. Separation assurance may be degraded by errors in intruder pressure altitude as reported by the transponder of the intruder aircraft.

> **Assumption:** This limitation holds for existing airspace, where many aircraft use pressure altimeters rather than GPS. As more aircraft install GPS systems with greater accuracy than current pressure altimeters, this limitation will be reduced or eliminated.

[L.3]
TCAS provides no protection against aircraft with an identification number of zero or all ones.

[L.4]
TCAS does not currently indicate horizontal escape maneuvers and therefore does not (and is not intended to) increase horizontal separation.

[L.5]
If only one of two aircraft is TCAS equipped while the other has only ATCRBS altitude-reporting capability, the assurance of safe separation may be reduced (9FTA-2 0).

[L.6]
Aircraft performance limitations constrain the magnitude of the escape maneuver that the flight crew can safely execute in response to a resolution advisory. It is possible for these limitations to preclude a successful resolu-
↓2.tion of the conflict (38, 2).3

[L.7]
Escape maneuvers are provided only for intruder aircraft with a mode C (altitude-reporting) transponder. For non-altitude-reporting intruders, only proximity advisories are provided.

[L.8]
TCAS will not issue an advisory if it is turned on or enabled to issue resolu-
→FTA-4tion advisories in the middle of a conflict (05).

[L.9] Use by the pilot of the self-test function in flight will inhibit TCAS operation for up to 20 seconds depending upon the number of targets being tracked. The ATC transponder will not function during some portion of the self-test sequence.

System Design Constraints

... wa ... ments ma
... imitations on potentia ... desi

Non-Safety-Related Design Constraints

[C.1]
The system must make use of the radar beacon transponders routinely carried by aircraft for ground ATC purposes.

[C.2]
Maneuvers expected in response to resolution advisories provided by the system should be acceptable to both pilots and air traffic controllers.

[C.3]
The design should not preclude future modification or interfacing with additional airborne or ground equipment that may be necessary to provide expanded collision-avoidance services.

[C.4]
TCAS must comply with all applicable FAA and FCC policies, rules, and philosophies (2.30, 2.79).

[C.5]
The system must provide the least disruptive advisory that will still achieve safe separation.

[C.6]
The system must be compatible with the existing ATC system when operating in controlled airspace and also with planned evolution of the ATC system.

[C.7]
The performance monitor must not interfere with the performance of the intended function of TCAS.

Safety-Related Design Constraints

[illegible]

[illegible] The system design must assure that the system does not have any characteristics that could adversely affect the safety of flight or interfere with the operation of other aircraft and airspace-management systems in a way that could adversely affect the safety of flight.

Specific safety constraints:

[SC1]
The system must function completely independently of the ground-based ATC system.

Assumption: TCAS will be used as a backup to the ATC system.

[SC2]
The system must not interfere with the ground ATC system or other aircraft transmissions to the ground ATC system (↓2.6, 2.13).

[SC2.1]
The system design must limit interference with ground-based secondary surveillance radar, distance-measuring equipment channels, and with other radio services that operate in the 1030/1090 MHz frequency band (↓2.5.1).

[SC2.1.1]
The design of the Mode S waveforms used by TCAS must provide compatibility with Modes A and C of the ground-based secondary surveillance radar system (↓2.6).

[SC2.1.2]
The frequency spectrum of Mode S transmissions must be controlled to protect adjacent distance-measuring equipment channels.

[SC2.1.3]
The design must ensure electromagnetic compatibility and non-occurrence of operationally significant interference between TCAS, secondary surveillance radar, and DME (Distance Measuring Equipment) systems (2.14).

[SC2.2]
The design must ensure that no transponder is suppressed by TCAS activity for more than 2 percent of the time and that TCAS does not create an unacceptably high FRUIT rate for the ATC radars (2.13, 2.13.4).

[SC2.3]
Multiple TCAS units within detection range of one another (approximately 30 nmi) must be designed to limit their own transmissions. As the number of such TCAS units within this region increases, the interrogation rate and power allocation for each of them must decrease in order to prevent undesired interference with ATC (2.13).

[SC3]
The system must generate advisories that require as little deviation as possible from ATC clearances (2.30).

[SC4]
The system must not fail to provide effective warnings and appropriate collision avoidance guidance on potentially dangerous threats nor fail to provide them within an appropriate time limit (2.17.2).

Note: This constraint has obvious conflicts with SCSC5.

[SC4.1]
A resolution advisory must not be removed (canceled) until the other aircraft is no longer a threat.

[SC4.2]
An RA must not be delayed beyond the closest point of approach because of an intruder maneuver (FTA-350, FTA-400).

[SC4.3]
An RA must not be delayed beyond the time that the pilot can respond appropriately.

[SC4.4]
TCAS must protect against a possible maneuver or speed change by either aircraft.

[SC4.5]
TCAS must allow for increased altimetry error as altitude increases (2. 1, 2.32).

[SC4.6]
TCAS must protect against a slow rate of closure allowing a threat to
2. slip in very close without triggering an advisory (2.4, 2.30.2).

[SC4.7]
A Traffic Advisory must be given with enough time for pilots to get visual acquisition before an RA becomes necessary (2. 20).

[SC4.8]
If there are more traffic advisories to generate than can be accomodated
2. on the screen, a priority must be used based on severity (22, FTA-335, FTA-).

[SC4.9]
TCAS must never incorrectly classify an aircraft as on the ground (2. 19).

[SC5]
The system must operate with an acceptably low level of unwanted or nuisance alarms. The unwanted alarm rate must be sufficiently low to pose no safety of flight hazard nor adversely affect the workload in the cockpit (2. 2.3, 2.5.2, 2.32, 2.43, 2.44).

[SC5.1]
The system must control synchronous garbling, nonsynchronous gar-
2. bling, and ground-reflected (multipath) signals (10, 2.11, 2.12, Page 61).

[SC5.1.1]
The probability that a surveillance track based on FRUIT replies
2. will be started and maintained must be extremely remote (11, 2.12).

[SC5.1.2]
The initiation of surveillance tracks based on multipath replies must be avoided (2. 12).

[SC6]
The system must not disrupt the pilot and ATC operations during critical
2. phases of flight nor disrupt routine aircraft operation (2.3, 2.19, 2.24.2).

[SC6.1]
The pilot of a TCAS-equipped aircraft must have the option to switch to the Traffic Advisory-Only Mode where TAs are displayed but display
2. of resolution advisories is inhibited (2.3).

Assumption: This feature will be used during final approach to parallel runways, when two aircraft are projected to come close to each other and TCAS would call for an evasive maneuver.

[SC6.2]
The number and complexity of the controls must be reduced to a minimum consistent with safe system operation. Where possible, control functions should be performed automatically.

[SC6.3]
Aural alerts must be inhibited below 400 feet AGL and by a higher priority windshear or GPWS alert (2 74.6).

[SC6.4]
TCAS must prevent, as much as possible, maneuvering advisories being issued for planned close separation (approaches to parallel runways) (2.42).

[SC7]
TCAS must not create near misses (result in a hazardous level of vertical separation) that would not have occurred had the aircraft not carried TCAS.

[SC7 1]
2.3 Crossing maneuvers must be avoided if possible (6, 2.38, 2.48, 2.4 2).

[SC7 2]
2. The reversal of a displayed advisory must be extremely rare (51, 2.56.3, 2.65.3, 2.66).

[SC7 3]
TCAS must not reverse an advisory if the pilot will have insufficient time to respond to the RA before the closest point of approach (four seconds or less) or if own and intruder aircraft are separated by less than 200 feet vertically when 10 seconds or less remain to closest point of approach (2 52).

[SC8]
The system must not cause or contribute to a controlled maneuver into terrain.

[SC8.1]
TCAS must not issue a descend advisory when the aircraft is near the ground (↓2.4 1, ←Page 87).

[SC8.2]
Advisories must be chosen such that the aircraft flight profile will not
↓2.4 fall below the standard glidepath (6).

[SC9]
TCAS must not cause or contribute to the pilot losing control of the aircraft (↓2.4 1).

[SC9 1]
TCAS must not command maneuvers that significantly reduce stall margins or result in stall warnings (↓2.3.7 1).

[SC10]
TCAS must not interfere with other safety-related systems on the aircraft or contribute to non-separation-related hazards.

[SC10.1]
At low altitudes, TCAS must not interfere with the Ground Proximity Warning System (↓2. 5.3).

[SC10.2]
TCAS must not interfere with the Windshear Alerting System.

[SC10.3]
The system must attempt to reduce RAs requiring the TCAS aircraft to maintain high vertical rates of climb or descent.

[SC11]
The design of the controls and displays must not contribute to safety-critical pilot errors.

[SC11.1]
The operation of controls intended for use during flight, in all possible combinations and sequences, must not result in a condition whose presence or continuation would be detrimental to the continued performance of TCAS or the safe behavior of the aircraft.

[SC11.2]
Controls that are not normally adjusted in flight must not be readily accessible to flight crew personnel.

[SC11.3]
Information must remain on the display for a minimum amount of time even if it is no longer relevant (21, 2.28.1, 2.50, 2.57, 4).

[SC11.4]
The design of the display must not confuse the pilots (79).

Hazard Analysis

[illegible]

> **HAZARD:** Near midair collision (NMAC).
>
> An encounter for which, at the closest point of approach, the vertical separation is less than 100 ft and the horizontal separation is less than 500 ft.

NMACs can occur in either of two ways:

1. The aircraft can be on near collision course and TCAS II fails to provide resolution (the unresolved NMAC)
2. The aircraft can be in close proximity and TCAS II can induce a maneuver that degrades vertical separation to the extent that an NMAC occurs (the induced NMAC).

> **HAZARD:** TCAS causes controlled maneuver into ground (e.g., descend command near terrain)

HAZARD: TCAS causes pilot to lose control of the aircraft.

HAZARD: TCAS interferes with other safety-related systems (e.g., interferes with ground proximity warning).

[illegible] ... in this e ... the spe ... in the p ... so

[illegible] ... ween e ... them ... ation o ... [illegible] TCAS [illegible]

[illegible] We have changed the standard fault tree notation in this example. The standard format required too much space to be practical for such a large tree and to fit even reasonable chunks of the tree on standard size paper with the print large enough to be legible. Instead, indentation is used to denote the normal vertical refinement. Sentences with no special beginning and ending punctuation represent the information that would normally be inside a box. Comments (not intermediate nodes) are denoted by ... and leaf nodes (those not refined further) by [...]. Statements connected by OR gates are grouped using a single dotted line while those connected by AND gates are denoted by a double dotted line. Links to other parts of the intent specification are printed in bold face, are contained in boxes with rounded corners.

FAULT TREE ANALYSIS

Near Midair Collision (NMAC): Two aircraft come within 100’ vertically with close horizontal proximity.

OR

- Maneuver is required to avoid NMAC; pilot does not maneuver aircraft such that NMAC is avoided. (A)
- No maneuver is required to avoid NMAC; an untimely maneuver is made that results in an NMAC (B)

A: Maneuver is required to avoid NMAC; pilot does not maneuver aircraft such that NMAC is avoided.

- Controller instructions (if any) do not lead pilot to maneuver aircraft so as to avoid NMAC.
  - Controller does not issue any instruction. A1
  - <Controller issues an instruction that the pilot does not receive.>
  - <Controller issues an instruction that the pilot may receive but does not follow.>
  - <Controller issues an instruction that does not avoid the NMAC.>
- Pilot does not maneuver the aircraft so as to avoid the NMAC based on his perception (if any) of the conflict.
  - Pilot does not maneuver aircraft so as to avoid NMAC based on his own evaluation
    - Pilot does not realize there is a conflict. A2
    - Pilot realizes there is a conflict but does not maneuver aircraft to avoid the NMAC. A3
    - <Pilot realizes there is a conflict but cannot avoid it due to aircraft system failure.>
  - Pilot does not avoid NMAC by the use of a TCAS resolution advisory.
    - TCAS does not display a resolution advisory. A4
    - TCAS displays a resolution advisory but not in time to avoid NMAC. A5
    - TCAS displays a resolution advisory that the pilot does not follow. A6
    - TCAS displays a resolution advisory that will not avoid the NMAC. A7

A1 Controller does not issue any instruction

80 Controller does not perceive the conflict.

- 85 Conflict alert does not cause controller to perceive conflict.
  - 90 No conflict alert is displayed.
    - 95 [Threat is non-transponder aircraft.]
    - 100 <Computer systems fail.>
    - 105 <Encounter is beyond conflict alert capabilities.>
  - 110 <Controller believes conflict alert is a false alarm.>
- 115 <Controller does not perceive the conflict from his flight information.>

120 <Controller perceives the conflict, but cannot devise a resolution advisory.>

125 <Controller perceives the conflict and may be able to devise a resolution advisory, but does not have the time.>

A2 Pilot does not realize there is a conflict.

130 Pilot does not realize there is a conflict (by visual "see and avoid) unaided by TCAS.

135 <Visual conditions inadequate>

140 Crew is not looking for the threat.

145 <Crew is relying on TCAS to identify potential threats. → **OP.7**

150 <Crew is preoccupied.>

155 <Crew may be looking but the threat is not in view.>

160 <Crew sees the threat but does not perceive that there is a conflict.>

165 <Pilot does not perceive that there is a conflict from monitoring voice communications.>

170 Pilot does not realize that there is a conflict from a TCAS traffic advisory.

175 TCAS does not display a traffic advisory. A8

180 <Crew does not perceive the traffic advisory>

185 <Crew does not perceive the conflict the traffic advisory indicates.>

A3 Pilot realizes there is a conflict but does not maneuver aircraft as to avoid the NMAC.

190 Pilot cannot select a maneuver.

195 [Inadequate visual conditions (not bright daylight.)]

200 Pilot does not make a maneuver in time to avoid NMAC.

205 <Has not visually acquired the threat.>

210 <Selects and executes a maneuver too late.> → **OP.10**

215 Pilot takes an action that does not avoid NMAC.

220 <Pilot did not avoid NMAC because he visualized the situation incorrectly.>

225 <Pilot did not avoid NMAC because he misinterpreted voice communications.>

230 Pilot did not avoid NMAC because of information provided on the TA display.

(Extended development of this fault was not done by MITRE as it was judged to be a human factors-dependent fault.)

A4 | TCAS does not display a resolution advisory.

- 235 TCAS unit is not providing RAs.
  - 240 <Self-monitor shuts down TCAS unit.>
  - 245 Sensitivity level set such that no RAs are displayed.
    - 250 <Own altitude less than 500 feet AGL.>
    - 255 <Pilot selects sensitivity level less than 4 manually.>
    - 260 <Mode S uplink selects sensitivity level less than 4.>
- 265 No RA inputs are provided to the display.
  - 270 No RA is generated by the logic.
    - 275 Inputs do not satisfy RA criteria.
      - 280 Surveillance puts threat outside corrective RA position.
        - 285 Surveillance does not pass adequate track to the logic.
          - 290 [Threat is non-Mode C aircraft.] **→ L.5**
          - 295 <Surveillance failure.>
        - 300 <Surveillance error causes incorrect range/range rate to be calculated.>
      - 305 Altitude reports put threat outside corrective RA position.
        - 310 Altitude errors put threat on ground.
          - 320 <Uneven terrain.> **↓ 2.19**
          - 315 <Intruder altitude error.>
          - 325 <Own Mode C altitude error.> **→ 1.23.1**
          - 330 <Own radar altimeter error.> **→ 1.23.1**
        - 335 Altitude errors put threat in non-threat position.
          - 340 <Own altitude error.> **→ 1.23.1**
          - 345 <Intruder altitude error.>
      - 350 <Intruder maneuver causes logic to delay RA beyond CPA.> **→ SC4.2 ↓ 2.35**
    - 355 <Undetected logic design flaw.>
    - 360 <Logic is coded incorrectly.>
    - 365 <Processing hardware failure.> **→ 1.23.1**
  - 370 <Processor/display connectors fail.>
- 375 <Display is preempted by other functions.> **→ SC4.8 ↓ 2.22**
- 380 <Display hardware fails.> **→ 1.23.1**

A5 — TCAS displays an RA but not in time to avoid NMAC.

385 RA is delayed beyond time when maneuver can avoid NMAC.

390 Conflict was created late.

395 <Own aircraft’s motion created the conflict.> **↓ 2.51, 2.71.3, 2.71.4**

400 <Intruder aircraft’s motion created the conflict.> **→ SC4.2**

405 TCAS was enabled to issue resolution advisories in the midst of the conflict. **→ L.8**

AND:

410 <Own aircraft was in a conflict when TCAS enabled to issue RAs>

415 TCAS enabled to issue RAs.

420 <TCAS was just turned on in any 25 second period.>

425 <Own altitude increases to the point where RAs are enabled.>

430 <Mode S ground station enables RAs.>

435 <Pilot switches sensitivity level to enable RAs.>

440 TCAS acquired track in the midst of a conflict.

AND:

445 <Own aircraft in a conflict when TCAS acquired track>

450 TCAS acquires track late.

455 <Aircraft previously judged "on ground" is now judged "in air.">

460 <Intruder transponder just turned on.>

465 <Interference-limiting feature previously eliminated threat.>

470 <Intruder motion not within limits expected by Mode S surveillance.>

475 <Surveillance acquired late.>

480 Low firmness delays RA. **↓ 2.35, 2.36.3**

485 Altitude credibility tests rejected reports.

490 <Noisy surveillance data.>

495 <Stuck Mode C bit.>

500 <Intruder acceleration exceeds that expected.>

505 <Intruder was perceived to be maneuvering.>

A6 TCAS displays a resolution advisory that the pilot does not follow.

510 Pilot does not execute the RA at all. → **OP.2**

515 Crew does not perceive RA alarm.

520 <Inadequate alarm design.> → **1.4 to 1.14** ↓ **2.75, 2.76**

525 <Crew is preoccupied.>

530 <Crew does not believe RA is correct.> → **OP.1**

535 Pilot must clear his airspace before maneuvering, but cannot.

540 <Pilot cannot clear his airspace due to visibility (IMC, glaring sun, ...)>

545 <Pilot can clear his airspace (good VMC) but is unable>

550 Pilot executes the RA but inadequately.

555 <Pilot stops before RA is removed.> → **OP.10**

560 <Pilot continues beyond point RA is removed.> → **OP.4, OP.10**

565 <Pilot delays execution beyond time allowed.> → **OP.10**

A7 TCAS displays a resolution advisory that does not avoid the NMAC.

570 <TCAS is not shut down by self-monitor or sensitivity level>

575 <TCAS generates for display a resolution advisory that will not avoid the NMAC.>

580 Own TCAS generates an incorrect RA.

585 RA is removed before aircraft is out of NMAC

590 <RA is given with incorrect sense and removed before altitude crossing +100’>

595 <RA is given with correct sense and removed before aircraft is out of NMAC.>

600 <Standard vertical rate is insufficient to achieve 100’ separation.>

605 TCAS receives (via coordination link) an incorrect RA complement.

610 [Threat is TCAS equipped.]

615 <Threat generates an incorrect RA.>

A8 TCAS does not display a traffic advisory.

- 620 TCAS unit is not providing traffic advisories.
  - 625 Sensitivity level set such that no TAs are provided.
    - 630 [Pilot sets sensitivity level manually.]
    - 635 [Mode S ground sensor sets sensitivity level.]
  - 640 <Self-monitor shuts down TCAS unit.>
- 645 No TA inputs are provided to the display.
  - 650 No traffic advisory is generated by the logic.
    - 655 Inputs do not satisfy threat criteria
      - 660 Surveillance does not provide a track that passes range test.
        - 665 Surveillance does not pass adequate track to the logic.
          - 670 <Threat is non-Mode C aircraft.>
          - 675 <Surveillance failure.>
        - 680 <Surveillance fault causes incorrect range/range rate to be calculated.>
      - 685 Altitude reporting causes threat not to be judged a threat.
        - 690 [Threat is altitude-encoding aircraft]
        - 695 Threat is judged not to be a threat by altitude tests.
          - 700 <Threat is judged to be on ground.>
          - 705 <Threat is judged to pass with greater than ZTHR separation.>
    - 710 <Undetected logic design flaw.>
    - 715 <Logic is coded incorrectly.>
    - 720 <Processing hardware fails.> → **1.23.1**
  - 725 <Processor/display connectors fail.>
- 730 Display limitation prevents display of threat
  - 735 <Multiple threats cause this one to be eliminated.> → **SC4.8** ↓ **2.22**
  - 740 <Intruder overlaps own-aircraft symbol.>
- 745 <Other function preempts display.>
- 750 <Display hardware fails.> → **1.23.1**

B — No maneuver is required to avoid NMAC; an untimely maneuver is made that results in an NMAC.

755 Own aircraft makes an untimely maneuver.

- 760 Pilot maneuvers aircraft because of instruction provided to him.
  - 765 Pilot is issued an instruction that will lead to NMAC. B1
  - 770 Pilot follows the instruction because he does not see it will lead to an NMAC. B2
- 775 Pilot maneuvers aircraft because of his own evaluation of the situation.
  - 780 Pilot evaluates the situation and selects an untimely maneuver. B3
  - 785 TCAS traffic advisory (if any) does not show the maneuver is incorrect. B4

790 No system independent of the pilot corrects own aircraft's maneuver before the NMAC occurs.

- 795 Controller does not transmit a corrective instruction. B5
- 800 TCAS does not display a resolution advisory that corrects.
  - 805 TCAS does not display a resolution advisory. A4
  - 810 TCAS displays a resolution advisory but not in time to avoid NMAC. A5
  - 815 TCAS displays a resolution advisory that the pilot does not follow. A6
  - 820 TCAS displays a resolution advisory that will not avoid the NMAC. A7
  - 825 TCAS displays a resolution advisory that would avoid NMAC except the threat maneuvers. B6
  - 830 TCAS does not issue an "Advisory Not OK."

(This branch was not completed in the MITRE fault tree.)

B1 **Pilot is issued an instruction that will lead to NMAC.**

835 <Incorrect instruction is transmitted by Controller>

840 TCAS displays an RA that will lead to NMAC.

845   <TCAS is not shut down by self monitor or sensitivity level>

850   TCAS generates for display an RA that will lead to NMAC. ↓ **2.67**

855   Altitude error causes wrong sense RA that leads to NMAC.

860   <Wrong sense RA is chosen>

865   Pilot stops following RA within 100 feet of threat → **OP.10**

870   <RA is removed within 100 feet of threat due to altimetry error.> → **1.23.1**

875   <RA is removed before NMAC; pilots follows it until within 100’ of threat.>

880   <RA is removed after altitude crossing; pilot stops following it within 100’ of threat, before it is removed.>

885   <C-bit error causes incorrect RA to be generated.>

890   <RA based on apparent trajectory is thwarted by intruder maneuver> ↓ **2.57.5**

895   False track causes spurious RA, which leads to NMAC with real aircraft.

900   <False track causes spurious RA>

905   <Spurious RA leads to NMAC with real aircraft.>

B2 | Pilot follows the instruction because he does not see it will lead to an NMAC.

- 910 Visual acquisition does not show the instruction is incorrect.
  - 915 Pilot does not visually acquire other aircraft.
    - 920 Pilot doesn’t acquire visually without TCAS aid.
      - 925 [No TA is displayed.]
      - 930 <Pilot doesn’t acquire other aircraft unaided>
    - 935 Pilot doesn’t acquire visually with TCAS aid.
      - 940 [TA is displayed.]
      - 945 <Pilot doesn’t acquire other aircraft aided by TCAS>
  - 950 <Pilot acquires other aircraft but does not take corrective action.>
- 955 <Voice communications monitored do not show the instruction is incorrect.>
- 960 TCAS TA display does not show the instruction is incorrect.
  - 965 No traffic advisory is being displayed.
    - 970 [No aircraft is currently a threat.]
    - 975 Aircraft that should be judged a threat is not displayed.
      - 980 <Aircraft is a threat.>
      - 985 Aircraft that is a threat is not displayed. A8
  - 990 <Pilot does not use his TCAS display to check whether the instruction is correct.>
  - 995 Threat and proximity (if any) advisory does not show the instruction is incorrect.
    - 1000 <TCAS displays a threat advisory>
    - 1005 Threat and proximity advisory shown does not show the instruction incorrect. B7
  - 1010 <Pilot does not realize that the display shows the instruction is incorrect.>
- 1015 TCAS is not displaying a preventive RA against the maneuver. A4

B3 | Pilot evaluates the situation and selects an untimely maneuver.

1020 <Pilot maneuvers for reasons other than separation assurance.>

1025 Pilot maneuvers off non-NMAC trajectory for separation assurance.

1035 <Pilot perceives conflict from non-TCAS source and maneuvers into NMAC.>

1040 TCAS provides a TA indicating a conflict that the pilot uses to maneuver into NMAC.

1045 <Pilot decides to maneuver based on a TCAS TA only> → **OP.9**

1050 <TCAS displays a traffic advisory.>

1055 Pilot selects an inappropriate maneuver based on TCAS TA display.

(This branch was not further expanded by MITRE as it was judged to be a human factors-dependent fault.)

B4 TCAS traffic advisory (if any) does not show the maneuver is incorrect.

1060 <Pilot does not use his TA display to check whether the instruction is correct.>

1065 No traffic advisory is being displayed.

1070 [No aircraft is current a threat.]

1075 Aircraft that is a threat is not displayed.

1080 <Aircraft is currently a threat.>

1085 Aircraft that is a threat is not displayed. A1

1090 Traffic advisory being displayed does not show the instruction is incorrect.

1095 <TCAS displays a traffic advisory.>

1100 Traffic advisory displayed does not show the instruction is incorrect. B7

1105 <Pilot does not see that the display shows the instruction is incorrect.>

B5 Controller does not transmit a corrective instruction.

1110 <Controller does not see conflict and generate an instruction.>

1115 Conflict alert does not cause controller to see conflict and generate a resolution

1120 <Controller believes the conflict alert displayed is a false alarm.>

1125 No conflict alert is displayed to the controller.

1130 <Not enough time for conflict alert to perceive conflict and generate alert.>

1135 [Newly created threat is a non-transponder aircraft.]

1140 <Other conflict alert fault.>

B6 TCAS issues an RA that would avoid NMAC except the threat maneuvers.

1145 Threat maneuvers after RA is issued and neither the pilot nor TCAS corrects. ↓ **2.2.2**

- 1150 <Threat maneuvers sufficient to counter RA> ↓ **2.36, 2.38**
- 1155 Neither the pilot nor TCAS recognizes the situation and corrects.
  - 1160 Pilot does not recognize the situation.
    - 1165 Does not see it (visual).
      - 1170 <IMC (if RAs allowed in IMC)>
      - 1175 <Assumes RA OK>
      - 1180 <Cannot acquire threat.>
    - 1185 Does not see it from TA display.
      - 1190 <Not monitoring the display.>
      - 1195 <Display does not show it.>
      - 1200 <Cannot tell that display shows it.>
    - 1205 TCAS does not tell pilot that advisory is not adequate.
      - 1210 <TCAS does not issue "Advisory Not OK">
      - 1215 <Pilot fails to perceive alarm.>
  - 1220 Pilot becomes aware of the situation but cannot correct
    - 1225 <Not enough time to maneuver.>
    - 1230 <Cannot devise a maneuver.>

1235 Other aircraft (different than threat) involved in NMAC.

B7 Traffic and proximity advisories do not show the instruction is incorrect.

- TCAS did not display the proximate aircraft own will maneuver into.
  - No proximity inputs are provided to the display.
    - No TA is displayed.
      - <No TA should be displayed.>
      - <No TA is displayed when one should be> A8
    - TA is displayed but proximate aircraft is not.
      - <No proximity advisory should be displayed.>
      - Proximity advisory should be displayed but is not.
        - Inputs do not satisfy proximity advisory criteria.
          - Surveillance does not pass a track to the logic that is within proximity range.
            - Surveillance does not pass adequate track to logic.
              - <Threat is non-Mode C aircraft>
              - <Surveillance failure.> ↓ **2.51**
            - <Surveillance provides incorrect range.>
          - Altitude reports pass a relative altitude that does not satisfy proximity criterion.
            - AND
              - [Threat is Mode C aircraft
              - Threat is judged not proximate
                - <Threat is judged "on the ground.">
                - <Threat is judged to be 1200’ away vertically.>
        - <Undetected logic design flaw.>
        - <Logic is coded incorrectly.>
        - <Processing hardware fails.> → **1.23.1**
      - Display limitation prevents display of the proximate aircraft.
        - <Multiple aircraft cause this one to be eliminated.> → **SC4.8** ↓ **2.22**
        - <Proximate aircraft overlaps own aircraft symbol.>
        - <Processing hardware fails.> → **1.23.1**
  - <Display hardware failure. → **1.23.1**
- Displays show the proximate aircraft but in the wrong location.
  - <Displays its bearing incorrectly.>
  - <Displays its range incorrectly.>
  - <Displays its relative altitude incorrectly.>

Level 2

System Design Principle

[illegible] the [illegible] ents an [illegible] ow the [illegible] ents a [illegible] algorithms

General Description

When in operation, TCAS periodically transmits interrogation signals. These interrogations are received by Air Traffic Control Radar Beacon System (ATCRBS) or Mode S altitude reporting transponders. In reply to the interrogations, the transponder transmits a signal that reports its altitude. TCAS computes the range of the intruding aircraft by using the round-trip time between the transmission of the interrogation and the receipt of the reply.

Altitude, range, and range rate are determined by tracking the reply information. These data, together with the current TCAS sensitivity level (which specifies the protected volume around the aircraft) are used to determine whether the intruding aircraft is a threat. Each threat aircraft is processed individually to permit selection of the minimum safe resolution advisory based on track data and coordination with other TCAS-equipped aircraft.

If the TCAS threat detection logic determines that a proximate aircraft represents a potential collision or near-miss encounter, TCAS determines the appropriate vertical maneuver that will ensure the safe separation of the TCAS aircraft. The appropriate maneuver is one that ensures adequate vertical separation while causing the least deviation of the TCAS aircraft from its current vertical rate.

If the threat aircraft is itself equipped with TCAS equipment that generates resolution advisories, a coordination procedure via the air-to-air Mode S data link is performed before displaying the advisory to the pilot. This procedure assures that the aircraft resolution advisories are compatible.

Two categories of *resolution advisories* can be displayed to the pilot: (1) *corrective advisories*, instructing the pilot to deviate from the current flight path (e.g. to alter the vertical speed of the aircraft to ensure safe separation from nearby traffic, CLIMB when the aircraft is level or descending); and (2) *preventive advisories* requiring no immediate action but warning the crew not to alter current vertical speed due to nearby traffic (e.g. DON'T CLIMB when the aircraft is in level flight).

The *traffic advisories* displayed to the pilot describe the positions of proximate aircraft that are not considered threats. The display of traffic advisories alert the flight crew to the presence of threats or potential threats, and may improve the ability of the crew to respond to subsequent resolution advisories. In addition, traffic advisories may improve the ability of the flight crew to acquire the traffic visually.

TCAS has the capacity to communicate with the ground-based air traffic control system when a ground-based Mode S sensor equipped with the necessary complementary features has been installed. TCAS can provide the Mode S ground system with the resolution advisories that are displayed to the pilot. These resolution advisories may be displayed to the air traffic controller if desired. In addition, airborne TCAS equipment can receive sensitivity-level commands from ground-based Mode S sensors. (To date, such facilities have not been used.)

TCAS System Components

Figure shows the TCAS II system components.

Mode S/TCAS Control Panel: This panel selects and controls all TCAS elements including the Surveillance and Collision Avoidance System (CAS) logic, the Mode S transponder, and the TCAS displays. Control information is provided to the surveillance and CAS logic via the Mode S transponder (page 77).

Mode S Transponder: The Mode S transponder performs the normal ATC functions of existing Mode A and C transponders. Because of its selective address capability, the Mode S transponder is used also to provide air-to-air data exchange between TCAS-equipped aircraft to ensure coordinated, complementary resolution advisories (page 9).

Surveillance and Collision Avoidance System Logic: This component performs airspace surveillance (page 77), intruder and own aircraft tracking (page 69), threat detection and resolution (page 83), and advisory generation (page 89). Pressure and radar altimeter inputs (page 9), and other aircraft configuration discrete (page 9) inputs are used to control the collision avoidance logic parameters that determine the protection volume around the TCAS aircraft. If a tracked aircraft is a collision threat, the collision avoidance logic selects the best avoidance maneuver. If the threat aircraft is also equipped with TCAS II or III, this maneuver is coordinated with the threat aircraft's maneuver.

Antennas: The antennas used by TCAS II include a directional antenna that is mounted on top of the aircraft. Typically, this antenna transmits interrogations on 1 030 MHz at varying power levels in each of four 90 degree azimuth segments. Transponder replies are received on 1 090 MHz and are sent to the TCAS surveillance component. The directional antenna permits the partitioning of replies to reduce synchronous garbling (5.1.2). Figure shows an

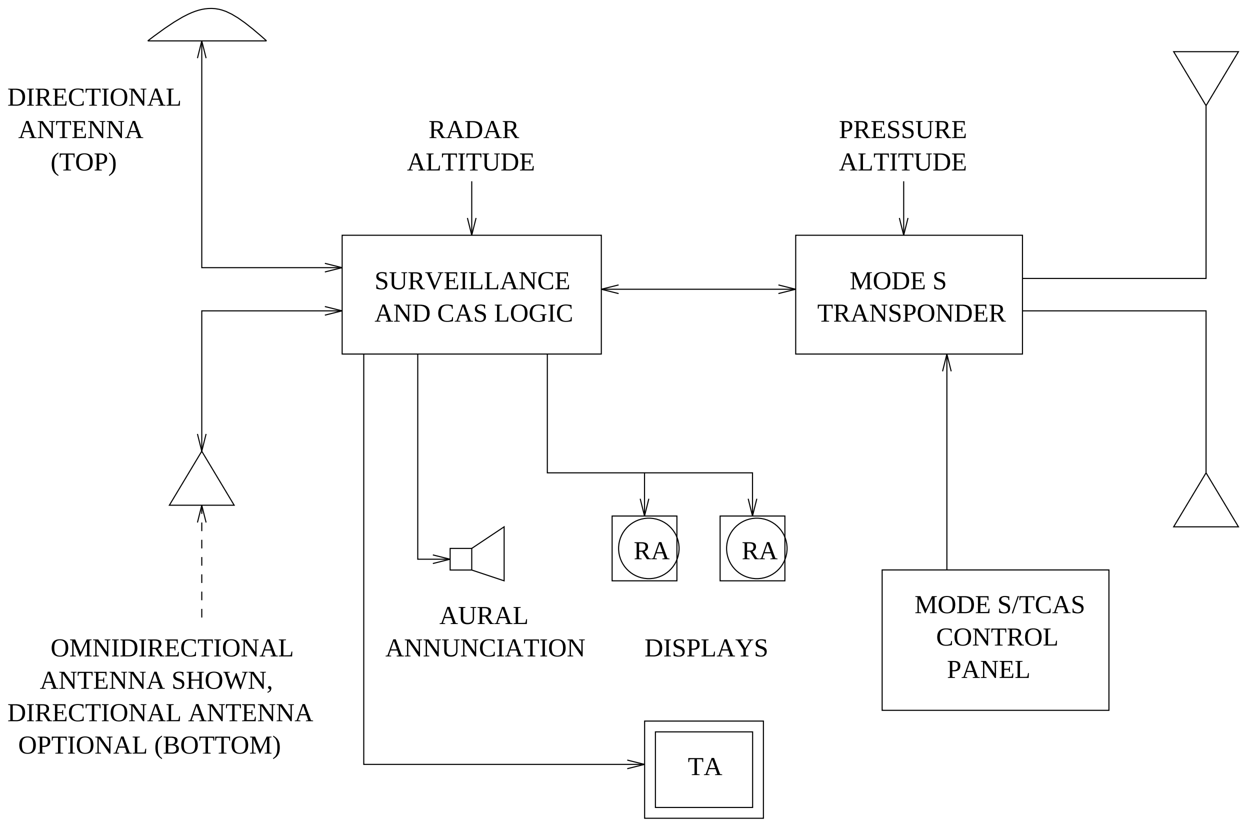


Figure TCAS II Component Diagram

omnidirectional transmitting and receiving antenna mounted on the bottom of the aircraft to provide range and altitude data on intruders that are below the TCAS aircraft (21). An optional directional antenna may be used on the bottom to obtain bearing information on these targets (1, page 7). The Mode S transponder is interrogated on 1030 MHz and responds on 1090 MHz, respectively using its upper and lower antennas to optimize signal strength and reduce multipath interference (5.1, 2→2).

Traffic Advisory Display: The traffic advisory (TA) display depicts the position of the traffic relative to the TCAS aircraft to assist the pilot in visually acquiring intruding aircraft (1, page 6). This display can be either a dedicated TCAS display or a weather radar display. Alternatively, in some aircraft the TA display will be an electronic flight instrument system (EFIS) panel display that combines traffic and resolution advisory information on the same scope face.

A number of manufacturer options are allowed for the TA display that will satisfy the official TCAS requirements. Displays can be on continuously or they may be activated only when TCAS detects an intruding aircraft or when the pilot wants to check the traffic situation. Displays can allow pilot selection of display range and altitude band or can be restricted to one range and one altitude band.

Resolution Advisory Display: The resolution advisory (RA) display is a standard modified Vertical Speed Indicator (VSI), that indicates the vertical rate that must be achieved to maintain safe separation from threatening aircraft (Page 107). The RA display contains segmented red and green eyebrow lights around the vertical speed scale.

When an RA is generated, TCAS II lights up the appropriate segments of the VSI display. Simply keeping the needle out of the red segments complies with the resolution advisory (↑1.8).

Normally, there are two RA displays, one for the Captain and one for the First Officer (FO) (↑3). In some cases, the TA, and RA displays will be combined; i.e., traffic information is shown in the center portion of an electronically displayed VSI.

Aural Annunciations: Displayed traffic and resolution advisories are supplemented by synthetic voice advisories generated by the TCAS computer (↑1.4 – Page 105). The words TRAFFIC, TRAFFIC are annunciated at the time of the traffic advisory, which directs the pilot to look at the TA display to locate the intruding aircraft (↑1.6). If the encounter does not resolve itself, a resolution advisory (e.g., CLIMB, CLIMB, CLIMB) is annunciated (↑1.6). At this point the pilot needs to adjust or maintains the vertical rate of the aircraft to keep the VSI needle out of the red segments (↑1.8).

Surveillance and Collision Avoidance Logic

General Concepts

[2.1]

ALIM: desired or "adequate" separation between aircraft that TCAS is designed to meet. This amount varies from 400 to 700 feet, depending on own aircraft [illegible] includes allowances to account for intruder and own altimetry errors and vertical tracking uncertainties. [illegible] increases with altitude to [illegible] altimetry error ([illegible] 5) and the need to increase tracked separation at higher altitudes.

[2.2]

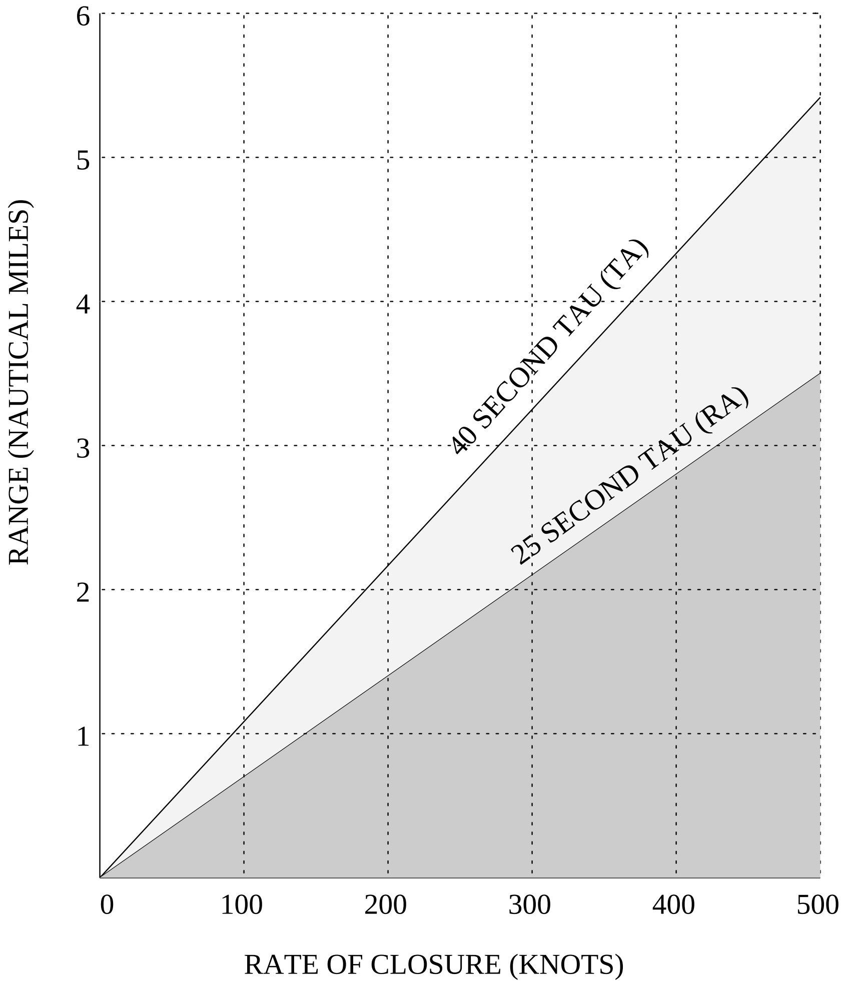
Protected Volume: Each TCAS-equipped aircraft is surrounded by a protected volume of airspace. The [illegible] of the protected volume depends on the speeds and headings of the aircraft involved and also on a selected sensitivity level. The dimensions are not based on actual distance, but rather on [illegible] time to closest point of approach [illegible]

[2.2.1]

Tau (range/range rate) concept: time-to-go [illegible] to the closest point of approach (CPA) [illegible]. Tau is an approximation of the time in seconds to CPA. Tau equals 3600 times the slant range in nmi, divided by the closing speed in knots.

$$\frac{3600 \times \text{slant-range}}{\text{closing-speed}}$$

Note that 3600 is used because tau is given in seconds while closing speed is specified in knots.



Sensitivity Level 5

TCAS II is based on the *tau* concept for all alerting functions. As an example, Figure 2.2 shows the combinations of range and closure rate that would trigger a TA with a 40-second *tau* and an RA with a 25-second *tau*. These TA and RA *tau* values correspond to SL (sensitivity level 5).

[2.2.2]

DMOD

If the rate of closure is very low, a target could slip in very close without crossing the *tau* boundaries and triggering a TA or an RA. In order to provide added protection against a possible maneuver or speed change by either aircraft (↑FTA-114), the *tau* boundaries are modified (see Figure 2.3). This modification is called DMOD.

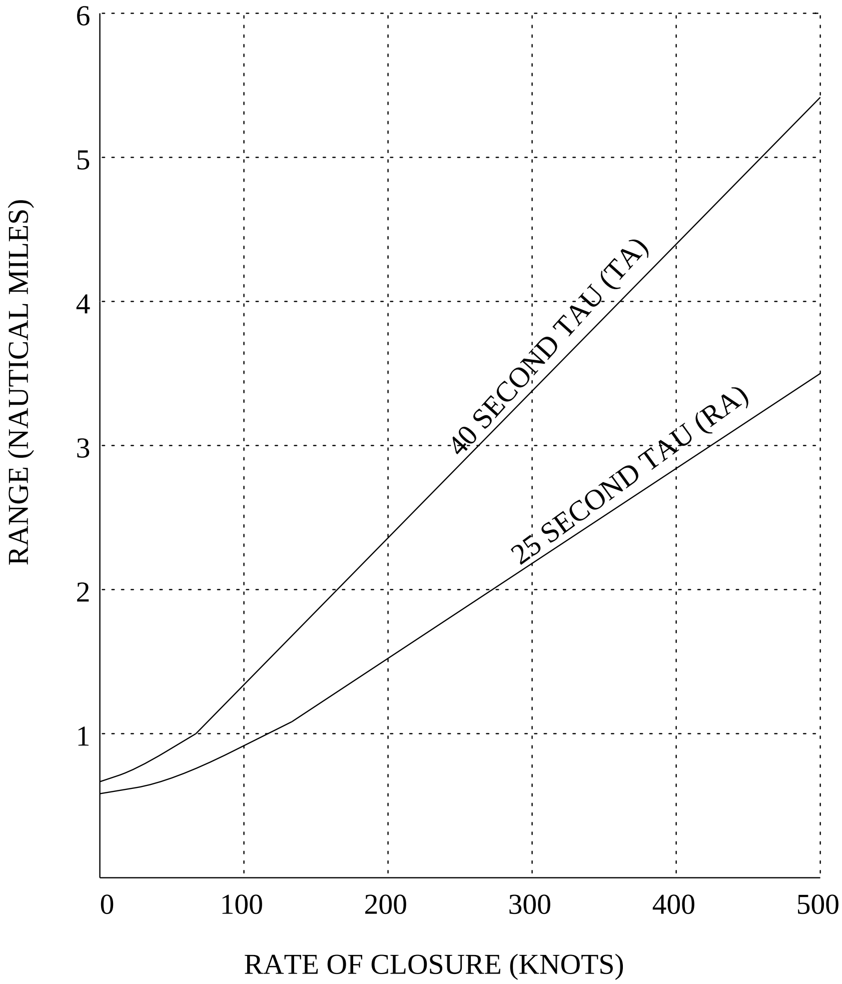


Figure 2.3: TA/RA *tau* Values for Sensitivity Level 5

varies depending on own aircraft's altitude regime (→2.2.4.)

[2.2.3] **Sensitivity level**: Tradeoffs must be made between necessary protection and unnecessary advisories (↑1.18, SC5, SC6). This is accomplished by controlling the sensitivity level (SL), which controls the *tau*, and therefore the dimensions of the protected airspace around each TCAS-equipped aircraft. The greater the SL, the more protection is provided but the higher is the incidence of unnecessary alerts. Progressively smaller sensitivity levels can be selected as a TCAS-equipped aircraft enters low-altitude or terminal airspace (↑OP.6, SC6.1).

The three primary means that TCAS uses to determine the operating SL ground-based, Mode S sensor selection (not in use in airspace), pilot switch selection, and altitude. In addition, aircraft discretes are used that indicate if own aircraft is on the ground and if display of traffic is allowed on the ground. There are 7 sensitivity levels:

Level 1: The TCAS unit is not operating (is in a standby mode) on board the aircraft; this does not place the TCAS transponder in a standby condition, but no interrogations are transmitted.

Assumption: This level will normally be used by the pilot when the aircraft is on the ground.

Level 2: TA-only mode, in which only traffic advisories are issued. The equipment performs all surveillance functions and provides TAs but not RAs. TCAS will select the appropriate level using other inputs, such as barometric and radio altitude. In this way, TCAS will generate traffic advisories at the thresholds appropriate for the altitude regime of own aircraft. SL 2 can be manually selected by the pilot or it is automatically selected when the TCAS aircraft is below 500 ft AGL (above ground level) as indicated by the radar altimeter.

Assumption: This mode will generally be used by the pilot to avoid unnecessary distractions while at low altitudes on final approach to an airport (↑SC 6.1).

Levels 3–7: TAs and RAs are generated in these modes. Table 2 shows the altitude thresholds at which TCAS automatically changes its SL selection and the associated *tau* values for altitude-reporting aircraft. SL 4 uses inputs from the radar altimeter, and SLs 5, 6, and 7 use the height above mean sea level (MSL) as indicated by the barometric altimeter. SL 3 cannot be selected automatically at this time—it is reserved for possible future use in areas where further decreases in sensitivity are required.

Table 2: Sensitivity Level Selection Based on Altitude

Altitude	Sensitivity Level	Tau Values in Seconds	
		TA	RA
0-500 AGL	2	20	n.a.
500-2500 AGL	4	35	20
2500-10000 MSL	5	40	25
10000-20000 MSL	6	45	30
Above 20000 MSL	7	48	35

[2.24]

3-D Protected Volume: For encounter geometries involving low vertical closure rates (↑SC4.6) the vertical dimension of the protected volume for TAs is 1200 feet above and below the altitude of own aircraft. The vertical dimension for RAs (called ZTHR) varies from 600 to 700 feet depending on own aircraft's altitude regime (see Figure 4). For high vertical closure rates, a TA or RA would be triggered when the predicted time to co-altitude drops below the *tau* values shown in Table 4 (↑1.18).

Surveillance

Mode S Surveillance

[2.3]

TCAS listens for the spontaneous transmissions (*squitters*) generated by Mode S transponders. The squitter transmissions contain the individual Mode S address of the sender.

[2.4]

When TCAS receives a squitter, it sends a Mode S interrogation message to the Mode S address contained in the message, and, from the reply, determines the range, bearing, and (if available) altitude of the Mode S aircraft. Detection of Mode S targets can also be determined by listening for Mode S replies generated in response to ground stations or by other TCAS aircraft.

[2.5]

TCAS determines the range and altitude of each Mode S target (↑1.18). These target reports are provided to the collision avoidance logic for advisory detection and display. Relative bearing is also provided for presentation on the traffic advisory display, but is not necessary for threat detection or resolution because TCAS currently provides resolution advisories only in the vertical direction (↑1.22)

[2.5.1]

The rate at which a Mode S aircraft is interrogated by TCAS depends on its range and closing speed. The interrogation rate increases to once per second when the aircraft warrants a traffic advisory (↑1.19, SC2 1).

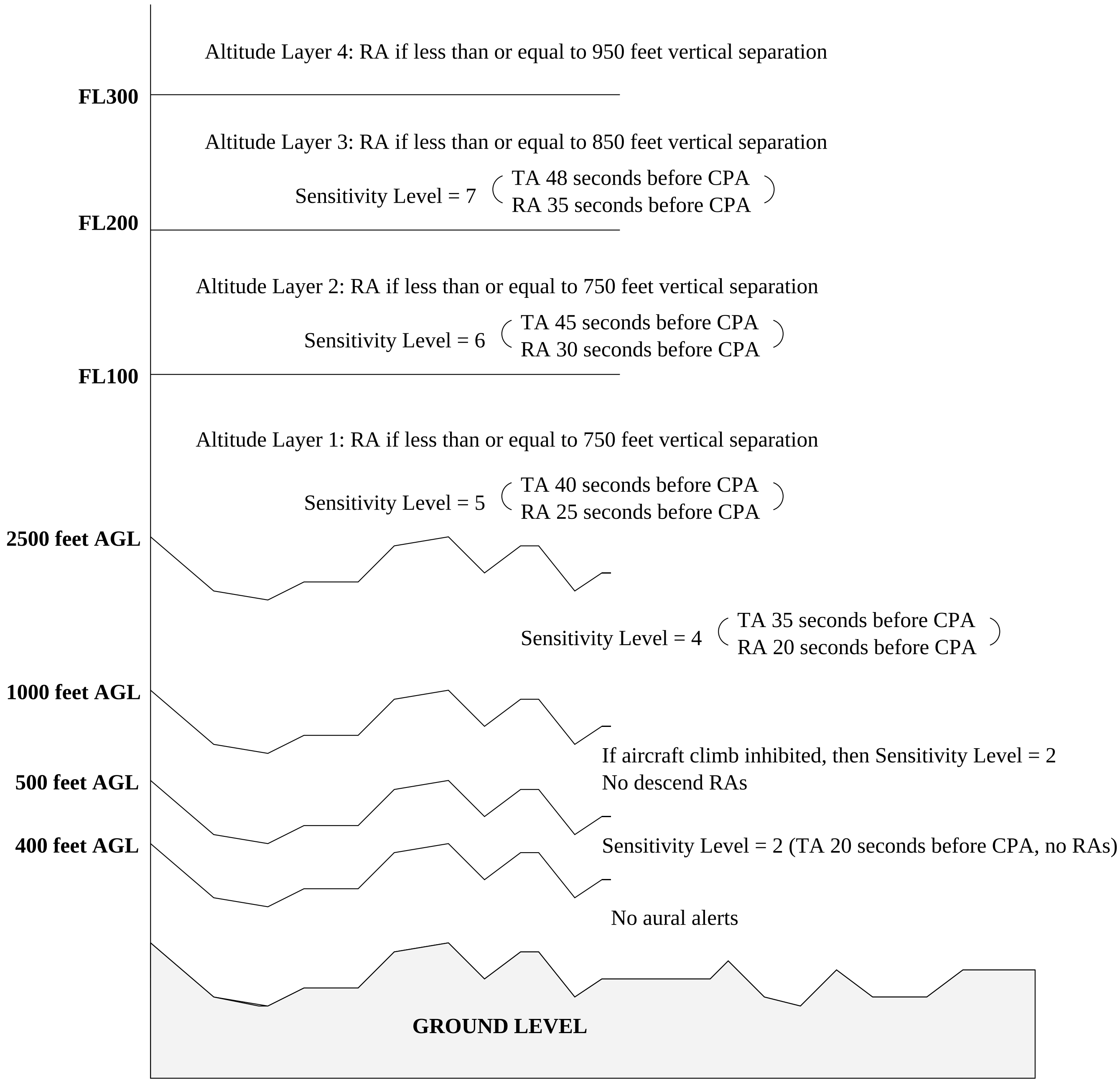


Figure 4: The relationship between altitude and sensitivity level

[2.5.2]

Mode S Resolution advisories: RAs are issued only against Mode S targets whose predicted positions penetrate the threat boundaries within six surveillance update intervals following the receipt of the last valid reply to a track. For Mode S targets whose predicted positions (↑SCC) penetrate the threat boundaries more than six surveillance update intervals following the receipt of the last reply to a track interrogation, RAs are displayed only if the target again satisfies the criteria.

[2.5.3]

Mode S address of the target is retained for ... acquisition process to shorten the reac... received.

Mode A and Mode C Surveillance

[2.6]

TCAS interrogates Mode A/C transponders at a normal rate of once per second, using a modified Mode C interrogation (Mode C only all call ↑SCC 1.1). Mode A transponder targets reply with no data in the altitude field. TCAS uses the framing pulses of the reply to initiate and maintain range and bearing track on these targets.

[2.7]

Range and bearing information from Mode A transponders is used by the traffic advisory detection and display logic. Range and bearing information from Mode C transponders is used by the CAS logic for TA and RA detection and display (↑1.20)

[2.8]

The following conditions are necessary for *Surveillance Initiation Criteria*: generating RAs for ATCRBS targets:

1. Initially, a mode C reply is received from the target in each of three consecutive surveillance update periods, and
   a. The replies do not correlate with surveillance replies from other tracks
   b. The range rate indicated by the two most recent replies is less than 1200 ft.

c. The oldest reply is consistent with the above range rate in the sense that its range lies within 312.5 feet of a straight line passing through the two most recent replies

d. The replies correlate with each other in their altitude code bits.

2. A fourth correlating reply is received within ... surveillance update intervals following the third of the three consecutive replies in 1 above.

[2.9]

TCAS generates resolution advisories for ATCRBS targets whose predicted positions penetrate the threat boundaries within six surveillance update intervals following the receipt of the last valid correlating reply. It does not generate resolution advisories for ATCRBS targets whose predicted positions penetrate the threat boundaries more than six surveillance update intervals following the receipt of the last valid correlating reply unless the target again satisfies the surveillance initiation criteria above.

[2.10]

Synchronous degarbling: When a Mode C only all-call interrogation is issued by TCAS, all transponders that detect it will reply. Because of the length of the reply train (20.3 microseconds), all Mode A/C-equipped aircraft within a range difference of 1.7 nmi from the TCAS aircraft, will generate replies that garble (overlap each other) when received at the TCAS aircraft (↑SC5.1).

[2.10.1]

Whisper-Shout (WS): Hardware degarblers can reliably decode up to three overlapping replies, and the combined use of variable power levels and suppression pulses reduces the number of transponders that reply to a single interrogation. This technique is called whisper-shout (WS). A first interrogation is issued at the lowest power level used for the WS sequence. During the next step, a suppression pulse is transmitted at a slightly lower level than the first interrogation, followed 2 microseconds later by an interrogation at a slightly higher lower level. This action suppresses most of the transponders that had replied to the previous interrogation, but elicits replies from an additional group of transponders that did not reply to the previous interrogation. The WS procedure is followed progressively in 24 steps, to separate the Mode A/C replies into several groups, and thus reduce the possibility of garbling. The WS sequence is transmitted once during each surveillance update period, which is nominally one second.

[2.10.2]

Directional transmissions further reduces the number of potentially overlapping replies. Slightly overlapping coverage must be provided in all directions to ensure 360 degree coverage.

[2.10.3]

Mode C only all-call interrogation. This waveform inhibits Mode S transponders from responding to a Mode C interrogation.

[2.11]

Nonsynchronous garble. This phenomenon is caused by the receipt of undesired transponder replies that were elicited by ground interrogators and other TCAS interrogators. These so-called "FRUIT" replies are transitory and identified and discarded (↑SC5.1, SC5.1.1). Reply-to-reply correlation algorithms ...

[2.12]

Multipath replies. Multipath results in the detection of more than one reply to the same interrogation, generally of lower power, from the same aircraft. It is caused by the reflection of an interrogation, usually from flat terrain (↑SC5.1, SC5.1.1). To control multipath, the direct-path power level is used to raise the minimum triggering level (MTL) of the TCAS receiver long enough to discriminate against the delayed and lower power reflections.

[2.13]

Interference Limiting. The surveillance and collision avoidance functions of the TCAS equipment must be capable of reliable operation in nominal traffic densities as specified in 1.19.1. However, the TCAS equipment must also operate without degradation in high density regions of either the electromagnetic or ATC environments (↑SC2, SC3).

[2.13.1]

To minimize its interference effects, TCAS equipment controls its interrogation rate or power and conforms to a set of three inequalities, which are a means of ensuring that all interference effects resulting from these interrogations together with the interrogations from all other TCAS airborne interrogations in the vicinity are kept to a low level.

[2.13.2]

The limits on interrogation rate and power are functions of the local aircraft environment. In the process of checking compliance with the limits, every TCAS unit counts the number of other TCAS units (airborne interrogators) within detection range. This is accomplished by monitoring "TCAS broadcast interrogations." This interrogation indicates that the aircraft is equipped with a TCAS interrogator and includes the Mode S address of the currently interrogating TCAS aircraft. Each TCAS aircraft that is currently interrogating spontaneously transmits these interrogations at 10-second intervals. TCAS monitors the receipt of such interrogations by its own Mode S transponder to update once each second the number of other TCAS units within detection range.

[2.13.3]

Broadcast messages are monitored by the interference limiting algorithms in TCAS to develop an estimate of the number of TCAS units within detection range. The number of total TCAS units is used by each

2 TCAS to limit the interrogation rate and power as necessary (↑SC 3).

[2.13.4]

The three inequalities are associated with the following physical mechanisms: (1) reduction in on-time of other transponders caused by TCAS interrogations, (2) reduction in on-time of own transponder caused by mutual suppression during transmissions of interrogations, and (3) 2) ATCRBS interrogations (↑SC ATCR

[2.13.5]

The inequalities use the following abbreviations:

I: The total number of excluding air-to-air coordination interrogations transmitted by own TCAS in a one-second period,

$P(i)$: The total radiated power (in watts) from the antenna for the i-th interrogation,

NTA: The number of airborne TCAS interrogations detected with a transponder receiver threshold of -74 dBm,

B: The beam sharpening factor ration (ratio of three-dB beamwidth to beamwidth resulting from interrogation sidelobe suppression),

$M(i)$: The duration of the mutual suppression interval for own transponder associated with the i-th interrogation,

K: The total number of ATCRBS interrogations transmitted by own TCAS in a one-second period,

k: The index number of ATCRBS interrogation where k=1,2,K,

$P(k)$: The total radiated power (in watts) from the antenna for the k-th ATCRBS interrogation.

1. Inequality 1 assures that a "victim" transponder will never detect more than 280 TCAS interrogations in a one-second period from all the TCAS interrogators within 30 nmi. For $i=1..I$,

$$\sum \frac{P_i}{250\ Watts} \leq min[\frac{280}{1 + NTA}, 18] \qquad (2.1)$$

The left-hand side of the inequality allows a TCAS unit to increase its interrogation rate if it transmits at less than 250 W since low-power transmissions are detected by fewer transponders. The denominator of the first term on the right-hand side of this inequality accounts for other TCAS interrogators in the vicinity and the fact that all TCAS units must limit their interrogation rate and power in a similar manner so that, as the number of TCAS units in a region increases, the interrogation rate and power from each of them decreases and the total TCAS interrogation rate for any nearby transponder remains less than 280 per second. If the victim is ta[ken] 35 μs suppression or reply dead time whenever it receives a TCAS interrogation, the total "off" time caused by TCAS interrogations will then never exceed one percent. The second term on the right-hand side assures that an individual TCAS unit never transmits more average power than it would if there were approximately 15 other TCAS units nearby $(280/(1+15) \approx 18)$.

2. Inequality 2 assures that the transponder on board the TCAS aircraft will not be turned off by mutual suppression signals from the TCAS unit on the same aircraft more than one percent of the time. For $i=1..I$,

$$M_i \leq 0.01 second \qquad (2.2)$$

3. Inequality 3 assures that a victim ATCRBS transponder will not generate more than 40 ATCRBS replies in a one-second period in response to interrogations from all the TCAS interrogators within its detection range. For $k=1..K$,

$$\frac{1}{B} \times \frac{PA_k}{250\ watt} \leq min[\frac{80}{1 + NTA}, 5] \qquad (2.3)$$

Like inequality 1, this inequality includes terms to account for reduced transmit power, to account for the other TCAS interrogations in the vicinity, and to limit the power of a single TCAS unit. Forty ATCRBS replies per second is approximately 20 percent of the reply rate for a transponder operating without TCAS in a busy area of multiple ATCRBS ground sensor coverage.

Example: Whisper interrogation limiting is not invoked. The ATCRBS and Mode S interrogation rates are as follows: The ATCRBS interrogation rate is constant at 26 top plus four bottom whisper-shout interrogations per second. The sum of the normalized whisper-shout powers, i.e., the ATCRBS contribution to the left-hand side of inequality 1, is 4.89. The Mode S interrogation rate depends on the number of Mode S aircraft in the vicinity. In en-route airspace, it is typically an average of about 0.8 interrogations per second for each Mode S aircraft within 30 nmi. In a uniform aircraft density of 0.02 aircraft per square nautical mile, the number of aircraft in 30 nmi is 57. If 20 percent of these are TCAS equipped, N_a=12 and the variable term on the right-hand side of inequality 1 is 1.5. If the number of TCAS aircraft in the area does not exceed 15, the fixed term continues to govern and no limiting occurs until there are approximately 160 Mode S aircraft within 30 nmi.

Similar considerations hold for inequalities 2 and 3. In inequality 2, the mutual suppression interval associated with each top-antenna interrogation is 70 μs. The bottom-antenna mutual suppression interval is 90 μs. Thus the ATCRBS contribution to the left-hand side of inequality 2 is 0.023. The Mode S interrogation rate can be as high as 109 top antenna interrogations per second before violating the limit. With the full whisper-shout sequence, the left-hand side of inequality 3 is 4.89. The number of TCAS aircraft within 30 nmi can be as high as 15 without violating inequality 3.

When the interrogation rate or density increases to the point at which one or more of the limits is violated, either the ATCRBS or Mode S normalized interrogation rate or both must be reduced to satisfy the inequality. If the density were to reach 0.1 aircraft per nmi² uniformly out to 30 nmi, there would be 283 aircraft within a 30-nmi radius. If 10 percent of these were equipped with TCAS, N_a=28. The right-hand limits in inequalities 1 and 3 would then be 0.65 and 3.6, respectively. To satisfy these limits, the ATCRBS and Mode S contributions to the left-hand side of inequality 1 would both have to be reduced. As a result, the surveillance range of both ATCRBS and Mode S would be less.

[2.14]

Mode S Electromagnetic Compatibility

Modes used by TCAS provide compatibility with the Mode S based secondary surveillance radar system. The frequency spectrum of transmissions is controlled to protect adjacent distance-measuring equipment (DME) channels (↑SC 2.1.3).

Tracking Logic

[2.15]
Using surveillance reports it receives each second, the Collision Avoidance System (CAS) logic tracks slant range and closing speed of each target aircraft to determine the time in seconds to the closest point of approach (CPA).

[2.16]
If the target is equipped with an altitude-coding transponder, the CAS logic can also project the altitude of the target at the CPA.

[2.17]
The CAS determines the vertical rate of the target by measuring the time it takes to traverse successive 100-foot increments of altitude (↑H05).

[2.17.1]
Altitude reports of other aircraft are tracked using a level-occupancy-time algorithm. This function derives altitude and altitude rate by tracking the time intervals between changes of the Mode C altitude reports, which are quantized to 100 ft., and the altitude reports in Mode S replies, which may be quantized to 25 ft or 100 ft. The tracker also performs tests on the other aircraft's altitude rate. This test classifies the rate as being high or low confidence (also called *firmness*) and determines a range within which the rate must fall to allow later selection of escape direction. During periods when this rate is considered uncertain ("low confidence"), certain logic functions may be inhibited.

[2.17.2]
The use of the confidence in determining when to declare a threat enables valuable information to be used without excessive delays (↑SC4.) For instance, the vertical trend may be firm [illegible] may be unclear whether an intruder is level or climbing but it may be quite clear that it is not descending. A safe and timely DESCEND

RA may well be given. Therefore, TCAS estimates reasonable inner and outer limits on the intruder vertical rate. Therefore, confidence on monitoring separation against any possible altitude crossing yields ALIM intruder rate within the rate limits will be given without delay. Almost always established (except that "intruder climbing" or "intruder descending" can be ruled out.

[2.17.3]
By comparing the onboard radar altimeter input to own barometric altitude, TCAS makes an estimate of the barometric altitude of the ground and can identify aircraft on the ground track. Resolution advisories are inhibited for such traffic.

[2.18]
TCAS uses data from its own aircraft pressure altimeter, either directly from the altitude encoder or as processed by the air data computer, to determine its own aircraft altitude, its vertical rate, and the relative altitude of each target. The data provided by tracking (target range and range rate, relative altitude, and vertical rate) are used to generate traffic advisories and to perform threat detection.

[2.19]
When below 1700 feet, the TCAS logic uses the difference between its own aircraft pressure altitude and radar altitude to determine the approximate elevation of the ground above sea level (see Figure 5). It then subtracts the latter value from the pressure altitude value received from the target to determine the approximate altitude of the target above the ground (barometric altitude − radar altitude + 180 feet). If this altitude is less than 180 feet, TCAS considers the target to be on the ground (↑SC4.9). Traffic and resolution advisories are inhibited for any intruder whose tracked altitude is below this estimate. Hysteresis is provided to reduce vacillations in the display of traffic advisories that might result from hilly terrain (↑FTA-320). All RAs are inhibited when own TCAS is within 500 feet of the ground (↑SC4.6).

Traffic Advisories

[2.20]
TCAS generates traffic advisories against nearby aircraft that may later require a resolution advisory. The TA provides range, altitude rate, and bearing information for the intruder. The criteria for generating TA's have been chosen to allow the pilot time to gain visual acquisition before an RA becomes

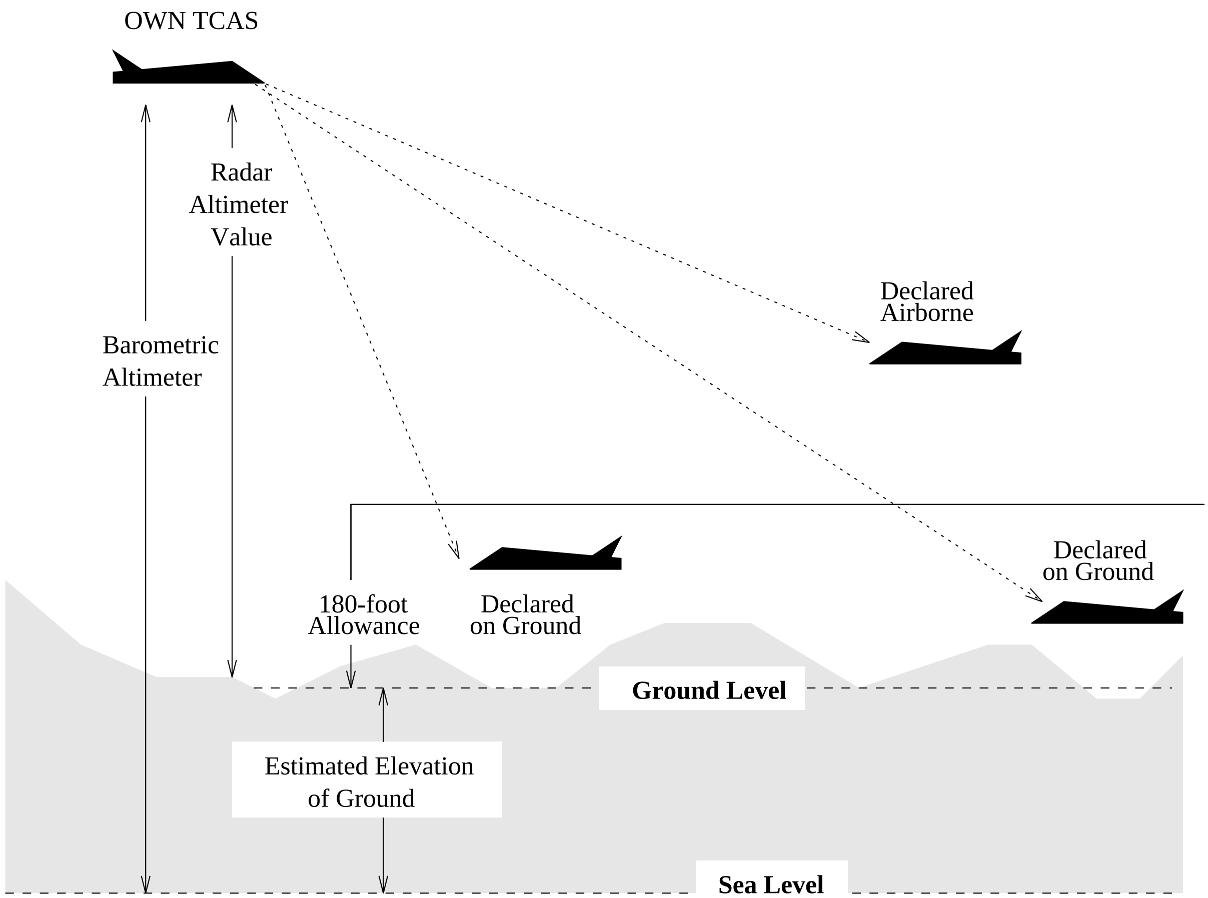


Figure: Determining Aircraft is on the Ground

TCAS selects (i.e., necessary) (↑SC4.7) intruder aircraft for generating traffic advisories by using algorithms analogous to those used for threat detection, but with larger thresholds.

[2.21]

TAs on altitude-reporting and nonaltitude-reporting intruders are maintained for a minimum of eight seconds, even though the criteria for a TA may no longer be met. Pilots do not like target information to flash on and off before they have a chance to read it (↑SC11.3). However, the duration of the TA is dependent on surveillance continuing to pass data on the intruder each cycle and may be shorter than eight seconds if surveillance drops track on the intruder aircraft.

[2.22]

Traffic advisories are of two classes: *proximity advisories* (governed by range and altitude thresholds) and *threat advisories* (governed by tau criteria). All TAs are generated each cycle and ranked in order of severity so that the most severe will receive priority in case of display limitation (↑SC4.8, FTA-375, FTA-735)

[2.22.1]

The logic classifies traffic advisories with three levels of priority:

1. The highest priority is given to a threat causing a resolution advisory to be generated. If the aircraft is equipped with a traffic advisory display, all targets causing resolution advisories are displayed as traffic advisories, even if there is no bearing estimate available. There is no ranking within this group.
2. The next priority is given to other threats, including targets without altitude reports. Within this group, altitude-reporting threats have highest priority.
3. The lowest priority is given to traffic satisfying a proximity test but not a *tau*-based detection test. The latter type is only displayed when a higher priority type is also being displayed. Within each group, those with small range get highest priority, followed by converging intruders, followed by diverging intruders. Again, altitude-reporting intruders are given higher priority.

[2.23]

TCAS selects intruder aircraft for generating traffic advisories by using algorithms analogous to those used in threat detection but with larger thresholds. The traffic advisory normally precedes the resolution advisory in time.

[2.23.1]

If an intruder is receiving an RA, it immediately qualifies for a TA.

. 2 If it is on the ground it does not receive a TA (→ 19.) Otherwise, thresholds are determined using own sensitivity level alone or the larger of own and intruder sensitivity level if the intruder is TCAS-equipped.

[2.23.1.1]

Altitude Reporting Targets: Range and altitude tests are performed on each altitude-reporting target. The range and altitude test must be passed for a traffic advisory to be generated. The range test is based on the RA *tau* plus approximately 15 seconds (↑1.20.2). In addition, the current or projected vertical separation at the CPA must be within 1200 feet for a target to be declared an intruder, generating a TA. A proximity TA target is displayed if the target does not qualify for a TA but is currently within 6 nmi and 1200 feet of the TCAS aircraft (↑1.20.2 3).

[2.23.1.2]

Nonaltitude Reporting Targets: A nonaltitude reporting target is declared an intruder if the range test alone shows that the calculated *tau* is within the RA *tau* threshold associated with the current SL being used (i.e., the threshold is set to values that are similar to those used for RAs from 20 seconds for SL-4 to 35 seconds for SL-7. When TCAS is in SL-2, the *tau* threshold is 15 seconds. The range *tau* threshold for RAs is used in order to restrict the number of non-Mode C traffic advisories, given the absence of altitude information, which for Mode C intruders acts as an additional discriminant.

Resolution Advisories

[2.24]

If an intruder is determined to be a threat to the TCAS-equipped aircraft, then a vertical resolution advisory is provided to the pilot. TCAS also communicates its resolution advisory to a ground Mode S sensor by means of a Resolution Advisories Report when so requested by the sensor.

[2.24.1]

No resolution advisories are chosen for intruders not reporting altitude, although they may generate traffic advisories (↑1.20.3, Classifications).

[2.24.2]
(↑SC1.6) RAs are inhibited below 5

[2.24.3]
Generally, a threat aircraft will result in a collision warning in advance of the closest approach by the amount of the prediction time (20-35 seconds). Additional time is provided for accelerating encounters (high closure rates) by means of an incremental protected volume (see DMOD) around own aircraft.

[2.25]
The resolution advisory has both a sense (upward or downward) and a strength (vertical rate) and is either positive or negative.

[2.25.1]
A *positive* advisory instructs the pilot to deviate from current vertical rate.

[2.25.2]
A *negative* advisory instructs the pilot to avoid certain deviations from current vertical rate.

[2.26]
The logic indicates whether a displayed advisory requires corrective action by the pilot or merely warns against initiating an action that could lead to impaired vertical separation. (↓Composite-Control$_{s\text{-}227}$)

[2.27]
Resolution advisories against all threats detected by own TCAS are combined for pilot display such that the most demanding resolution advisory of those active is displayed for each sense (up and down). (↓Composite-RA$_{s\text{-}266}$)

[2.28]
After the CPA is passed and the range between the TCAS aircraft and threat aircraft begins to increase, the resolution advisory is cancelled and pilots then return to their assigned altitude or to their original vertical rate (if climbing or descending).

[2.28.1]
In order to avoid overly brief displays of resolution advisories (↑SC11.3), TCAS forces an intruder, once declared a threat or after a RA changes, to remain a threat for at least five cycles (approximately five seconds). (↓No-Weaken$_{m\text{-}537}$)

Threat Detection

[2.29]
The threat detection criteria consist of a range test and an altitude test. Both must be satisfied to declare the intruder a threat. Also, the estimate of the intruder's vertical rate must either be of high confidence or be judged safe despite a range of uncertainty. Otherwise, the intruder is not declared a threat. (↓Threat-Condition$_{m\text{-}373}$)

[2.30]
Range and altitude tests are performed on each altitude-reporting intruder. If the RA *tau* and either the time to coaltitude or relative altitude criteria associated with the current SL are met, then the intruder is declared a threat. These boundaries are chosen to make TCAS resistant to abnormal deviations from separation typically maintained by air traffic control or procedurally by flight rules (↑C.4, SC3).

[2.30.1]
Range Test: To be declared a threat, an aircraft's expected altitude separation at the point of closest approach (ZTHR, which is either 750 or 850 feet) must be less than an altitude threshold. (↓Threat-Alt-Test$_{m\text{-}371}$)

[2.30.2]
Range Test: The normal case involves an intruder that is converging in range. In this case, the criterion is simply that the time to CPA (called *tau*) is computed as the range divided by range rate. A *modified tau* is computed in the same way except that the quantity *DMOD*^{2}/*range* is subtracted from the range. The modified *tau* is used as the range test criterion to determine whether the range criterion is met. This formula is used in the detection logic to account for the fact that lateral accelerations are much more likely than accelerations in the direction of the relative velocity vector. The effect of the modification to tau is made dependent upon the range and range rate of the threat. For slow closure encounters that are close in range, as in parallel flight, *tau* may not be high until the aircraft are dangerously close (↑SC4.6). In these cases, subtracting *DMOD*^{2}/*range* from the range causes the *modified tau* to be substantially smaller, triggering the alert when the threat is at a greater (safer) range. As range and range rate increase, the effect of subtracting *DMOD*^{2}/*range* decreases such that the range

tau is nearly equal to the true tau (range/range rate) (↓ Modified-Tau-Uncapped$_{i-402}$, ↓ Modified-Tau-Uncapped$_{i-424}$)

[2.30.3]
A special situation occurs when the range is virtually unchanging or diverging, because time to closest point of approach may be undefined or inapplicable. In these cases, the modified tau is arbitrarily set to zero. The range test is considered passed only if the intruder is within the threat volume and the range/range-rate product is small. (↓ Modified-Tau-Uncapped$_{i-402}$)

[2.31]
Depending on the geometry of the encounter and the quality of the vertical track data, an RA may be delayed or not selected at all.

[2.32]
Several of the thresholds used in the pure geometrical criteria for an intruder to become a threat vary with the intruder's equipage and sensitivity level. Higher sensitivity levels imply higher altitude flight (in nonterminal areas), which implies larger altimetry errors and sparser traffic. The detection thresholds are therefore made larger to help overcome the altimetry errors (↑SC5) and the sparser traffic (↑SC4), to reduce the number of unwanted alerts (↑SC5). 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 the number of unwanted alerts (↑SC5). For example, near a terminal, the intruder may be in a landing pattern but momentarily appear to be on collision course with own aircraft. The longer TCAS waits, the more likely it is that an unwanted alert will be avoided. (↓ Threat-Alt-Test$_{m-371}$, ↓ Threat-Range-Test$_{m-374}$)

[2.33]
If the intruder is equipped with TCAS, it is desirable that both aircraft use the same thresholds against each other, even if they are in different sensitivity levels. Hence, the parameters are set according to the larger of own and intruder's sensitivity level. The capability is reserved to set certain thresholds differently for an equipped intruder. (↓ Other-SL$_{i-385}$)

[2.34]
Once an aircraft has been declared a threat and causes a resolution advisory to be displayed, it will remain classified as a threat

feet. And diverge in range, even though the altitude separation exceeds
aircraft, regardless of the range test, five seconds, threat for at least
(Classification s-275) decision, to avoid overly brief displays of advisories. (

Delays:

[2.35]
Under some circumstances, TCAS may delay resolutions advisories even when they satisfy both the range and altitude tests. The two reasons for the delay are (1) to avoid an altitude-crossing resolution advisory and (2) to allow a TA with (temporary ... data) (↑FTA -480.)
In all cases, a TA will still be given while the RA is delayed (↑FTA-350, FTA-480) (RA -Display-Delay$_{m-344}$)

Biasing Against Altitude Crossing RAs

[2.36]
A *Biasing Against Altitude Crossing RAs:* TCAS aircraft that is in conflict with an intruder that is projected to cross its path is disposed to select an altitude-crossing resolution advisory because that advisory is projected to provide the greater separation. This, however, leaves the TCAS aircraft susceptible to a late level-off maneuver by the intruder (↑FTA-1150)
To bias strongly against such susceptibility, tests are incorporated to delay threat declaration, with the knowledge that the threat may level off, or, if TCAS equipped, choose a noncrossing resolution advisory itself (↑SC7. 1). However, if the crossing geometry persists despite the delays, an altitude-crossing resolution advisory will be issued, albeit later in the encounter.

[2.36.1]
As a means of reducing altitude crossings in TCAS/TCAS conflicts, TCAS has the level TCAS aircraft defer threat declaration up to three seconds against a nonlevel intruder that is projected to cross altitude if the intruder has not yet sent its intent to the level TCAS aircraft. This tends usually to force the nonlevel aircraft to choose first; its choice will be the noncrossing maneuver made all the more probable by other noncrossing-biasing logic. Because the level TCAS aircraft will then select the sense complementary to the one already chosen by the nonlevel TCAS aircraft, an altitude crossing will generally be avoided. In the event that the nonlevel aircraft does not send its intent within the three-second waiting period, the level aircraft will choose its own best sense, still possibly avoiding an altitude crossing if A feet separation can be achieved in the noncrossing direction.

[2.36.2]
A bias against altitude crossing RAs is also used in situations involving intruder level-offs at least 600 feet above or below the TCAS aircraft. In such a situation, an altitude-crossing advisory is deferred if an intruder aircraft that is projected to cross own aircraft's altitude is more than 600 feet away vertically. (↓Alt-Separation-Test$_{m-351}$)

Assumption: In most cases, the intruder will begin a level-off maneuver when it is more than 600 feet away and so should have most of its greatly reduced vertical rate by the time it is within 200 feet of its altitude (thereby either not requiring an RA if it levels off more than 600 feet away or requiring a non-crossing advisory if it levels off before the 600 feet threshold is reached.)

For those intruders still projected to cross the TCAS aircraft's altitude by the time they are within 600 feet, an altitude-crossing RA is issued.

[2.36.3]
TCAS also avoids issuing altitude-crossing RAs under conditions of low confidence in the intruder's vertical rate (track firmness) (↑FTA-480). Normally, TCAS will delay sense selection until there is high confidence in the intruder's vertical rate. This delay is bypassed, however, if projected safe separation is poor under the most conservative bounding estimate of vertical rate. As a result, in certain situations, an altitude-crossing RA could be issued. (↓Other-Track-Firmness$_{f-431}$)

For small vertical separations, crossing under these circumstances is not a concern, however, flight tests and operational evaluations disclosed that altitude-crossing advisories could be issued in situations where there was substantical vertical separation. TCAS therefore prevents selection of an altitude-crossing advisory on low-confidence unless the two aircraft are within 150 feet of each other vertically, a distance too small for the crossing status to be of concern (given the high degree of uncertainty in the intruder's vertical rate). Otherwise, RA selection will be delayed pending a better estimate of the intruder's track vertical rate. Such delays are usually less than three seconds.

[2.37]
Biasing against altitude-crossing RAs is an effective means for reducing the susceptibility of a sudden intruder level-off maneuver. For those situations

in which the noncrossing sense will not provide adequate separation. However, the crossing RA is issued and annunciated in such as way as to alert the pilot to the potential that the TCAS resolution could be thwarted by a sudden adverse maneuver by the intruder. In most encounter situations, the initial resolution advisory sense will be maintained for the duration of an encounter with a threat aircraft. However, if the intruder does maneuver, TCAS recognizes the situation as it develops and reverses the RA sense. This is immediately displayed and annunciated, giving the pilot time to react with an expedited, though not violent, maneuver (↑C5, OP.11, SC9 ↓ Noncrossing-Biased-Climb$_{m\text{-}338}$)

Inhibits:

[2.38]
INCREASE-CLIMB RAs by TCAS II may need to be inhibited because of inadequate aircraft climb performance (↑L.6, SC9.1). The collision avoidance maneuvers posted as RAs by TCAS II assume an aircraft's ability to safely achieve them. If it is likely they are beyond the capability of the aircraft, then TCAS II must know beforehand so it can change its strategy and issue an alternative RA. These performance limits must be provided to TCAS II from the aircraft interface and discretes (↑Page relative to altitude/configuration). However, the need to inhibit TCAS II INCREASE-CLIMB RAs should be carefully considered because the alternative RAs will not provide the optimum solution to the encounter. Inhibiting these RAs will increase the likelihood of TCAS II (a) issuing crossing maneuvers (crossing through an intruder's altitude), which in turn increases the possibility that an RA may be thwarted by the intruder maneuvering (↑SC7.1, FTA-1150), (b) causing an increase in DESCEND RAs at low altitude (↑SC8.1), and (c) providing no RAs if below the descend inhibit or 1200 feet AGL, 1000 feet on approach.

[2.39]
Because of the limited number of inputs to TCAS for airplane performance inhibits, in some instances where inhibiting RAs would be appropriate it is not possible to do so (↑L.6). In these cases, TCAS may command maneuvers that may significantly reduce stall margins or result in stall warning (↑SC9.1). Conditions where this may occur include bank angle (wings level is assumed), weight, altitude, temperature combinations, and initial speeds outside the envelope demonstrated as within performance limits for this particular RA aircraft configuration, engine out, fixed configurations, such as approach flap and landing gear not retracted. The

)Aircraft flight manual supplement (AF
flight manual provide information concerning this aspect of TCAS so that
↓ take appropriate action (3.P.M.), 4.AF

[2.40]

An aircraft's climb capability when operating at or near its maximum approved operating altitude is also affected by excess thrust and true airspeed available to safely trade if needed for climb rate.

MR RAs should not be inhibited if the aircraft has adequate performance because it may exceed its maximum certificated altitude by several hundred feet during an RA. If the aircraft is approved for significant alternate configurations,

e.g., the initial airspeed used for the analysis should be appropriate for them spare engine pod, gear down operation, etc. In the analysis of the aircraft's ability to accelerate and return to the initial speed and altitude following an RA, an overshoot of approximately 200 feet is permissible.

[2.41]

MRA If a maneuver causes airspeeds below the minimum demonstrated as within performance limits for this particular aircraft, then the (NF I Climb) RA should be inhibited (↑SC Increase-Climb-Inhibit$_{s-243}$) -
Climb-Inhibit$_{v-244}$)

[2.42]

Radio altitude information is used to inhibit DESCEND RAs below 1200 feet and Increase Descend RAs below 1450 feet on Approach, I DESCEND resolution advisories below 1000 feet (↑SC). These inhibits allow automatic sensitivity level selection when close to the ground and determination of whether individual targets are on the ground (↑ Page -Descend-Inhibit
(Increase-Inhibit$_{s-245}$) -Descend-Inhibit$_{s-246}$)

Nuisance Alarm Filter (↑SC5)

[2.43]

Observations over a period of time have shown that there is one class of unnecessary TCAS alarms that can be particularly disconcerting (↑SC5.)
These are alarms that occur almost exactly at the time of closest approach. They almost always have a substantial miss distance, occur well off to the side of the aircraft, and would last for only one or two seconds, except that they are maintained for the minimum display period, usually five seconds.
These encounters usually trip the range *tau* threshold early, but the alarm is finally issued when the vertical altitude test passes, often only at the very last.
↓Tau is issued, but often near the time of closest approach. (-Rising$_{m-550}$)

lter is used. To eliminate such nuisance alarms, a logic filter is used. The
on the fact that the range *tau* goes through a minimum and finally rises
just before the aircraft reach the CPA. (Once *tau* goes below the range *tau*
) threshold, however, alerts are capped to prevent fluctuations of the alerts.)
The filter eliminates alerts against intruders whose range *tau* has begun to
rise when they are more than 1.5 nmi away but that later violate the altitude
threshold. As long as the range *tau* continues to rise, the intruder will not
qualify for an RA. If, however, the range *tau* subsequently decreased, the intruder would again immediately
qualify for an RA.

Vertical divergence

[2.44]
The aircraft may be close enough in relative altitude to warrant an advisory
(i.e., the projected miss distance at CPA
.) The test for divergence (↑SC
and that relative altitude is within
the absolute value of ((relative altitude) + relative altitude rate
) exceeds, in which case, increasing vertical separation at CPA is
↓(Inhibit -Chec $_{m-323}$)

Advisory Sense Selection

[2.45]
The sense of an RA may be upward (CLIMB) or downward (DESCEND)
(↓Sense $_{s-273}$)

[2.45.1]
The climb sense is used when the TCAS aircraft is to pass above the
threat and results in a resolution advisory such as
SCLIMB / DON'T DESCEND

[2.45.2]
The descend sense is used when TCAS is to pass below the threat.

[2.46]
All descend RAs are inhibited below 1000 ft so that the aircraft flight pro-
file will not fall below the standard glide path (↑SC8.2, ↓Descend-Inhibit $_{s-245}$)

[2.47]
Advisory sense selection may be inhibited during an interval in which there
is low confidence in the threat's vertical rate (→ 17.1, 2.36.3). However, when

… one sense gives adequate separation against an intruder having a vertical rate falling anywhere in the confidence interval, that sense is selected.

[2.48]
Selection of an altitude-crossing sense is not permitted unless the two aircraft are close in altitude such that the advised maneuver would generate adequate separation (↑SC7.1). If the two aircraft are not close in altitude, sense selection is inhibited. (↓Crossing s-280)

Sense Selection Logic:

[2.49]
Based on the range and altitude tracks of the intruder, the intruder's path is modeled to the CPA. The predicted vertical separation is computed for each of the two senses and the sense is chosen that provides the greater vertical separation.

[2.49.1]
Path Modeling: TCAS, using range-altitude coordinates, models the path of the threat aircraft based on the track data of the threat. TCAS also models for itself the path that would result if TCAS maneuvered to a climb and that which would result if TCAS maneuvered to a descent. The path model in all cases consists of a linear segment, followed by a constant-acceleration altitude rate change, followed by a linear segment. The linear escape rate modeled is nominally 1500 fpm. The model will use a higher rate for one direction when current performance exceeds the computed rate in that direction or a lower rate when indicated by current resolution advisories. (↓Climb-Inhibit s-243, ↓ Descend-Inhibit s-245)

In certain flight regimes, TCAS may determine that the TCAS aircraft cannot execute the nominal climb escape maneuver. This determination may be made according to the unique characteristics of the aircraft on which TCAS is installed (aircraft climb performance may be limited at high altitude) and flight configuration (e.g., landing configuration) as indicated by inputs not otherwise required for use on TCAS. Path modeling will substitute level flight for the climb maneuver in such instances.

Similarly, radar altimeter data is used to recognize when TCAS is too near to the ground to issue a descend advisory (↑SC8.1). Path modeling recognizes this condition and substitutes level flight at the appropriate altitude.

[2.4.2] **Computing Vertical Separation:** TCAS computes the predicted vertical separation for each of the two sense selection options. The directional sense that gives the greater predicted vertical separation from the threat aircraft at the time of closest approach is selected as the preferred sense of the resolution advisories caused by this threat.

In the following cases, due to additional considerations, the preferred sense may not be the direction that gives the greatest modeled separation

1. Against a TCAS that has already selected its sense, own TCAS will select a complementary sense choice. (↓Sense s-275)
2. Against an unequipped intruder, when own aircraft is in a multi-aircraft situation and the two sense choices give adequate separation, own aircraft will select the same sense selected due to another simultaneous threat. (↓Sense s-275)
3. Against any intruder, when the non-altitude crossing sense is chosen because that sense will result in at least ALIM ft separation at closest approach. In those cases where an altitude crossing by the threat or TCAS aircraft is projected, TCAS will select the sense that avoids crossing if the desired amount of vertical separation can be maintained at the CPA (↑SC7.1). If ALIM cannot be achieved, a crossing RA will be issued. (↓Noncrossing-Biased-Climb m-338, ↓Noncrossing-Biased-Descend m-339)

[2.4.5] **Don't Care Test:** When TCAS is displaying an RA against one threat and then attempts to choose a sense against a second threat, it is often desirable to choose the same sense against it as was chosen against the first threat, even if this sense is not optimal for the new threat. One advantage is display continuity (↑SC11.3). Another advantage is that the pilot may maneuver more sharply to increase separation against both threats. If a dual sense advisory is given, such as DON'T CLIMB AND DON'T DESCEND, a vertical maneuver to increase separation against one threat reduces separation against the other threat. The most important advantage, however, is to avoid sacrificing separation inappropriately against the first threat in order to gain a marginal advantage against the second threat. (↓Don't-Care-Test m-357)

The don't-care test determines the relative advantages of optimizing the sense against the new threat versus selecting the same sense for both threats. When the former outweighs the latter, the threat is called a do-care threat; otherwise, the threat is a don't-care threat.

Sense Reversals

[2.51]
In most encounter situations, the resolution advisory sense, selected according to the principles described above, will be maintained for the duration of an encounter with a threat aircraft. However, under certain circumstances, it may be necessary for that sense to be reversed (↑SC7.2) For example, a conflict between two TCAS-equipped aircraft will, with very high probability, result in selection of complementary advisory senses because of the coordination protocol between the two aircraft. However, if coordination communications between the two aircraft are disrupted at a critical time of sense selection, both aircraft may choose their advisories independently (↑FTA-1300) This could possibly result in selection of incompatible senses (↑FTA-395 (Reversal-Provides-More-Separation_m301))

[2.51.1]
Should a sense incompatibility be discovered by the TCAS logic, the logic will rectify the situation by having the TCAS with the higher Mode S ID reverse its sense. The TCAS with the lower Mode S ID retains its original sense.

[2.51.2]
Similarly, in a conflict with a non-TCAS-equipped threat aircraft, sense reversals will be considered if an altitude-crossing advisory has been chosen. This is because the threat could level-off after the advisory has been issued, thereby thwarting effective resolution of the original advisory.

[2.51.3]
The resolution logic monitors the progress of the crossing maneuver. If it detects a substantial change in the encounter geometry, indicating that the intruder has begun to level off, it models pilot response to the reversed-sense advisory. If the predicted result of that response exceeds the maximum predicted altitude of the intruder at closest point of approach (using the maximum rate bound), the advisory sense is reversed.

[2.52]
RA sense reversals are prevented (operation of the reversal logic is inhibited):

1. If an increase rate RA has been issued.
2. If the TCAS and intruder aircraft are about to diverge.
3. If the pilot of the TCAS aircraft would have insufficient time to respond to the RA before CPA is reached (four seconds or less) (↑SC7.3).
4. If TCAS and intruder aircraft are separated by less than 200 feet vertically when 10 seconds or less remain at closest approach (↑SC7.3).

(↓Reversal-Provides-More-Separation $_{m\text{-}301}$)

[2.53]
In all cases, after the decision to reverse sense is made, normal advisory strength selection occurs. (↓RA-Strength $_{s\text{-}277}$)

[2.54]
Whenever the sense of a displayed advisory is reversed, the old advisory is removed, the new advisory is posted, and the reversal is annunciated. However, if a reversal occurs before display of the RA in an encounter with a TCAS-equipped aircraft, no annunciation of the reversal is necessary. (↓Vertical-Control $_{s\text{-}231}$) (↓Sense [illegible])

Advisory Selection (Types)

[2.55]
The types of advisories that TCAS can generate are:

- [illegible] 2500
- [illegible]ENT RAT[illegible] 2500
- M[illegible]L
- M[illegible]EN
- CL[illegible]B
- DESCEND
- DON'T [illegible]L
- DON'T [illegible]SCEND
- DON'T [illegible] 500
- DON'T [illegible] 1000
- DON'T [illegible] 2000
- DON'T [illegible]N 500
- DON'T [illegible]N 1000
- DON'T [illegible]N 2000

(↓Vertical-Control $_{s\text{-}231}$) (↓Climb-[illegible]) (↓Descend-RA $_{s\text{-}233}$) (↓RA $_{s\text{-}235}$) (↓RA-Strength $_{s\text{-}277}$)

[2.5]
Selecting a Vertical Rate: The collision avoidance logic selects the least disruptive vertical rate maneuver that will still achieve safe separation (↑C5.) Advisory strength is continuously evaluated, and modified if necessary, during the course of the encounter. (↓RA-Strength$_{s-277}$)

[2.5.1]
TCAS will generate a positive advisory only if the threat is projected to be within the positive advisory altitude threshold (ALIM) at the time of CPA. This threshold value is smaller than the altitude boundary used in threat declaration and is derived from the measure of altitude separation that is considered safe at the time of closest approach. (↓Dont-Care-Test$_{m-357}$, ↓Wea-Test$_{m-363}$, ↓Positive-VSL-Test$_{m-345}$, VSL-OK$_{m-378}$)

[2.5.2]
TCAS will select the least disruptive Vertical Speed Limit (VSL) or negative advisory if one of these types achieves safe separation. Otherwise, a positive advisory is generated. If safe separation depends on own aircraft maintaining an existing vertical rate, a maintain will be given. The rate used by the modeling program will also be provided. (↓RA-Strength$_{s-277}$)

[2.5.3]
TCAS recomputes the selection of an advisory on every processing cycle (approximately once per second). The sense or type of advisory may change as the geometry of the conflict changes. Display changes to a less restrictive advisory are delayed for several seconds (↑SC7.2) Advisory recomputation is inhibited when TCAS has low confidence in the threat's vertical rate. (↓Reconsider-Advisory$_{s-417}$)

[2.5.4]
In modeling positive advisories (↓RA-Strength$_{s-277}$), TCAS assumes that the pilot can achieve the modeled rate (either 1500 fpm or the current rate if higher than 1500 fpm) in the appropriate direction in response to the advisory. Logic modeling of potential aircraft maneuvers accounts for acceleration and a delay for pilot observation and reaction to the advisory display. The pilot can optionally increase the margin of safety by increasing the response. TCAS never assumes that the aircraft can climb or descend faster than the modeled rate unless that aircraft's vertical rate shows it is already doing so. TCAS does not display climb when it has been determined that TCAS aircraft cannot climb unless a sense

↓Climb quired. (reversal to a climb advisory is re -Inhibit$_{s-243}$)

[2.57]
Increasing Rate: Vertical If TCAS detects that an intruder aircraft has increased its vertical rate toward the TCAS aircraft, such that the TCAS aircraft executing its nominal escape maneuver will not be able to obtain sufficient vertical separation, TCAS displays an advisory to increase the vertical rate from 1500 **fpm** to **2500** ↓Increase in the existing sense. (-Check$_{m-323}$)

[2.57.1]
An advisory to increase vertical rate is permissible if the current RA is non-crossing or if sufficient time remains in a crossing situation for successful execution of a sense reversal. If the intruder is a TCAS aircraft, an increase rate RA is permitted whether or not the encounter involves crossing altitudes.

[2.57.2]
Increase rate climb RAs are inhibited when own aircraft cannot achieve either 1500 **fpm** or the new rate. (↓Increase -Climb-Inhibit$_{s-244}$)

[2.57.3]
Increase rate descend RAS are inhibited when the TCAS aircraft is within 1450 feet of the ground to avoid interaction with the Ground Proximity Warning System (↑SC 10.1, 1.3, ↓Increase -Descend-Inhibit$_{s-246}$)

[2.57.4]
RAs to increase vertical rate are displayed for a minimum of 10 seconds to provide continuity in advisory presentation (↑SC11.3) when the de-) and typically after one second (fl ... criteria are satis ... to ensure that some additional vertical separation will be provided over that of the current 1500 **fpm** RA RA. (↓ -Strength$_{s-277}$)

[2.57.5]
If a threat aircraft maneuvers vertically in a way that thwarts a TCAS aircraft's RA, then

[2.57.5.1]
If the threat is not equipped with TCAS, own aircraft will be advised either to increase its vertical rate from 1500 **fpm** to **2500 fpm** or to reverse sense. (↓Strength$_{s-277}$, ↓Reversal -Provides-More-Separation$_{m-301}$)

[2.57.5.2]
If the threat is equipped with TCAS II, an unexpected vertical

maneuver is handled only by an increase-vertical-rate advisory, as sense-reversal advisories are not permitted in this case. (↓RA-Strength$_{s-277}$, Reversal-Provides-More-Separation$_{m-311}$)

Multi-Aircraft Situations

[2.58]

When TCAS faces more than one threat, special logic is required. TCAS may elect to provide separation either above or below a single threat by means of sense selection. Against two or more threats, TCAS has three options: provide separation (1) above all (by climbing), (2) below all (by descending), or (3) above some and below the others (by leveling off). TCAS selects the option that maximizes the minimum modeled separation over all threats. (↓RA-Strength$_{s-277}$, ↓Threats-Above-And-Below$_{m-330}$)

[2.59]

If multiple aircraft are involved in a TCAS encounter, then the following priority is used to select an RA:

1. TCAS attempts to resolve the situation with a single RA, if that will maintain safe separation from each of the threat aircraft.
2. If a single RA cannot resolve the situation, TCAS selects an RA that is a composite of noncontradictory climb and descend restrictions.
3. In the extreme case, TCAS tells the pilot to maintain current altitude.

(↓RA-Strength$_{s-277}$)

TCAS/TCAS Coordination

[2.60]

The purpose of coordination is to ensure that compatible (opposite) senses are selected in encounters between two TCAS-equipped threats. Coordination interrogations contain information about an aircraft's intended vertical maneuver or "intent" with respect to the threat. (↓Coordination-Update$_{a-15}$)

[2.61]

In a TCAS/TCAS encounter, each aircraft transmits interrogations to the other via the Mode S link to ensure the selection of complementary resolution advisories.

[2.62]
Coordination interrogations use the same 1030/1090 MHz channels as surveil-lance interrogations and are transmitted once per second by each aircraft for the duration of the RA, i.e., each aircraft transmits a coordination interrogation to the other once per second as long as the other aircraft is causing an RA.

[2.63]
To increase the probability of success in coordination with TCAS-equipped threats, a validity check is imposed on all range reports received for any intruder (TCAS-equipped or otherwise). This is done by an algorithm that determines if two out of the last three surveillance range reports, including the current report, are valid updates (i.e., not coasted). If this is the case, declaration of a TCAS-equipped intruder as a threat can occur. Otherwise, threat declaration will be deferred pending better surveillance range report. The reasons for this are (1) if range surveillance of a TCAS-equipped intruder is poor, there is also a high probability that the interfering condition could cause a missed TCAS-TCAS coordination on the current cycle, and (2) if the surveillance logic is coasting the other TCAS aircraft Mode S track at the current time, it would not be desirable to issue an RA against an intruder that may be dropped a few seconds later. Therefore, an RA against a TCAS-equipped aircraft can only be issued if it has already sent its Resolution Advisory complement in a Resolution message (intent) or if the report is judged to be valid. (↓Two-Of-Three $_{m370}$)

[2.64]
The communication is in the form of a negative command (e.g., DO NOT CLIMB, DO NOT DESCEND), which is referred to as the aircraft's *resolution intent* or, more formally, its *coordination complement*. (↓Coordination-Update $_{a15}$, ↓ Vertical-RAC $_{p22}$)

[2.64.1]
This information is expressed in the form of a complement (e.g., if one aircraft has selected an "upward-sense" advisory with respect to the threat, it will transmit a "climb intent" message in its coordination interrogation to the threat, restricting the threat's solution of RAs to those in the "downward sense." (↓Coordination-Vertical-RAC $_{a15, p15-22}$)

[2.64.2]
The actual strength of the "downward-sense" RA selected by the other TCAS aircraft would depend on the geometry of the situation.

[2.5]

The basic rule for sense selection in a TCAS/TCAS encounter is that before selecting a sense, each TCAS must check to see if it has received an intent from the threat. If so, TCAS selects the opposite sense. If not, TCAS selects the sense based on the encounter geometry just as if the threat were not TCAS-equipped. (↑)

[2.6.1]

In the majority of cases, the two aircraft see each other as threats at slightly different moments in time. Coordination proceeds in a straightforward manner with the first aircraft selecting and transmitting its (geometry-based) sense and, slightly later, the second aircraft selecting and transmitting the opposite sense.

[2.6.2]

Occasionally, the two aircraft see each other as threats simultaneously and, therefore, both select a sense based on geometry. In this case, there is some probability (e.g., when the aircraft are flying level at the same altitude) that both will select the same sense. If this should happen, the aircraft with the higher Mode S address will detect the incompatibility and will reverse its sense. It is desirable that this reversal be invisible to the pilot and, therefore, display of the RA is deferred under the following conditions: In a TCAS/TCAS encounter, if the higher address aircraft selects a sense without knowledge of the threat's intent, i.e., if there is a possibility that the higher address aircraft will have to reverse its sense, then the higher address aircraft will delay up to 3 seconds in displaying its advisory to the pilot, waiting to receive the threat's intent. (↓Reversal-Provides-More-Separation$_{m-301}$, ↓RA-Display-Delay$_{m-344}$)

[2.6.3]

If an incompatible intent is not received by the higher address aircraft within the 3 seconds, due to a temporary failure of the Mode S link, the RAs are assumed to be incompatible and the pilot will see a reversal of the displayed resolution advisory (↑SC7.2, ↓Displayed-Advisory$_{s-274}$)

[2.66]

Selecting a noncrossing sense providing ALIM separation A sense-selection algorithm recognizes when an altitude-crossing sense has been selected by the modeling logic and converts it to the noncrossing sense if that sense will provide at least ALIM feet separation at closest approach (↑SC7.1). This logic is operative if the two aircraft are initially separated by more than 300 feet. (Crossing$_{s-280}$)

[2.661]

The value of 300 feet was chosen for this threshold to ensure that effective separation would be achieved when the initial separation made crossing or noncrossing of small concern (given high confidence in the intruder's vertical rate). Furthermore, the threshold precluded the possibility of selecting a noncrossing sense in situations where a TCAS aircraft with a sufficiently high vertical rate would cross through the intruder aircraft's altitude twice as it executed the noncrossing RA maneuver.

[2.626]

The logic considers only the separation achievable at closest approach, not the vertical rate of the TCAS aircraft, in its decision to select a crossing sense. The primary benefit of this logic is that it eliminates a significant proportion of altitude crossing RAs, independent of which aircraft have the vertical rates. An additional benefit is that RAs that would have required the TCAS aircraft to maintain high vertical rates are significantly reduced.

Performance Monitoring

[2.67]
The purpose of automatic performance monitoring and self-test is to detect malfunctions that degrade or preclude TCAS protection. Particular attention should be given to monitoring functions whose failure could result in close vertical proximity (100 ft vertical separation or less) at the point of closest approach when such vertical proximity would not have occurred had the aircraft not carried TCAS (↑FTA-850)

[2.68]
When a failure is detected, RAs or both TAs and RAs may be inhibited (↑1.23.4.2). Failure messages are displayed on RA and TA displays (↑ 1.23.1).

[2.69]
System Performance Monitoring: The monitor includes a self-test function capable of being initiated by the pilot (↑1.23).

> [2.69.1]
> When on the ground, manually activated self-test checks the aural alarms and activates each display element in a predetermined temporal pattern that allows visual verification that display outputs issued by the digital processor can be correctly interpreted by the pilot.
>
> [2.69.2]
> When in the air, the aural alarm will sound and the TRAFFIC light (↑1.23.4.1).

[2.70]
Computer Monitoring: Among the monitoring tests include random access memory pattern tests, CPU instruction tests, program memory tests, timing tests, output function tests, CPU [illegible] rectional antenna. In addition, the monitor must be capable of recognizing a

kstpaetresiastdteunptlon recognition shall declare a TCAS failure. The
dwuqe tmoeniTtCoAr S must also be capable of detecting a
fdwiegluhepnrceiyoritDyetinecptuiotn of in the TCAS high
qpuriofroitythirneuptconsecutive seconds shall cause the monitor to
declare a TCAS failure. [4]

---

[4]Declaring a TCAS failure during a threat situation is potentially hazardous, and this hazard should be mitigated somehow.

Pilot Tasks and Procedures

We could find little documentation on the rationale behind the design of the pilot procedures so this section is very incomplete. The following represents an example of what might be included

[2.71]
Normal Procedures:

[2.71.1]
The TCAS RA algorithms are based on the pilot initiating the initial maneuver within approximately 5 seconds and within approximately 2 1/2 seconds if an additional corrective RA, for example, increase or reverse, is issued (↑OP.10).

[2.71.2]
An aircraft's low altitude climb capability during takeoff, approach or landing is significantly affected by the aircraft's configuration, the initial climb true airspeed available to safely trade if needed for climb rate, and the initial airspeed margin from the current stall speed. To be consistent with normal operation, the aircraft flight manual or flight manual supplement should indicate that when a climb RA occurs with the aircraft in the landing configuration, the pilot should initiate the normal go-around procedure when complying with the TCAS II RA (↑OR.5, 4).

[2.71.3]
The pilot is not permitted to start an evasive maneuver based solely on the TA display information because the information is designed to assist in visually identifying traffic and anticipatory maneuvers might invalidate the TCAS logic (↑OP. 9, ↓FTA-395, 3.3).

[2.71.4]
Once the pilot has established visual contact with the threat traffic, he/she may maneuver the aircraft without being given a resolution advisory by TCAS (↑OP. 1). However, this should be discouraged as it

greatly reduces the accuracy of TCAS to predict the separation (↑FTA-395). The maneuver logic provides ample time to obtain separation once conditions require it.

[2.71.5]
When parallel visual approaches are in progress to runways spaced less than 3000 feet apart, pilots should select TA-only mode of operation to prevent maneuvering advisories being issued for planned close separation (↑OP-6).

[2.72]
Nonnormal Procedures:

[2.72.1]
Because of the limited number of inputs to TCAS for aircraft performance inhibits, in some instances where inhibiting RA's would be appropriate it is not possible to do so. In these cases, TCAS may command maneuvers that may significantly reduce stall margins or result in stall warning. Conditions where this may occur include bank angle (wings level is assumed), weight, altitude, temperature combinations or initial speeds outside the safe operational envelope of the aircraft, engine out, extreme climb RA on approach, and abnormal configurations such as landing gear not retracted (↑SC9.1, 4.AFMS).

[2.72.2]
In the evaluation of aircraft performance for the identification of required aircraft inhibits, airspeeds between $1.2V_{sg}$ and stall warning are not considered unsafe and represent usable airspeed trade for evaluation of some low probability events. The altitude/temperature envelope used represents a range of values that exist at busy airports in the continental U.S. Operation routinely outside this envelope may require special crew procedures if the normal AFM weight, altitude, temperature limitations, or configuration are not sufficient, e.g., operation at Mexico City (↑SR4, 4.AFMS).

Pilot–TCAS Interface

Controls

. *We do not know the rationale behind the design of the controls*

Annunciations

[2.73]
Displayed traffic and resolution advisories are supplemented by computer-generated synthetic voice advisories. TCAS uses annunciations that have been adopted as aviation industry standards (↑SC11.4)

[2.73.1]
An aural annunciation is generated when the first resolution advisory of an encounter is displayed and each time a subsequent change in the advisory is displayed (strengthened or weakened).

[2.73.2]
The words "TRAFFIC, TRAFFIC" are annunciated at the time of the traffic advisory and are used to direct the pilot to look at the TA display to locate the traffic.

[2.73.3]
If after a traffic advisory the encounter does not resolve itself, a resolution advisory is annunciated. (↓Aural-Alarm_Message)

[2.73.4]
An aural annunciation shall also be provided to the flight crew to indicate that the TCAS aircraft is clear of conflict with all threatening aircraft. This annunciation shall occur when the resolution advisory is cleared and the aircraft have begun to diverge in range. (↓Aural-Alarm_Message)

[2.73.5]
Other aural annunciations may differentiate resolution advisories by type: climb, descend, or vertical speed limit, crossing, reversal of advisory sense, increase in vertical rate, or maintain current vertical rate. It may also indicate that an RA is either corrective or preventive or, when an RA is removed, that a threat aircraft's track has been dropped or that the threat is no longer reporting altitude. (↓Aural-Alarm-Message$_{a16}$, ↓ Vertical-Control$_{s231}$)

[2.73.6]
Aural annunciations must be inhibited below 4 3, (↑SC GO) ↓ Aural-Alarm-Inhibit$_{v222}$)

Display

[2.74]
The display of time-critical TCAS advisories should emphasize guidance information. Graphic formats are preferable, however, formatted information may be supplemented with alphanumeric data or messages. Color coding is recommended to aid the pilot in discriminating and responding with the required priority (↑SC 11.4)(↓Resolution-Advisory$_{a17}$)

[2.75]
The display cancels automatically when the advisory no longer exists. (↓Resolution-Advisory$_{a17}$)

Traffic Display

[2.76]
Purpose: The primary purpose of the traffic display is to aid the flight crew in the visual acquisition of threat aircraft. This goal is accomplished by displaying the intruder aircraft's horizontal and vertical position relative to the TCAS II equipped aircraft. A secondary purpose of the traffic display is to provide the flight crew with confidence in proper system operation and to give them time to prepare to maneuver the aircraft in the event TCAS issues a resolution advisory.

[2.77]
TA Display: The functions of the TA display are to aid the flight crew in visually acquiring intruder aircraft, discriminating between intruders, threats, and other aircraft, and in determining the horizontal position (range and bearing) of transponder-equipped aircraft. Relative altitude

flight level or, optionally, shown for those targets reporting pressure altitude. It may also be used to observe the flight paths of nearby traffic.
This allows the pilots to see dangerous situations developing and to prepare both mentally and physically for possible evasive maneuvering.

All of this information, which is updated once per second, tends to instill confidence in the RA that may follow, as well as being an aid to see-and-avoid. At a minimum, there must be means for displaying TAs for at least three targets including range, altitude, and bearing of intruder aircraft. TAs may also indicate range rate and altitude rate of the intruding aircraft relative to own aircraft. Traffic advisories without altitude may also be provided on non-altitude reporting, transponder-equipped aircraft. (↓Intruder-Info)

[2.78]
The FAA has worked closely with the ATA, NASA, and both the SAE S-7 and G-10 Committees in an effort to standardize TCAS symbology and features. A consensus has been reached for many TCAS symbols. Other symbology and features may be used provided the human factors technology is used to demonstrate that a clear and substantial benefit can be derived. Otherwise, the display must depict the standard symbology, features, or information (↑SC11.4,C.4)

Resolution Advisory Display

[2.79]
RA Display: The RA display is the primary instrument used by the pilot to determine whether an adjustment in the aircraft vertical rate is necessary to comply with the RA determined by TCAS. This determination is based on the position of the VSI needle with respect to the lighted segments or eyebrow lights. (↓Resolution-Advisory)

The lighted eyebrows on an instantaneous Vertical Speed Indicator (IVSI) fulfill the requirement. Similarly, conspicuous illumination of the vertical speed tape or a suitable written message on a Primary Flight Display (PFD) or Electronic Attitude Director Indicator (EADI) fulfills the requirement

[2.80]
As a minimum, the following display functions shall be provided:

1. Either singly or in combination the following resolution advisories:

2500

2500

MAINTAIN ... L

MAINTAIN ... EN

CLIMB

DESCEND

DON'T CLIMB

DON'T DESCEND

DON'T ... 500

DON'T ... 1000

DON'T ... 2000

DON'T DESCEND FPM 500

DON'T DESCEND FPM 1000

DON'T DESCEND FPM 2000

(↓Resolution ↓Advisory ↓TCAS ↓Composite -RA ↓RA-Strength)

2. Alert to the flight crew that own TCAS equipment has failed.

3. Alert to the flight crew that the TCAS ... either automatically or through crew action.

Optional Displays

[2.81]

: Optionally, other displays may be provided such as

1. Means of displaying the following resolution advisories (↓Own ... rate to maintain safe separation. (- Goal-Altitude-Rate ... Combined ... Vertical ... -Control)

2. Means of displaying the following traffic advisories: (... true *tau* (intruder range, altitude, vertical miss distance at closest approach ... relative altitude and bearing ... between own and intruder ... range rate between own and intruder aircraft. (↓Intruder - Info)

3. Other optional display functions, available from TCAS (see level 3) or existing aircraft systems, may be used provided the intended function of TCAS is not impaired.

(↓Altitude ↓Range ↓Bearing ... Other ... -Trac -Alt-Rate ↓Other -Trac -Range-Rate)

Note: These display details are included in the principles section because they help determine what changes are possible or allowed when a change to the behavioral design in level 3 is proposed.

Testing and Validation

This section includes requirements on and information about the validation of the design principles included in this level of the Intent Specification

Simulations

[*This section would include descriptions of simulation requirements and either the results once they are completed or a reference to where the results can be found. We know simulations were done on TCAS, but we do not have the references or any information on the requirements that were developed for the simulations (if any)*] *so we have not included this information*

Experiments

Shortly after the decision to proceed with the implementation of TCAS in 1981, prototypes of TCAS II were installed on two Piedmont Airlines B-727 aircraft. The displays were located outside the view of the flightcrew and were seen only by trained observers. The tests provided information on the frequency and circumstances of alerts and their potential for interaction with the ATC system. In a follow-on phase II program, a later version of TCAS II was installed on a single Piedmont Airlines B-727, and the system was certified in April 1986 and subsequently approved for operational evaluation in early 1987. The equipment was not developed to full standards, and therefore the system was only operated in visual meteorological conditions. Although the flightcrew operated the system, the evaluation was primarily for the purpose of data collection and its correlation with flightcrew and observer observation and response.

Bendix/King Air Transport Avionics version of TCAS II manufactured by [illegible] were installed and approved on United Airlines airplanes in early 1988. Honeywell units manufactured by Sperry were installed and approved on Northwest Airlines airplanes in late 1988. This Limited Installation Program operated

TCAS II units approved for operation as a full time system in both visual and instrument meteorological conditions on three different aircraft types. The operational evaluation programs continued through 1988 to validate the operational suitability of the systems. The results are documented in . . . .

Other Validation Procedures

[*This section would include requirements for or descriptions of any other types of validation done on the system design*]

Level 3
Blackbox Behavior

This level describes the inputs and outputs of each component and their relationships with the blackbox behavior of each component. The description includes no internal component design information and behavior is described only in terms of externally visible variables, objects, and mathematical functions. Any of these components theoretically could be implemented using analog or digital technology although practical considerations will normally limit the implementation medium. The level above will answer questions about the intent or purpose of the information in this level.

Environment

This section includes information about the behavior of environment components that is needed for correct TCAS design. We have included part of a SpecTRM-RL specification for the altimeter as an example. Because SpecTRM-RL specifications are executable, the TCAS and environment-component behavioral descriptions at this level of intent specification can be used in simulations, animations, and analyses of the system's blackbox behavior.

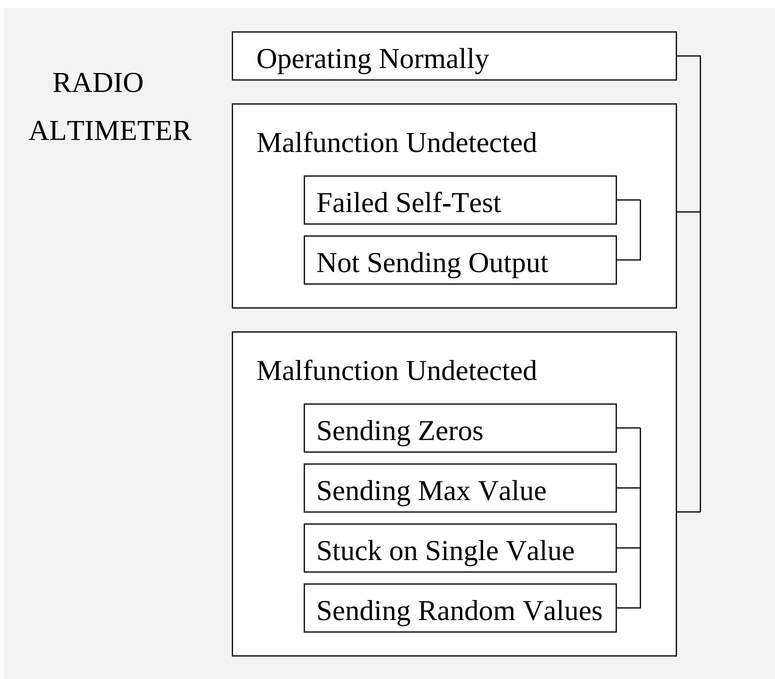


Figure 3.6: Part of a SpecTRM-RL Specification of an Altimeter

Flightcrew Behavioral Requirements

Task Analysis

This section would include a task analysis of the tasks involved in using TCAS. We do not have any examples of such a task analysis for TCAS and are not competent to do one ourselves so we have not included an example here.

Operational Procedures

This section of an intent specification includes the operational procedures for TCAS. We are designing a language for these procedures that will execute along with the other specifications at this level of an Intent Specification, but we need to get some more experience with this modeling language so we have not included an example here. We will add one later. We have included an example English description of part of the operational procedures.

[3.1]
Pilot-Initiated Self-Test:

1. During cockpit preparation, a self-test of TCAS [illegible] .
2. During [illegible], the pilot may use the self-test [illegible] feature.

[3.2]
Ground Operation

1. [illegible] prior to takeoff.
2. [illegible] ter clearing runway [illegible] landing.

[3.3]
Normal Procedures

1. The system operation of t ... used to preclude unnecessary RAs when operating near closely spaced parallel runways (than 2500 feet).

2. Upon receipt of a TRAFFIC, TRAFFIC ..., the pilot shall attempt to acquire the target visually ... information provided (procedure 2.71.3).

3. The pilot ... maneuver ... before an RA (procedure 2.71.3).

4. Prior to taking action in the RA, the pilot shall clear the ... airspace ... into ... area.

5. TCAS ... maneuver ... in a timely gentle fashion.

6. Proper response ... the ... green light segment.

(a) If the RA ... a ... segment, the pilot should ... aircraft ... vertical rate until the green ... (e.g. corrective RA).

(b) If ..., the pilot should maintain the current ... (preventive RA).

7. Upon the single CLEAR OF CONFLICT ..., the pilot should promptly .... If the threat aircraft ... altitude information ... RA, the RA will terminate ... a CLEAR OF CONFLICT ... ation.

8. Any time ... de... reporting ..., TCAS ... turned off.

9. If a climb RA ... the aircraft in the landing configuration, a normal go-around ... appropriate ... the increase ... configuration change.

[3.4]
Non-Normal Procedures

Not included due to lack of information

Pilot–TCAS Interface

This section ... approach ... presenting information to the pilot ... recommended by the Air Transport Association of America (ATA) ... for TCAS Displays, Symbology, and Controls ... FAA ... AC20-131.

Controls

[3.5]
As a minimum, the following ... control ... function:

1. Selection of one of the following operational modes:
   - **STANDBY:** Eq... 1.
   - **TA-ONLY:** Eq... 2 (↑1.13).
   - **AUTOMATIC:** ... current altitude ... of aircraft ... 3-7).
2. Means of initiating a ... 2.6).

Aural Annunciations

The following aural annunciations ...:

Traffic Advisory: TRAFFIC, TRAFFIC

Resolution Advisories:

Preventive:

MONITOR VERTICAL SPEED, MONITOR VERTICAL SPEED: [illegible]

Corrective:

CLIMB—CLIMB—CLIMB: [illegible] rate [illegible] RA [illegible] indicator [illegible] 1500 fpm.

CLIMB, CROSSING CLIMB—CLIMB, CROSSING CLIMB: [illegible] except [illegible] indicate [illegible] that [illegible] at [illegible].

DESCEND—DESCEND—DESCEND: [illegible] rate [illegible] RA in- [illegible] 1500 fpm.

DESCEND, CROSSING DESCEND—DESCEND, CROSSING DESCEND: [illegible] except [illegible] that [illegible] at [illegible].

REDUCE CLIMB—REDUCE CLIMB: Reduce vertical [illegible] on the RA indicator [illegible].

REDUCE DESCENT—REDUCE DESCENT: Reduce vertical [illegible] on the RA indicator [illegible].

INCREASE CLIMB—INCREASE CLIMB: Follows a CLIMB advisory. [illegible] RA [illegible] 2500 fpm.

INCREASE DESCENT—INCREASE DESCENT: F[illegible]. The [illegible] vertical [illegible] RA indicator [illegible] 2500 fpm.

CLIMB, CLIMB NOW—CLIMB, CLIMB NOW: Follows a DESCEND [illegible] determine [illegible] separation [illegible].

DESCEND, DESCEND NOW—DESCEND, DESCEND NOW: Follows a CLIMB ad- [illegible] determine [illegible] separation [illegible].

[illegible] CLEAR OF CONFLICT [illegible] encounter [illegible] clearance.

Traffic Advisory Display

... characteri ... the following requirements.

... at lea ... the aircraft ... nmi ... the aircraft.

... intruder aircraft ... colli ... within 20 to 30 seconds. If the intruder aircraft ... determine ..., threat ... color ...

... targets ... Figure 3.7 ...

1. Own aircraft ... or airplane-li ... cyan.
2. ... target ... range ... open ... Non-altitude reporting aircraft ... target ... shall not ... a data tag.
3. ... target ... within 6 nmi in range and ... a data tag.
4. A ... target ... filled ... a data tag.
5. An ... target ... filled ... a data tag.

... data tag ... the relative altitude ... target in ... feet. ... or minus ... to ... target ... own aircraft. If the target aircraft ... 500 feet or more per minute, ... altitude ... arrow pointing up or ...

... approach or departure ... from a terminal area ... from 2700 feet to ... lower ... from 2700 feet ... the upper limit of ..., ... respecti ...

RA or TA target ... range of ... tag ... relative ... bearing from own aircraft.

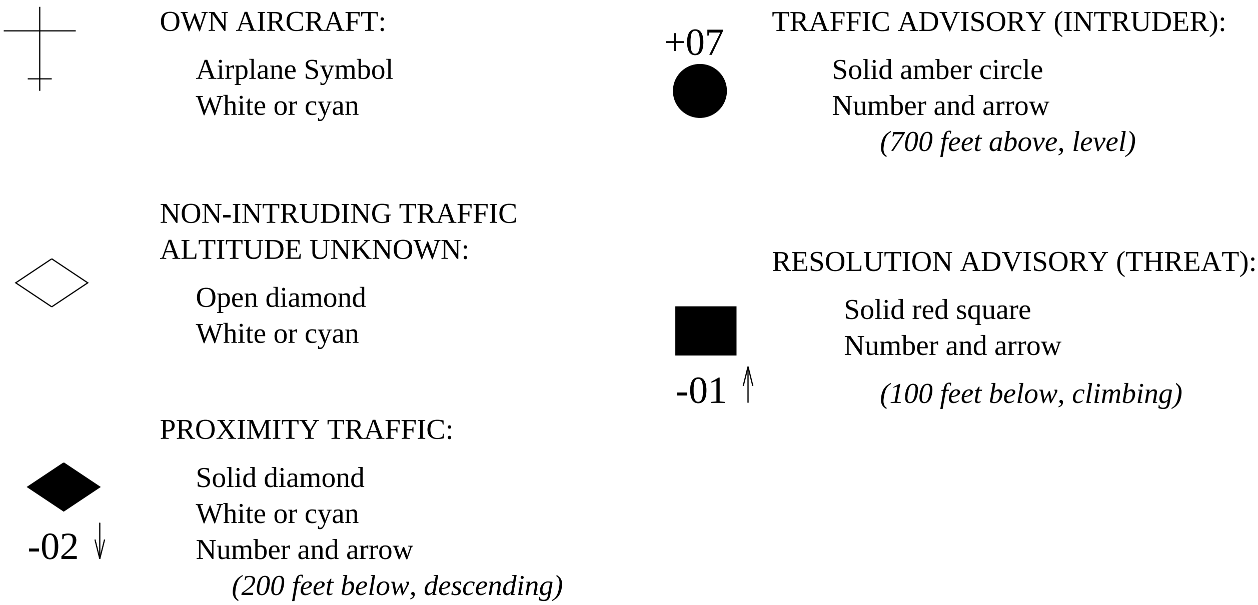


Figure 3.7: ... or TA Display

If a TA or an RA is generate ... intru ... n ... g ... ing inf ... ormation range an ... relative altitu ... TA ... numericall ... a ... f or a TA or r ... or an RA.

... TA ... optionall ... a ... TA and R ... A ... ays ...

Resolution Advisory Display

) ... TCAS ... o e ... erence to ... b ... clim ... ent i ... requi ... de ... tion ... ea tance.

The R A d ... ay ... contain re ... a green ... , ... e ... d ... t ... all 200, 500, 1000, 1500, 2000, 2500, and 6000 fpm. A ... ment ... li ... g up re ... or green. Green ... ment ... te ... spi ... is ... to .

A ... ment ... lit ... l ... R correcti ... As. A ... RA is ... s e ... ting out ... t ...

e, apinment counter eRRgonta a correcti A ad
awirce .

Wh hisrat colli ntewsoinger e shxialigai .

Communication and Interfaces

This section describes ... Communication ... .

Figure 5.8 shows all ... communication ... 1) the uplink ... messages for ... me aircraft, ... aircraft ... 2) ... requirement ... on the ... aircraft ... ship ... .

Message Formats

Uplin rmat

contain orma

ill tr an to illicit rom

.

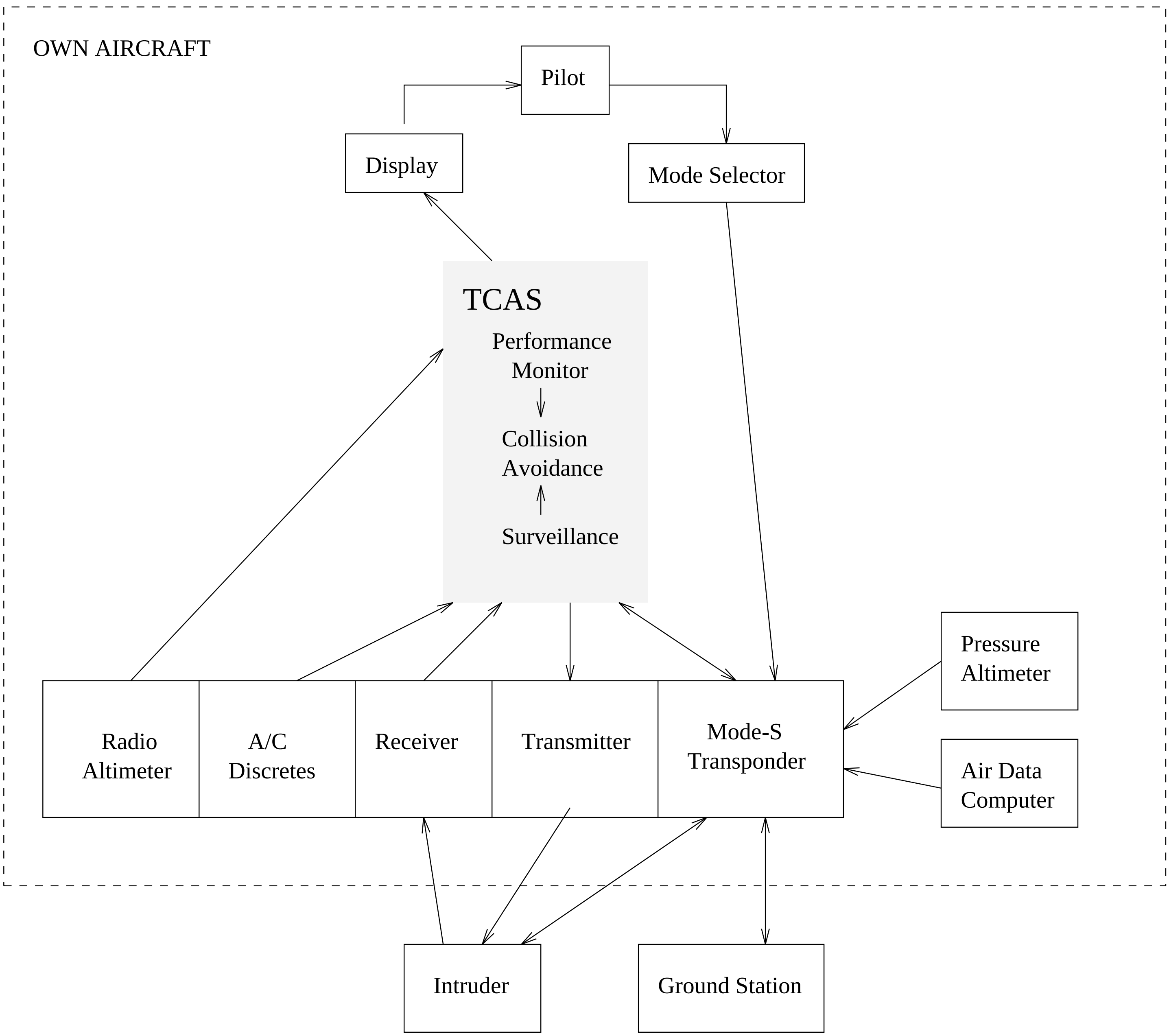


Figure 3.8: TCAS environment

Name: `Mode-S Acquisition`
Message Format: UF-0 (Short Special Surveillance)
MOPS Reference: Tracking 2.2.2.2

Source: TCAS equipped aircraft (ownship).
Destination: Mode-S equipped aircraft.
Timetype: S-R

Data Representation:

UF Uplink Format 1 5	- Not assigned 6 8	RL Reply Length 9 9	- Not assigned 10 13	AQ Acquisition 14 14	- Not assigned 15 32	AP Address/-Parity 33 56

Contained Subfields

Field	**Description**
UF	UF transmissions are interrogations. 0 = Short Air to Air Surveillance
RL	Commands a specific air-to-air reply format. 0 = Request short reply. (Long Replies requested only by ground stations.)
AQ	Identifies interrogation as an acquisition or non-acquisition interrogation. 1 = Acquisition Interrogation.
AP	Mode-S address (overlaid on parity) as described in Ref. B 4.1

Comments:

Name: `TCAS Tracking`
Message Format: UF-0 (Short Special Surveillance)
MOPS Reference: Tracking 2.2.2

Source: TCAS equipped aircraft (own aircraft).
Destination: Mode-S equipped aircraft.
Timetype: S-R

Data Representation:

UF Upkink Format 1 5	- Not assigned 6 8	RL Reply Length 9 9	- Not assigned 10 13	AQ Acquisition 14 14	- Not assigned 15 32	AP Address/-Parity 33 56

Contained Subfields

Field	**Description**
UF	UF transmissions are interrogations. 0 = Short Air-to-Air Surveillance
RL	Commands a specific air-to-air reply format. 0 = Request short reply. (Long Replies requested only by ground stations.)
AQ	Identifies interrogation as an acquisition or non-acquisition interrogation. 0 = Non-Acquisition Interrogation.
AP	Mode-S address (overlaid on parity) as described in Ref. B 4.1.

Comments:

Name: `Mode-S Surveillance, Altitude Request`
Message Format: UF=4 (Surveillance Request)
MOPS Reference: [illegible]

Source: Mode-S 1
Destination: Mode-S 2
Timetype: S-R

Data Representation:

UF	PC	RR	DI
Uplink Format	PC Protocol	Reply Request	Destinator Identification
1 – 5	6 – 8	9 – 13	14 – 16

If `DI` = 1 then

(SD)IIS	(SD)MBS	(SD)MES	(SD)LOS	(SD)RSS	(SD)TMS	AP
Interrogator Identifier	Multisite Comm-B	Multisite ELM	Lockout	Reservation Subfield	Tactical Subfield	Address/-Parity
17 – 20	21 – 22	23 – 25	26 – 26	27 – 28	29 – 32	33 – 56

else if `DI` = 7

(SD)IIS	(SD)RRS	(SD)-	(SD)LOS	(SD)-	(SD)TMS	AP
Interrogator Identifier	Reply Request	Not assigned.	Lockout	Not assigned.	Tactical Subfield	Address/-Parity
17 – 20	21 – 24	25 – 25	26 – 26	27 – 28	29 – 32	33 – 56

Contained Subfields

Field	Description
UF	UF transmissions are interrogations. 4 = Surveillance Altitude Request

PC	This field is mainly concerned with acknowledging the conclusion of various data-link transactions. PC code - Meaning 0 = No changes in transponder state 1 = Nonselective All-call lockout 2 = Not assigned 3 = Not assigned 4 = Comm-B close out 5 = Uplink ELM close out 6 = Downlink ELM close out 7 = Not assigned
RR	Specifies length and content of the reply requested. Airborne TCAS equipment responds to RR=19. RR code · Reply Length · MB field contents 0-15 · Short · - 16 · Long · Air-initiated Comm-B 17 · Long · Data-link capability 18 · Long · Aircraft Mode A identity 19 · Long · TCAS information 20-31 · Long · Not assigned
DI	Specifies the format of the SD field. DI - Meaning 0 = Return only the `(SD)IIS` field 1 = Maintain or closeout communication 7 = Maintain or specify format of return Messages
SD	Can contain control codes affecting link protocols. Content of SD is specified by the DI field.
(SD)IIS	Identification of interrogating site
(SD)MBS	Reserve/close out multisite Comm-B
(SD)MES	Reserve/close out multisite ELM
(SD)LOS	Multisite lockout control
(SD)RSS	Reservation status request
(SD)RRS	Specifies BDS2 code of MB messages
AP	Mode-S address (overlaid on parity) as described in Ref. B 4.1.

Comments:

Name: `Mode-S Surveillance, Identity Request`
Message Format: UF=5 (Surveillance, Identity Request)
MOPS Reference: Tmgc

Source: Mode-S 1
Destination: Mode-S 2
Timetype: S-R

Data Representation:

UF	PC	RR	DI
Uplink Format	PC Protocol	Reply Request	Destinator Identification
1 – 5	6 – 8	9 – 13	14 – 16

If `DI` = 1 then

(SD)IIS	(SD)MBS	(SD)MES	(SD)LOS	(SD)RSS	(SD)TMS	AP
Interrogator Identifier	Multisite Comm-B	Multisite ELM	Lockout	Reservation Subfield	Tactical Subfield	Address/-Parity
17 – 20	21 – 22	23 – 25	26 – 26	27 – 28	29 – 32	33 – 56

else if `DI` = 7

(SD)IIS	(SD)RRS	(SD)-	(SD)LOS	(SD)-	(SD)TMS	AP
Interrogator Identifier	Reply Request	Not assigned.	Lockout	Not assigned.	Tactical Subfield	Address/-Parity
17 – 20	21 – 24	25 – 25	26 – 26	27 – 28	29 – 32	33 – 56

Contained Subfields

Field	Description
UF	UF transmissions are interrogations. 5 = Surveillance Identity Request

PC	This field is mainly concerned with acknowledging the conclusion of various data-link transactions. PC code Meaning 0 = No changes in transponder state 1 = Nonselective All-call lockout 2 = Not assigned 3 = Not assigned 4 = Comm-B close out 5 = Uplink ELM close out 6 = Downlink ELM close out 7 = Not assigned
RR	Specifies length and content of the reply requested. Airborne TCAS equipment responds to RR=19. RR code / Reply Length / MB field contents 0-15 / Short / - 16 / Long / Air-initiated Comm-B 17 / Long / Data-link capability 18 / Long / Aircraft Mode A identity 19 / Long / TCAS information 20-31 / Long / Not assigned
DI	Specifies the format of the SD field. DI Meaning 0 = Return only the `(SD)IIS` field 1 = Maintain or closeout communication 7 = Maintain or specify format of return Messages
SD	Contains the meaning of the SD can contain control codes affecting link protocols. Content of SD is specified by the DI field.
(SD)IIS	Identification of interrogating site
(SD)MBS	Reserve/close out multisite Comm-B
(SD)MES	Reserve/close out multisite ELM
(SD)LOS	Multisite lockout control
(SD)RSS	Reservation status request
(SD)RRS	Specifies BDS2 code of MB messages
AP	Mode-S address (overlaid on parity) as described in Ref. B 4.1.

Comments:

Name: `Mode S only All Call`

Message Format: UF-11 (Mode S only All Call)

MOPS Reference:

Source: Mode S equipped aircraft.

Destination: Mode-S to Ground Station or .

Timetype: periodic (Reporting period 12 s)

Data Representation:

DF Downlink Format 1 5	PR Reply Probability 6 9	II Interrogator Identifier 10 13	- Not assigned. 14 32	PI Parity/-Identity 33 56

Contained Subfields

Field	Description
DF	Defines the type of transmission. DF transmissions are replies. 11 = All-Call Interrogation
PR	Assigns a probability of reply to the transponder controlled by an internal changing random number. The meanings are: PR code — Meaning 0 — Reply with probability = 1 1 — Reply with probability = 1/2 2 — Reply with probability = 1/4 3 — Reply with probability = 1/8 4 — Reply with probability = 1/16 5, 6, 7 — Do not reply 8 — Disregard lockout, reply with probability = 1 9 — Disregard lockout, reply with probability = 1/2 10 — Disregard lockout, reply with probability = 1/4 11 — Disregard lockout, reply with probability = 1/8 12 — Disregard lockout, reply with probability = 1/16 13, 14,15 — Do not reply

II	Permits a sensor to disable an acquired transponder from replying to further All-call interrogations from that sensor without inhibiting the transponder from reploying All-call interrogations from other sensors which may yet need to acquire the aircraft.
PI	Ref.B 4.1.

Comments:

Namemeodbceasgt:ion

lgMessa rmat: UF-16 (Lbng Speci ailSur a)nce

MOPS Reference: TtCA Srogation 2.2.2 4

Source: TCASpeaf)bain(e

Destination: AlCASpe aircraft)(ba .

Timetype/Period: . Brata 10s interval .

Data Representation:

UF Uplink Format 1 5	- Not assigned 6 8	RL Reply Length 9 9	- Not assigned 10 13	AQ Acquisition 14 14	- Not assigned 15 32
(MU)UDS1 U Subfield 1 33 36	(MU)UDS2 U Subfield 2 37 40	- Not Assigned 41 64	(MU)MID Mode S address 65 88	AP Address/-Parity 89 112	

Contained Subfields

Field	**Description**
UF	UF transmissions are interrogations. 16 = Long Air-to-Air Surveillance
RL	Not Used, no reply is generated
AQ	Identifies interrogation as an acquisition or non-acquisition interrogation. 0 = Non-Acquisition Interrogation.
MU	Used by airborne TCAS to indicate own presence and identity to other TCAS equipment. (Does not use Comm-A protocol Ref.B 4.10)
(MU)UDS1	3 = Sets MU format for a TCAS Broadcast Interrogation
(MU)UDS2	2 = Sets MU format for a TCAS Broadcast Interrogation
(MU)MID	Mode-S Address of Interrogating TCAS

AP	Mode-S address (overlaid on parity) as described in Ref. B 4.1.

Comments:

Name: `Mode-S Resolution Message`
Message Format: UF-16 (Long Special Surveillance)
MOPS Reference: TCAS Resolution Message 2.2.8 3.1.2

Source: TCAS equipped aircraft (broadcast)
Destination: All TCAS equipped aircraft (broadcast).
Timetype: S-R.

Data Representation:

UF	-	RL	-	AQ	-	MU
Uplink Format	Not assigned	Reply length	Not assigned	Acquisition Interrogation	Not assigned	Message Comm-U
1 5	6 8	9 9	10 13	14 14	15 32	33 88

(MU)UDS1	(MU)UDS2	(MU)-	(MU)MTB	(MU)CVC	(MU)VRC	(MU)CHC	(MU)HRC
U Subfield 1	U Subfield 2	Not assigned.	Multiple Threats	Cancel Res. Adv. Comp.	Vertical Res Adv. Comp	Cancel Horiz. Res. Adv. Comp.	Horiz. Res. Adv. Comp.
33 36	37 40	41 41	42 42	43 44	45 46	47 49	50 52

-	(MU)HSB	(MU)VSB	(MU)MID	AP
Not assigned.	Horizontal Sense Bits	Vertical Sense Bits	Mode-S Address	Address/-Parity
53 55	56 60	61 64	65 88	89 112

Contained Subfields

Field	**Description**
UF	UF transmissions are interrogations. 16 = Long Air-to-Air Surveillance
RL	RL=1, requests long reply

AQ	AQ=0, non-aquisition interrogation.
MU	Used by airborne TCAS to transmit resolution advisory coordination information and to indicate own presence and identity to other TCAS equipment. (Does not use Comm-A protocol Ref.B 4.10)
(MU)UDS1	3 = Sets MU format for a TCAS Resolution message
(MU)UDS2	0 = Sets MU format for a TCAS Resolution message
(MU)MTB	Indicates if aircraft has identified more than one threat. 0 = Interrogating TCAS has no more than one threat. 1 = Interrogating TCAS has more than one threat.
(MU)CVC	Used to cancel a previously sent Vertical Resolution Advisory Complement. 0 = No Cancellation. 1 = Cancel, don't descend. 2 = Cancel, don't climb. 3 = Not assigned.
(MU)VRC	Contains Verical Resolution Advisory Complement. 0 = No vertical resolution advisory complement sent. 1 = Don't Descend. 2 = Don't Climb. 3 = Not assigned.
(MU)CHC	Not Used. TCAS II does not use a Horizontal Advisory so (MU)CHC =0. 0 = No Cancellation. 1 = Cancel, don't turn left. 2 = Cancel, don't turn right. 3 = Not assigned.
(MU)HRC	Not Used. TCAS II does not use Horizontal Advisory so (MU)HRC =0. 0 = No horizontal resolution advisory complement sent. 1 = Don't turn left. 2 = Don't turn right. 3 = Not assigned.
(MU)HSB	Contains parity check for HRC and CHC
(MU)VSB	Contains parity check for VRC
(MU)MID	The discrete address of the interrogating TCAS aircraft.
AP	Mode-S address (overlaid on parity) as described in Ref. B 4.1.

Comments:

Transmitting aircraft: ... on
... Reporte ... 4 (during trac ... .e. TCAS) ... upcoming .
Receiving aircraft: Receives ... receiv ... threat . Resp...
... Coordination Re... .

Name: `Mode-S Comm-A, Altitude Request`
Message Format: UF-20 (Comm-A Altitude Request)
MOPS Reference: Tkagc

Source: Mode-S 1
Destination: Mode-S 2
Timetype: S-R

Data Representation:

UF	PC	RR	DI
Uplink Format	PC Protocol	Reply Request	Destinator Indentification
1 – 5	6 – 8	9 – 13	14 – 16

If `DI` = 1 then

(SD)IIS	(SD)MBS	(SD)MES	(SD)LOS	(SD)RSS	(SD)TMS
Interrogator Identifier	Multisite Comm-B	Multisite ELM	Lockout	Reservation Subfield	Tactical Subfield
17 – 20	21 – 22	23 – 25	26 – 26	27 – 28	29 – 32

else if `DI` = 7

(SD)IIS	(SD)RRS	(SD)-	(SD)LOS	(SD)-	(SD)TMS
Interrogator Identifier	Reply Request	Not assigned.	Lockout	Not assigned.	Tactical Subfield
17 – 20	21 – 24	25 – 25	26 – 26	27 – 28	29 – 32

(MA)ADS1	(MA)ADS2	(MA)SLC	AP
A Subfield 1	A Subfield 2	TCAS Sensitivity Level	Address/-Parity
33 – 36	37 – 40	41 – 44	89 – 112

Contained Su[illegible]

Field	Description
UF	UF transmissions are interrogations. 20 = Comm-A, Altitude Request
PC	This field is mainly concerned with acknowledging the conclusion of various data-link transactions. PC code Meaning 0 = No changes in transponder state 1 = Nonselective All-call lockout 2 = Not assigned 3 = Not assigned 4 = Comm-B close out 5 = Uplink ELM close out 6 = Downlink ELM close out 7 = Not assigned
RR	Specifies length and content of the reply requested. Airborne TCAS equipment responds to RR=19. RR code / Reply Length / MB field contents 0-15 / Short / - 16 / Long / Air-initiated Comm-B 17 / Long / Data-link capability 18 / Long / Aircraft Mode A identity 19 / Long / TCAS information 20-31 / Long / Not assigned
DI	Specifies the format of the SD field. DI Meaning 0 = Return only the `(SD)IIS` field 1 = Maintain or closeout communication. Since the DI=1, the SD field will contain the following subfi[illegible]d `TM` 7 = Maintain or specify format of return messages. Since the DI=7, the SD field will contain the following subfi[illegible] `TM` [illegible] he rest of the bits in SD are not assigned.
SD	Contains the meaning of the SD which contains control codes affecting link protocols. Content of SD is specified by the DI field. Since there are more subfields then bits available, DI specifies which subfields are used.
(SD)IIS	Identification of interrogating site
(SD)RRS	Specifies BDS2 code of MB messages
(SD)MBS	Reserve/close [illegible]ut multisite Comm-B
(SD)MES	Reserve/close [illegible]ut multisite ELM
(SD)LOS	Multisite lockout control

(SD)RSS	Reservation status request
(SD)RRS	Specifies BDS2 code of MB messages
(SD)TMS	Specifies e hange of Comm-A messages. 0 = Unlimited.
MA	Used by a Mode-S sensor to tranmit a TCAS sensitivity level command message to airborne TCAS.
(MA)ADS1	Indicates subfield for TCAS sensitivity Level Command is contained in MA b[illegible]
(MA)ADS2	Indicates subfield for TCAS sensitivity Level Command is contained in MA b[illegible]
(MA)SLC	This subfield contains a sensitivity level command for a TCAS aircraft. 0 = No command issued. 1 = Set TCAS sensitivity level to 1. 2 = Set TCAS sensitivity level to 2. 3 = Set TCAS sensitivity level to 3. 4 = Set TCAS sensitivity level to 4. 5 = Set TCAS sensitivity level to 5. 6 = Set TCAS sensitivity level to 6. 7 = Set TCAS sensitivity level to 7. 8-14 = Not assigned. 15 = Cancel previous sensitivity level command from this site.
AP	Mode-S address (overlaid on parity) as described in Ref. B 4.1.

Comments:

Name: `Mode-S Comm-A, Identit Request`
Message Format: UF-21 (Comm-A, Ident- Request)
MOPS Reference: Tmgc

Source: Mode-S 1
Destination: Mode-S 2
Timetype: S-R

Data Representation:

UF	PC	RR	DI
Uplink Format	PC Protocol	Reply Request	Destinator Indentification
1–5	6–8	9–13	14–16

If `DI` = 1 then

(SD)IIS	(SD)MBS	(SD)MES	(SD)LOS	(SD)RSS	(SD)TMS
Interrogator Identifier	Multisite Comm-B	Multisite ELM	Lockout	Reservation Subfield	Tactical Subfield
17–20	21–22	23–25	26–26	27–28	29–32

else if `DI` = 7

(SD)IIS	(SD)RRS	(SD)-	(SD)LOS	(SD)-	(SD)TMS
Interrogator Identifier	Reply Request	Not assigned.	Lockout	Not assigned.	Tactical Subfield
17–20	21–24	25–25	26–26	27–28	29–32

(MA)ADS1	(MA)ADS2	(MA)SLC	AP
A Subfield 1	A Subfield 2	TCAS Sensitivity Level	Address/-Parity
33–36	37–40	41–44	89–112

Contained Subfields

Field	**Description**
UF	UF transmissions are interrogations. 21 = Comm-A Identity Request
PC	This field is mainly concerned with acknowledging the conclusion of various data-link transactions. PC code Meaning 0 = No changes in transponder state 1 = Nonselective All-call lockout 2 = Not assigned 3 = Not assigned 4 = Comm-B close out 5 = Uplink ELM close out 6 = Downlink ELM close out 7 = Not assigned
RR	Specifies length and content of the reply requested. Airborne TCAS equipment responds to RR=19. RR code / Reply Length / MB field contents 0-15 / Short / - 16 / Long / Air-initiated Comm-B 17 / Long / Data-link capability 18 / Long / Aircraft Mode A identity 19 / Long / TCAS information 20-31 / Long / Not assigned
DI	Specifies the format of the SD field. DI Meaning 0 = Return only the (SD)IIS field 1 = Maintain or closeout communication. Since the DI=1, the SD field will contain the following subfields: `IIS, [illegible] and TM` 7 = Maintain or specify format of return messages. Since the DI=7, the SD field will contain the following subfields: `IIS, RRS, and TMS` t he rest of the bits in SD are not assigned.
SD	Contains the meaning of the SD which contains control codes affecting link protocols. Content of SD is specified by the DI field. Since there are more subfields then bits available, DI specifies which subfields are used.
(SD)IIS	Identification of interrogating site
(SD)RRS	Specifies BDS2 code of MB messages
(SD)MBS	Reserve / close out multisite Comm-B

(SD)MES	Reserve/close out multisite ELM
(SD)LOS	Multisite lockout control
(SD)RSS	Reservation status request
(SD)RRS	Specifies BDS2 code of MB messages
(SD)TMS	Specifies exchange of Comm-A messages. 0 = Unlimited.
MA	Used by a Mode-S sensor to tranmit a TCAS sensitivity level command message to airborne TCAS.
(MA)ADS1	Indicates subfield for TCAS sensitivity Level Command is contained in MA by/AS=0
(MA)ADS2	Indicates subfield for TCAS sensitivity Level Command is contained in MA by/AS=5
(MA)SLC	This subfield contains a sensitivity level command for a TCAS aircraft. 0 = No command issued. 1 = Set TCAS sensitivity level to 1. 2 = Set TCAS sensitivity level to 2. 3 = Set TCAS sensitivity level to 3. 4 = Set TCAS sensitivity level to 4. 5 = Set TCAS sensitivity level to 5. 6 = Set TCAS sensitivity level to 6. 7 = Set TCAS sensitivity level to 7. 8-14 = Not assigned. 15 = Cancel previous sensitivity level command from this site.
AP	Mode-S address (overlaid on parity) as described in Ref. B 4.1.

Comments:

[illegible] rmat

[illegible] contain [illegible] format of t [illegible] munication [illegible] communicating to ot [illegible] od [illegible] Tran [illegible].

Name: SALTR

Message Format: ... Reply

MOPS Reference: ... Std ... SSR

Source: Mode-C aircraft.

Destination: Mode-S equipped aircraft.

Timetype: S-R

Data Representation:

ID
Identity
1 16

Contained Subfields:

Field	Description
ID	Sends the altitude in a range from -1000ft to 126,... ft in 100ft increments.

Comments:

Detail ... .

Name: `Mode-S Tracking Reply`
Message Format: DF 0 (Short Special) Surveillance
MOPS Reference: Section 2.2.2 3.2

Source: Mode S equipped aircraft.
Destination: TCAS equipped aircraft.
Timetype: S-R

Data Representation:

DF Downlink Format 1 5	VS Vertical Status 6 6	- Not assigned 7 8	SL Sensitivity Level 9 11	- Not Assigned 12 13	RI Reply Information 14 17	- Not Assigned 18 19
AC Altitude Code 20 32	AP Address/-Parity 33 56					

Contained Subfields:

Field	Description
DF	Defines the type of transmission. DF transmissions are replies. 0 = Short Special Reply
VS	Indicates if aircraft is airborne or not. If the aircraft is being tracked then it is already determined it is in the air. 0 = Aircraft airborne. 1 = Aircraft grounded.

<table>
<tr><td>SL</td><td>Current TCAS sensitivity level.
The codes are
0 = No TCAS sensitivity level reported.
1 = TCAS operating at sensitivity level 1.
2 = TCAS operating at sensitivity level 2.
3 = TCAS operating at sensitivity level 3.
4 = TCAS operating at sensitivity level 4.
5 = TCAS operating at sensitivity level 5.
6 = TCAS operating at sensitivity level 6.
7 = TCAS operating at sensitivity level 7.</td></tr>
<tr><td>RI</td><td>Reports TCAS capability of tracked aircraft.
The codes are
0 = No on board TCAS or TCAS has failed or Sensitivity Level is 1.
1 = Not assigned.
2 = TA only Mode.
3 = On-board TCAS with vertical-only resolution capability.
4 = On-board TCAS with vertical and horizontal resolution capability.</td></tr>
<tr><td>AC</td><td>Contains the encoded altitude of the aircraft . Ref.A 2.7.13.2.5.
The M-bit (bit 25) shall be 0 if the altitude is reported in feet. (M=1 is reserved for future use to indicate altitude reported in metric units. Format for metic reporting is undefined.)
The Q-bit (bit 28) Q = 0 indicates that the Mode S altitude is reported in 100ft increments.
Q = 1 indicates that the altitude is reported in 25ft increments.
The possible codes are
<table>
<tr><th>M-bit</th><th>Q-bit</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Altitude coding according to Ref.A 2.7.13.2.5. Starting with bit 20 the sequence shall be
C1, A1, C2, A2, C4, A4, M=0, B1, Q=0, B2, D2, B4, D4.</td></tr>
<tr><td>0</td><td>1</td><td>The 11-bit field represented by bits 20-25, 27, and 29-32 represents a binary field whose least significant bit has a value of 25ft. The field is to be used to report the pressure altitude in the range $(25 \times N - 1000)$ (± 12.5) feet.</td></tr>
</table>
Zero (0) is transmitted each of the 13 bits if no altitude information is available.</td></tr>
<tr><td>AP</td><td>Mode S address (overlaid on parity) as described in Ref. B 4.1.</td></tr>
</table>

Comments:

[illegible] te [illegible] ation [illegible] el to 0.

d b[illegible] um air [illegible] the initial

[illegible] ation [illegible] o c for T[illegible] ecapa .

Name: `Mode-S Acquisition Rp`
Message Format: DF 0 (Srt Speci) avSance
MOPS Reference: Aqion 2.22 3.1

Source: Mod Sippe aircraft.
Destination: TCAS aircraf)(bae .
Timetype: S-R

Data Representation:

DF Downlink Format 1 5	VS Vertical Status 6 6	- Not assigned 7 8	SL Sensitivity Level 9 11	- Not Assigned 12 13	RI Airspeed of Aircraft 14 17	- Not Assigned 18 19
AC Altitude Code 20 32	AP Address Priority 33 56					

Contained Subfields

Field	Description
DF	Defines the type of transmission. DF transmissions are replies. 0 = Short Special Reply
VS	Indicates if aircraft is airborne or not. If aircraft is airborne, then TCAS would proceed to track the aircraft. If aircraft is not airborne TCAS will not track the aircraft 0 = Aircraft airborne.

<table>
<tr><td>SL</td><td>Current TCAS sensitivity level.
The codes are
0 = No TCAS sensitivity level reported.
1 = TCAS operating at sensitivity level 1.
2 = TCAS operating at sensitivity level 2.
3 = TCAS operating at sensitivity level 3.
4 = TCAS operating at sensitivity level 4.
5 = TCAS operating at sensitivity level 5.
6 = TCAS operating at sensitivity level 6.
7 = TCAS operating at sensitivity level 7.</td></tr>
<tr><td>RI</td><td>Reports either airspeed of transmitting aircraft. When airspeed is reported, the maximum true airspeed in normal operation is given.
The codes are
8 = No maximum airspeed data available.
9 = Airspeed .LE. 75 knots.
10 = Airspeed .GT. 75 knots and .LE. 150 knots.
11 = Airspeed .GT. 150 knots and .LE. 300 knots.
12 = Airspeed .GT. 300 knots and .LE. 600 knots.
13 = Airspeed .GT. 600 knots and .LE. 1200 knots.
14 = Airspeed .GT. 1200 knots.
15 = Not assigned</td></tr>
<tr><td>AC</td><td>Contains the encoded altitude of the aircraft . Ref.A 2.7.13.2.5.
The M-bit (bit 25) shall be 0 if the altitude is reported in feet. (M=1 is reserved for future use to indicate altitude reported in metric units. Format for metric reporting is undefined.)
The Q-bit (bit 28) Q = 0 indicates that the Mode S altitude is reported in 100ft increments.
Q = 1 indicates that the altitude is reported in 25ft increments.
The possible codes are
<table>
<tr><th>M-bit</th><th>Q-bit</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Altitude coding according to Ref.A 2.7.13.2.5. Starting with bit 20 the sequence shall be C1, A1, C2, A2, C4, A4, M=0, B1, Q=0, B2, D2, B4, D4.</td></tr>
<tr><td>0</td><td>1</td><td>The 11-bit field represented by bits 20-25, 27, and 29-32 represents a binary field whose least significant bit has a value of 25ft. The field is to be used to report the pressure altitude in the range $(25 \times N - 1000)$ (± 12.5) feet.</td></tr>
</table>
Zero (0) is transmitted each of the 13 bits if no altitude information is available.</td></tr>
<tr><td>AP</td><td>Mode S address (overlaid on parity) as described in Ref. B 4.1.</td></tr>
</table>

Comments:

This ... n ... reply ... acquisition ... is
... RL field ... to 0 and ... field ... to 1.
The ... transmitting aircraft ... will ... during ... t ... of reply.

Name: `Mode S Surveillance, Altitude Re`
Message Format: DF 4 (Surveillance), Altitu
MOPS Reference: 2.2.3.5.1

Source: Mode-S equipped aircraft
Destination: Mode-S Ground Station
Timetype: S-R

Data Representation:

DF Downlink Format 1 5	FS Flight Status 6 8	DR Downlink Request 9 13	UM Utility Message 14 19	AC Altitude Code 20 32	AP Address/-Parity 33 56

Contained Subfields

Field	**Description**
DF	Defines the type of transmission. DF transmissions are replies. 4 = Surveillance Altitude Reply
FS	Defined as follows: FS code / Mode A code change / SPI pulse present / Aircraft 0 / No / No / Airborne 1 / No / No / Ground 2 / Yes / No / Airborne 3 / Yes / No / Ground 4 / Yes / No / Either 5 / No / No / Either 6,7 / - / Not assigned / -

DR	This is used to request extraction of Downlink messages from the transponder by the interrogator. Initiates Downlink by setting the DR field to 2, 3, 6, or 7 as appropriate. The purpose is to send a Resolution Advisory Report. The codes are 0 = No Downlink Request. 1 = Request to send Comm-B message (B bit set). 2 = TCAS bit set. 3 = TCAS bit set and B bit set. 4 = Comm-B broadcast 1 available. 5 = Comm-B broadcast 2 available. 6 = TCAS bit set and Comm-B broadcast 1 available. 7 = TCAS bit set and Comm-B broadcast 2 available. 8-15 = Not assigned. 16-31 = Comm-D protocol (Ref. B 4.11.2.) Code DR=2, 3, 6, or 7 will be set as appropriate when either a TCAS Resolution Advisory or a TCAS Resolution Advisory Complement or both exist. DR codes 1-15 have a higher priority over requests associated with DR codes 16-31.
UM	Contains Control and Status of Uplinking Messages. Trasponder status readouts are used for data link multisite protocols. Not used for TCAS. The Field contains two fields Subfield — Bit Position — Meaning IIS — 14-17 — Indicates the Identity of the interrogator involved in the datalink activity. IDS — 18-19 — Shows current data-link status of the transponder. 0 = No activity 1 = Comm-B active 2 = Uplink ELM active 3 = downlink ELM active

AC	Contains the encoded altitude of the aircraft . Ref.A 2.7.13.2.5. The M-bit (bit 25) shall be zero 0 if the altitude is reported in feet. (M=1 is reserved for future use to indicate altitude reported in metric units. Format for metric reporting is undefined.) The Q-bit (bit 28) Q = 0 indicates that the Mode S altitude is reported in 100ft increments. Q = 1 indicates that the altitude is reported in 25ft increments. The possible codes are <table><tr><th>M-bit</th><th>Q-bit</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>Altitude coding according to Ref.A 2.7.13.2.5. Starting with bit 20 the sequence shall be C1, A1, C2, A2, C4, A4, M=0, B1, Q=0, B2, D2, B4, D4.</td></tr><tr><td>0</td><td>1</td><td>The 11-bit field represented by bits 20-25, 27, and 29-32 represents a binary field whose least significant bit has a value of 25ft. The field is to be used to report the pressure altitude in the range (25 × N — 1000) (±12.5) feet.</td></tr></table>Zero (0) is transmitted each of the 13 bits if no altitude information is available.
AP	Mode S address (overlaid on parity) as described in Ref. B 4.1.

Comments:

Name: `Mode-S Surveillance,` Reply
Message Format: DF 5 (Surveillance Identity Reply)
MOPS Reference:

Source: Mode-S Aircraft
Destination: Mode-S Ground Station
Timetype: S-R

Data Representation:

DF	FS	DR	UM	ID	AP
Downlink Format	Flight Status	Downlink Request	Utility Message	Identity	Address/-Parity
1 5	6 8	9 13	14 19	20 32	33 56

Fields Contained Su

Field	Description
DF	Defines the type of transmission. DF transmissions are replies. 5 = Surveillance Identity Reply
FS	Defined as follows: FS code / Mode A code change / SPI pulse present / Aircraft 0 / No / No / Airborne 1 / No / No / Ground 2 / Yes / No / Airborne 3 / Yes / No / Ground 4 / Yes / No / Either 5 / No / No / Either 6,7 / - / Not assigned / -

DR	This is used to request extraction of Downlink messages from the transponder by the interrogator. The codes are 0 = No Downlink Request. 1 = Request to send Comm-B message (B bit set). 2 = TCAS bit set. 3 = TCAS bit set and B bit set. 4 = Comm-B broadcast 1 available. 5 = Comm-B broadcast 2 available. 6 = TCAS bit set and Comm-B broadcast 1 available. 7 = TCAS bit set and Comm-B broadcast 2 available. 8-15 = Not assigned. 16-31 = Comm-D protocol (Ref. B 4.11.2.) DR=2, 3, 6 or 7 as appropriate when either a TCAS Resolution Advisory or a TCAS Resolution Advisory Complement or both exist. DR codes 1-15 have a higher priority over requests associated with DR codes 16-31.
UM	Contains Control and Status of Uplinking Messages. Trasponder status readouts are used for data link multisite protocols. Not used for TCAS. The Field contains two fields Subfield — Bit Position — Meaning IIS — 14-17 — Indicates the Identity of the interrogator involved in the datalink activity. IDS — 18-19 — Shows current data-link status of the transponder. 0 = No activity 1 = Comm-B active 2 = uplink ELM active 3 = downlink ELM active
ID	Uses Standard SSR Mode A Identity codeUses Standard SSR Mode A Identity code
AP	Mode S address (overlaid on parity) a described in Ref. B 4.1.

Comments:

Name: Mode S Squitter (All-Call Re

Message Format: DF 11 (All-Call Reply)

MOPS Reference: Detection 2.2.8.1

Source: Mode S equipped aircraft.

Destination: .

Timetype: periodic (approx. 1 2 s)

Data Representation:

DF	CA=0	AA	PI
Downlink Format	Transponder Capability	Address Announced	Parity/-Identity
1 5	6 8	9 32	33 56

Contained Subfields

Field	Description
DF	Defines the type of transmission. DF transmissions are replies. 11 = All-Call Reply
CA	Reports the capability of the transponder. The codes are 0 = No extended capability report available. 1 = Comm A/B attached capability report available. 2 = Comm A/B C attached capability report available. 3 = Comm A/B C/D attached capability report available. 4-7 = Not assigned
AA	Contains aircraft address in the clear.
PI	Ref.B 4.1.

Comments:

interrogation or ... .

Name: `Mode-S Coordination Reply`
Message Format: DF 16 (Long Special Surveillance)
MOPS Reference: Coordination Reply to TCAS 2.2.3.3.2

Source: TCAS equipped aircraft.
Destination: TCAS equipped aircraft.
Timetype: S-R

Data Representation:

DF Downlink Format 1 5	VS Vertical Status 6 6	- Not Assigned 7 8	SL Sensitivity Level 9 11	- Not Assigned 12 13	RI TCAS Capabilities 14 17	- Not Assigned 18 19	AC Altitude Code 20 32
(MV)VDS1 V-Definition 1 33 36	(MV)VDS2 V-Definition 2 37 40	(MV)ARA Active Resolution Advisories 41 54	(MV)RAC Resolution Advisory Complements 55 58	- Not Assigned 59 88	AP Address/-Parity 89 112		

Contained Subfields

Field	Description
DF	Defines the type of transmission. DF transmissions are replies. 16 = Long Special Reply
VS	Indicates if aircraft is airborne or not 0 = Aircraft airborne.

<table>
<tr><td>SL</td><td>The sensitivity level the TCAS is currently operating in. The codes are
0 = No TCAS sensitivity level reported.
1 = TCAS operating at sensitivity level 1.
2 = TCAS operating at sensitivity level 2.
3 = TCAS operating at sensitivity level 3.
4 = TCAS operating at sensitivity level 4.
5 = TCAS operating at sensitivity level 5.
6 = TCAS operating at sensitivity level 6.
7 = TCAS operating at sensitivity level 7.</td></tr>
<tr><td>RI</td><td>Information about TCAS capability is sent.
The codes are
0 = No on board TCAS or TCAS has failed or Sensitivity Level is 1.
1 = Not assigned.
2 = TA-only Mode.
3 = On-board TCAS with vertical-only resolution capability.
4 = On-board TCAS with vertical and horizontal resolution capability.</td></tr>
<tr><td>AC</td><td>Contains the encoded altitude of the aircraft . Ref.A 2.7.13.2.5.
The M-bit (bit 25) shall be zero 0 if the altitude is reported in feet. (M=1 is reserved for future use to indicate altitude reported in metric units. Format for metric reporting is undefined.)
The Q-bit (bit 28) Q = 0 indicates that the Mode S altitude is reported in 100ft increments.
Q = 1 indicates that the altitude is reported in 25ft increments.
The possible codes are
<table>
<tr><th>M-bit</th><th>Q-bit</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Altitude coding according to Ref.A 2.7.13.2.5. Starting with bit 20 the sequence shall be C1, A1, C2, A2, C4, A4, M=0, B1, Q=0, B2, D2, B4, D4.</td></tr>
<tr><td>0</td><td>1</td><td>The 11-bit field represented by bits 20-25, 27, and 29-32 represents a binary field whose least significant bit has a value of 25ft. The field is to be used to report the pressure altitude in the range $(25 \times N - 1000)$ (± 12.5) feet.</td></tr>
</table>
Zero (0) is transmitted each of the 13 bits if no altitude information is available.</td></tr>
</table>

MV	Used to transmit air-to-air resolution advisory coordination to requesting TCAS equipped aircraft. The codes are **VDS1** | **VDS2** | **Transmission** 3 | 0 | `Coordination Reply/`
(MV)VDS1	VDS1=3 for a Coordination Reply to requesting aircraft.
(MV)VDS2	VDS2=0 for a Coordination Reply to requesting aircraft.
(MV)ARA	Indicates the currently active resolution advisories (if any) generated by transmitting aircraft against one or several threat aircraft. Each bit in (MV)ARA dedicated bit representing a specific resolution advisory. The bits in (MV)ARA have the following meaning **Bit** | **Resolution** 41 | Climb 42 | Do not descend 43 | Do not descend faster than 500 FPM 44 | Do not descend faster than 1000 FPM 45 | Do not descend faster than 2000 FPM 46 | Descend 47 | Do not climb 48 | Do not climb faster than 500 FPM 49 | Do not climb faster than 1000 FPM 50 | Do not climb faster than 2000 FPM 51 | Turn left 52 | Turn right 53 | Do not turn left 54 | Do not turn right A bit set to 1 indicates that the associated advisory is active (0 indicates inactive).

(MV)RAC	Indicates the currently active resolution advisory complement (if any) received by the transmitting aircraft from all other TCAS equipped aircraft with on-board resolution capability. Each bit in (MV)RAC/ is a dedicated bit representing a specific resolution advisory complement. The bits in (MV)RAC/ have the following meaning: Bit position — Resolution complement 55 — Do not descend 56 — Do not climb 57 — Do not turn left 58 — Do not turn right A bit set to 1 indicates that the associated advisory complement is active (0 indicates inactive).
AP	Mode S address (overlaid on parity) as described in Ref. B 4.1.

Comments:

Message i upon receipt of `TCAS Coordination` .

Name: Mode-S Resolution Advisories Report, Altitude Reply
Message Format: DF 20 (Comm-B Altitude Reply)
MOPS Reference: 2.2.8.5.1

Source: Airborne TCAS
Destination: Mode-S Ground Station
Timetype: S-R

Data Representation:

DF	FS	DR	UM	AC	MB	AP
Downlink Format	Flight Status	Downlink Request	Utility Message	Altitude Code	Comm-B Message	Address/-Parity
1 – 5	6 – 8	9 – 13	14 – 19	20 – 32	33 – 88	89 – 112

(MB)BDS1	(MB)BDS2	(MB)ARA	(MB)RAC	-
B-Definition 1	B-Definition 2	Active Resolution Advisories	Resolution Advisory Complements	Not Assigned
33 – 37	37 – 40	41 – 54	55 – 58	59 – 88

Contained Subfields:

Field	Description
DF	Defines the type of transmission. DF transmissions are replies. 20 = Comm-B Altitude Reply

FS	Defined as follows: FS code / Mode A code change / SPI pulse present / Aircraft 0 / No / No / Airborne 1 / No / No / Ground 2 / Yes / No / Airborne 3 / Yes / No / Ground 4 / Yes / No / Either 5 / No / No / Either 6,7 / - / Not assigned / -
DR	This is used to request extraction of Downlink messages from the transponder by the interrogator. The codes are 0 = No Downlink Request. 1 = Request to send Comm-B message (B bit set). 2 = TCAS bit set. 3 = TCAS bit set and B bit set. 4 = Comm-B broadcast 1 available. 5 = Comm-B broadcast 2 available. 6 = TCAS bit set and Comm-B broadcast 1 available. 7 = TCAS bit set and Comm-B broadcast 2 available. 8-15 = Not assigned. 16-31 = Comm-D protocol (Ref. B 4.11.2.) DR=2,3,6 or 7 will be set as appropriate when either a TCAS Resolution Advisory or a TCAS Resolution Advisory Complement or both exist. DR codes 1-15 have a higher priority over requests associated with DR codes 16-31.
UM	Contains Control and Status of Uplinking Messages. Trasponder status readouts are used for data link multisite protocols. Not used for TCAS. The Field contains two fields: Subfield / Bit Position / Meaning IIS / 14-17 / Indicates the Identity of the interrogator involved in the datalink activity. IDS / 18-19 / Shows current data-link status of the transponder. 0 = No activity 1 = Comm-B active 2 = uplink ELM active 3 = downlink ELM active

<table>
<tr><td>AC</td><td>Contains the encoded altitude of the aircraft . Ref.A 2.7.13.2.5.
The M-bit (bit 25) shall be zero 0 if the altitude is reported in feet. (M=1 is reserved for future use to indicate altitude reported in metric units. Format for metric reporting is undefined.)
The Q-bit (bit 28) Q = 0 indicates that the Mode S altitude is reported in 100ft increments.
Q = 1 indicates that the altitude is reported in 25ft increments.
The possible codes are
<table>
<tr><th>M-bit</th><th>Q-bit</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Altitude coding according to Ref.A 2.7.13.2.5. Starting with bit 20 the sequence shall be
C1, A1, C2, A2, C4, A4, M=0, B1, Q=0, B2, D2, B4, D4.</td></tr>
<tr><td>0</td><td>1</td><td>The 11-bit field represented by bits 20-25, 27, and 29-32 represents a binary field whose least significant bit has a value of 25ft. The field is to be used to report the pressure altitude in the range $(25 \times N - 1000)$ (± 12.5) feet.</td></tr>
</table>
Zero (0) is transmitted each of the 13 bits if no altitude information is available.</td></tr>
<tr><td>MB</td><td>Used to transmit TCAS resolution advisories information.</td></tr>
<tr><td>MB</td><td>The following subfields are use if BDS1=3 and BDS2=0, the combination of which is equivalent to BDS=48. The bit fields 60-88 are not assigned in this case.</td></tr>
<tr><td>(MB)BDS1</td><td>BDS1 should be set to 3 to indicate that a Resolution Advisories Report is contained in MB</td></tr>
<tr><td>(MB)BDS2</td><td>BDS2 should be set to 0 to indicate that a Resolution Advisories Report is contained in MB</td></tr>
</table>

<table>
<tr>
<td>(MB)ARA</td>
<td>Indicates the currently active resolution advisories generated by own aircraft against one or more threat aircraft.
Each bit in (MB)ARA dedicated bit representing a specific resolution advisory. The bits in (MB)ARA have the following meaning

<table>
<tr><th>Bit</th><th>Resolution advisory</th></tr>
<tr><td>41</td><td>Climb</td></tr>
<tr><td>42</td><td>Do not descend</td></tr>
<tr><td>43</td><td>Do not descend faster than 500 FPM</td></tr>
<tr><td>44</td><td>Do not descend faster than 1000 FPM</td></tr>
<tr><td>45</td><td>Do not descend faster than 2000 FPM</td></tr>
<tr><td>46</td><td>Descend</td></tr>
<tr><td>47</td><td>Do not climb</td></tr>
<tr><td>48</td><td>Do not climb faster than 500 FPM</td></tr>
<tr><td>49</td><td>Do not climb faster than 1000 FPM</td></tr>
<tr><td>50</td><td>Do not climb faster than 2000 FPM</td></tr>
<tr><td>51</td><td>Turn left</td></tr>
<tr><td>52</td><td>Turn right</td></tr>
<tr><td>53</td><td>Do not turn left</td></tr>
<tr><td>54</td><td>Do not turn right</td></tr>
</table>

A bit set to 1 indicates that the associated advisory is active (0 indicates inactive).</td>
</tr>
<tr>
<td>(MB)RAC</td>
<td>Indicates the currently active resolution advisory complement (if any) received by the transmitting aircraft from all other TCAS equipped aircraft with on-board resolution capability.
Each bit in (MB)RAC dedicated bit representing a specific resolution advisory complement. The bits in (MB)RAC have the following meaning

<table>
<tr><th>Bit</th><th>Resolution advisory complement</th></tr>
<tr><td>55</td><td>Do not descend</td></tr>
<tr><td>56</td><td>Do not climb</td></tr>
<tr><td>57</td><td>Do not turn left</td></tr>
<tr><td>58</td><td>Do not turn right</td></tr>
</table>

A bit set to 1 indicates that the associated advisory complement is active (0 indicates inactive).</td>
</tr>
<tr>
<td>AP</td>
<td>Mode S address (overlaid on parity) as described in Ref. B 4.1.</td>
</tr>
</table>

Comments:

[illegible] i upon receipt of a UF=4,5,20 or 21 [illegible] 19 [illegible]
DI=7, [illegible] 19, DI=7 [illegible] [illegible] 0.

Name: Mode-S Comm-B Reply, Altitude Report
Message Format: DF 20 (Comm-B, Altitude)
MOPS Reference: 2.2.5.
Source: Airborne TCAS
Destination: Mode-S Ground Station
Timetype: S-R

Data Representation:

DF Downlink Format 1 5	FS Flight Status 6 8	DR Downlink Request 9 13	UM Utility Message 14 19	AC Altitude Code 20 32	MB Comm-B Message 33 88	AP Address/-Parity 89 112
(MB)BDS1 B-Definition 1 33 37	(MB)BDS2 B-Definition 2 37 40	(MB)- Not assigned. 41 44	(MB)ACS Comm-A Capability Subfield 45 64	(MB)BCS Comm-B Capability Subfield 65 80	(MB)ECS Extended Capability Subfield 81 88	

If (MB)BDS1 is set to 1 and (MB)BDS2 is set to 0:

((MB)BCS)-ORC On-Board Resolution Capability 69 70	((MB)BCS)-FIRC Flight Identification Report Capability 65 65

Contained Subfields

Field	**Description**

DF	Defines the type of transmission. DF transmissions are replies. 20 = Comm-B Altitude Reply
FS	Defined as follows FS code — Mode A code change — SPI pulse present — Aircraft 0 — No — No — Airborne 1 — No — No — Ground 2 — Yes — No — Airborne 3 — Yes — No — Ground 4 — Yes — No — Either 5 — No — No — Either 6,7 — - — Not assigned — -
DR	This is used to request extraction of Downlink messages from the transponder by the interrogator. The codes are 0 = No Downlink Request. 1 = Request to send Comm-B message (B bit set). 2 = TCAS bit set. 3 = TCAS bit set and B bit set. 4 = Comm-B broadcast 1 available. 5 = Comm-B broadcast 2 available. 6 = TCAS bit set and Comm-B broadcast 1 available. 7 = TCAS bit set and Comm-B broadcast 2 available. 8-15 = Not assigned. 16-31 = Comm-D protocol (Ref. B 4.11.2.) DR codes 1-15 have a higher priority over requests associated with DR codes 16-31.
UM	Contains Control and Status of Uplinking Messages. Trasponder status readouts are used for data link multisite protocols. Not used for TCAS. The Field contains two fields Subfield — Bit Position — Meaning IIS — 14-17 — Indicates the Identity of the interrogator involved in the datalink activity. IDS — 18-19 — Shows current data-link status of the transponder. 0 = No activity 1 = Comm-B active 2 = uplink ELM active 3 = downlink ELM active

<table>
<tr><td>AC</td><td>Contains the encoded altitude of the aircraft . Ref.A 2.7.13.2.5.
The M-bit (bit 25) shall be zero 0 if the altitude is reported in feet. (M=1 is reserved for future use to indicate altitude reported in metric units. Format for metric reporting is undefined.)
The Q-bit (bit 28) Q = 0 indicates that the Mode S altitude is reported in 100ft increments.
Q = 1 indicates that the altitude is reported in 25ft increments.
The possible codes are
<table>
<tr><th>M-bit</th><th>Q-bit</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Altitude coding according to Ref.A 2.7.13.2.5. Starting with bit 20 the sequence shall be
C1, A1, C2, A2, C4, A4, M=0, B1, Q=0, B2, D2, B4, D4.</td></tr>
<tr><td>0</td><td>1</td><td>The 11-bit field represented by bits 20-25, 27, and 29-32 represents a binary field whose least significant bit has a value of 25ft. The field is to be used to report the pressure altitude in the range $(25 \times N - 1000)$ (± 12.5) feet.</td></tr>
</table>
Zero (0) is transmitted each of the 13 bits if no altitude information is available.</td></tr>
<tr><td>MB</td><td>Used to send extended capability information to Mode S sensors</td></tr>
<tr><td>MB</td><td>The following subfields are used if BDS1=1 and BDS2=0 or BDS2=1</td></tr>
<tr><td>(MB)BDS1</td><td>Code is 1 for extended capability reports.</td></tr>
<tr><td>(MB)BDS2</td><td>Allows more complex Mode S installations to report additional Capabilities in various formats.
The codes are
0 = Basic Report.
1 = Additional Report (Ref B. 4.6.2.1).
2-15 = Not assigned</td></tr>
<tr><td>(MB)ACS</td><td>Reports the data link services supported by this installation.
ACS =1 for all TCAS equipped aircraft.
If ACS=0 then no Comm-A datalink services are supported.</td></tr>
<tr><td>(MB)BCS</td><td>Reports the installed data sources that can be accessed by the ground for transmission via a ground initiated Comm-B.
BCS =1 for all TCAS equipped aircraft.
If BCS=0 then no data is accessible by a ground-initiated Comm-B.</td></tr>
</table>

((MB)BCS)-ORC	These bits form a capability code subfield which indicates on-board aircraft resolution advisory generation capability. The codes are 0 = No on-board resolution advisory generation capability. 1 = An on-board vertical only resolution advisory generation capability exists. 2 = An on-board vertical and horizontal resolution advisory generation capability exists. 3 = not assigned.
((MB)BCS)-FIRC	If BDS2=0 (sending the basic report) this bit is set to 1 if the aircraft has the capability to transmit a Flight Identification Report. Otherwise, bit 65 remains in (MB)BCS.
(MB)ECS	Reports the ELM capability of the installation. If ELM=0, no ELM data link services are supported.
AP	Mode S address (overlaid on parity) as described in Ref. B 4.1.

Comments:

Message is sent upon re… Ground Station Mode S specific…ion to learn the data link capability of the aircraft.

Name: Mode-S Resolution Advisories Report

Message Format: DF 21 (Comm-B Identity Reply)

MOPS Reference: 2.2.5.

Source: Airborne TCAS

Destination: Mode-S Ground Station.

Timetype: S-R

Data Representation:

DF	FS	DR	UM	ID	MB	AP
Downlink Format	Flight Status	Downlink Request	Utility Message	Identity	Comm-B Message	Address/-Parity
1 5	6 8	9 13	14 19	20 32	33 88	89 112

(MB)BDS1	(MB)BDS2	(MB)ARA	(MB)RAC	-
B-Definition 1	B-Definition 2	Active Resolution Advisories	Resolution Advisory Complements	Not Assigned
33 36	37 40	41 54	55 58	59 88

Contained Subfields:

Field	Description
DF	Defines the type of transmission. DF transmissions are replies. 21 = Comm-B Identity Reply

<table>
<tr><td>FS</td><td>Defined as follows:
<table>
<tr><td>FS code</td><td>Mode A code change</td><td>SPI pulse present</td><td>Aircraft</td></tr>
<tr><td>0</td><td>No</td><td>No</td><td>Airborne</td></tr>
<tr><td>1</td><td>No</td><td>No</td><td>Ground</td></tr>
<tr><td>2</td><td>Yes</td><td>No</td><td>Airborne</td></tr>
<tr><td>3</td><td>Yes</td><td>No</td><td>Ground</td></tr>
<tr><td>4</td><td>Yes</td><td>No</td><td>Either</td></tr>
<tr><td>5</td><td>No</td><td>No</td><td>Either</td></tr>
<tr><td>6,7</td><td>-</td><td>Not assigned</td><td>-</td></tr>
</table></td></tr>
<tr><td>DR</td><td>This is used to request extraction of Downlink messages from the transponder by the interrogator.
The codes are
<table>
<tr><td>0</td><td>=</td><td>No Downlink Request.</td></tr>
<tr><td>1</td><td>=</td><td>Request to send Comm-B message (B bit set).</td></tr>
<tr><td>2</td><td>=</td><td>TCAS bit set.</td></tr>
<tr><td>3</td><td>=</td><td>TCAS bit set and B bit set.</td></tr>
<tr><td>4</td><td>=</td><td>Comm-B broadcast 1 available.</td></tr>
<tr><td>5</td><td>=</td><td>Comm-B broadcast 2 available.</td></tr>
<tr><td>6</td><td>=</td><td>TCAS bit set and Comm-B broadcast 1 available.</td></tr>
<tr><td>7</td><td>=</td><td>TCAS bit set and Comm-B broadcast 2 available.</td></tr>
<tr><td>8-15</td><td>=</td><td>Not assigned.</td></tr>
<tr><td>16-31</td><td>=</td><td>Comm-D protocol (Ref. B 4.11.2.)</td></tr>
</table>
Code DR=2, or 7 will be set as appropriate when either a TCAS Resolution Advisory or a TCAS Resolution Advisory Complement or both exist.
DR codes 1-15 have a higher priority over requests associated with DR codes 16-31.</td></tr>
<tr><td>UM</td><td>Contains Control and Status of Uplinking Messages.
Trasponder status readouts are used for data link multisite protocols. Not used for TCAS.
The Field contains two fields
<table>
<tr><td>Subfield</td><td>Bit Position</td><td>Meaning</td></tr>
<tr><td>IIS</td><td>14-17</td><td>Indicates the Identity of the interrogator involved in the datalink activity.</td></tr>
<tr><td>IDS</td><td>18-19</td><td>Shows current data-link status of the transponder.
0 = No activity
1 = Comm-B active
2 = uplink ELM active
3 = downlink ELM active</td></tr>
</table></td></tr>
<tr><td>ID</td><td>Uses Standard SSR Mode A Identity code</td></tr>
<tr><td>MB</td><td>Used to transmit TCAS resolution advisories, TCAS resolution advisory complements, and extended capability information to Mode S sensors</td></tr>
</table>

<table>
<tr><td>MB</td><td>The following subfields are use if BDS1=3 and BDS2=0 the combination of which is equivalent to BDS=48. The bit fields 60-88 are not assigned in this case.</td></tr>
<tr><td>(MB)BDS1</td><td>BDS1 should be set to 3 to indicate that a Resolution Advisories Report is contained in MB</td></tr>
<tr><td>(MB)BDS2</td><td>BDS2 should be set to 0 to indicate that a Resolution Advisories Report is contained in MB</td></tr>
<tr><td>(MB)ARA</td><td>Indicates the currently active resolution advisories generated by own aircraft against one or more threat aircraft.
Each bit in (MB)ARA dedicated bit representing a specific resolution advisory. The bits in (MB)ARA have the following meaning
<table>
<tr><th>Bit</th><th>Resolution advisory</th></tr>
<tr><td>41</td><td>Climb</td></tr>
<tr><td>42</td><td>Do not descend</td></tr>
<tr><td>43</td><td>Do not descend faster than 500 FPM</td></tr>
<tr><td>44</td><td>Do not descend faster than 1000 FPM</td></tr>
<tr><td>45</td><td>Do not descend faster than 2000 FPM</td></tr>
<tr><td>46</td><td>Descend</td></tr>
<tr><td>47</td><td>Do not climb</td></tr>
<tr><td>48</td><td>Do not climb faster than 500 FPM</td></tr>
<tr><td>49</td><td>Do not climb faster than 1000 FPM</td></tr>
<tr><td>50</td><td>Do not climb faster than 2000 FPM</td></tr>
<tr><td>51</td><td>Turn left</td></tr>
<tr><td>52</td><td>Turn right</td></tr>
<tr><td>53</td><td>Do not turn left</td></tr>
<tr><td>54</td><td>Do not turn right</td></tr>
</table>
A bit set to 1 indicates that the associated advisory is active (0 indicates inactive).</td></tr>
<tr><td>(MB)RAC</td><td>Indicates the currently active resolution advisory complement (if any) received by the transmitting aircraft from all other TCAS equipped aircraft with on-board resolution capability.
Each bit in tt(MB)RAC is a dedicated bit representing a specific resolution advisory complement. The bits in (MB)RAC have the following meaning
<table>
<tr><th>Bit</th><th>Resolution advisory complement</th></tr>
<tr><td>55</td><td>Do not descend</td></tr>
<tr><td>56</td><td>Do not climb</td></tr>
<tr><td>57</td><td>Do not turn left</td></tr>
<tr><td>58</td><td>Do not turn right</td></tr>
</table>
A bit set to 1 indicates that the associated advisory complement is active (0 indicates inactive).</td></tr>
</table>

Comments:

Message i upon receipt of a UF=4,5,20 or 21 ... t 19

Name: Mode-S ... Capability Report
Message Format: DF 21 (Comm-B ...)
MOPS Reference: 2.2...

Source: ... TC
Destination: Mode-S Ground Station.
Timetype: S-R

Data Representation:

DF	FS	DR	UM	ID	MB	AP
Downlink Format	Flight Status	Downlink Request	Utility Message	Identity	Comm-B Message	Address/-Parity
1 5	6 8	9 13	14 19	20 32	33 88	89 112

(MB)BDS1	(MB)BDS2	(MB)-	(MB)ACS	(MB)BCS	(MB)ECS
B-Definition 1	B-Definition 2	Not Assigned.	Comm-A Capability Subfield	Comm-B Capability Subfield	Extended Capability Subfield
33 37	37 40	41 44	45 64	65 80	81 88

If (MB)BDS1 is set to 1 and (MB)BDS2 is set to 0:

((MB)BCS)-ORC	((MB)BCS)-FIRC
On-Board Resolution Capability	Flight Identification Report Capability
69 70	65 65

Contained Subfields

Field	Description

DF	Defines the type of transmission. DF transmissions are replies. The codes are 21 = Comm-B Identity Reply
FS	Defined as follows: FS code — Mode A code change — SPI pulse present — Aircraft 0 — No — No — Airborne 1 — No — No — Ground 2 — Yes — No — Airborne 3 — Yes — No — Ground 4 — Yes — No — Either 5 — No — No — Either 6,7 — - — Not assigned — -
DR	This is used to request extraction of Downlink messages from the transponder by the interrogator. The codes are 0 = No Downlink Request. 1 = Request to send Comm-B message (B bit set). 2 = TCAS bit set. 3 = TCAS bit set and B bit set. 4 = Comm-B broadcast 1 available. 5 = Comm-B broadcast 2 available. 6 = TCAS bit set and Comm-B broadcast 1 available. 7 = TCAS bit set and Comm-B broadcast 2 available. 8-15 = Not assigned. 16-31 = Comm-D protocol (Ref. B 4.11.2.)
UM	Contains Control and Status of Uplinking Messages. Trasponder status readouts are used for data link multisite protocols. Not used for TCAS. The Field contains two fields Subfield — Bit Position — Meaning IIS — 14-17 — Indicates the Identity of the interrogator involved in the datalink activity. IDS — 18-19 — Shows current data-link status of the transponder. 0 = No activity 1 = Comm-B active 2 = uplink ELM active 3 = downlink ELM active
ID	Uses Standard SSR Mode A Identity code
MB	Used to transmit TCAS resolution advisories, TCAS resolution advisory complements, and extended capability information to Mode S sensors
MB	The following subfields are used if the extended capability information is transmitted when BDS1=1 and BDS2=0 or BDS2=1

(MB)BDS1	Code is 1 for all extended capability reports.
(MB)BDS2	Allows more complex Mode S installations to report additional Capabilities in various formats. The codes are 0 = Basic Report. 1 = Additional Report (Ref B. 4.6.2.1). 2-15 = Not assigned
(MB)ACS	Reports the data link services supported by this installation. [illegible] =1 for all TCAS equipped aircraft. If ACS=0 then no Comm-A datalink services are supported.
(MB)BCS	Reports the installed data sources that can be accessed by the ground for transmission via a ground initiated Comm-B. [illegible] =1 for all TCAS equipped aircraft. If BCS=0 then no data is accessible by a ground-initiated Comm-B.
((MB)BCS)-ORC	These bits form a capability code subfield which indicates on-board aircraft resolution advisory generation capability. The codes are 0 = No on-board resolution advisory generation capability. 1 = An on-board vertical only resolution advisory generation capability exists. 2 = An on-board vertical and horizontal resolution advisory generation capability exists. 3 = not assigned.
((MB)BCS)-FIRC	If BDS2=0 (sending the basic report) this bit is set to 1 if the aircraft has the capability to transmit a Flight Identification Report. Otherwise, bit 65 remains in [illegible]
(MB)ECS	Reports the ELM capability of the installation. If ELM=0 no ELM data link services are supported.
AP	Mode S address (overlaid on parity) as described in Ref. B 4.1.

Comments:

Message [illegible] upon re[illegible] Ground Station M[illegible] to learn t[illegible] the aircraft.

Receiver, Transmitter, Antennas

Range Accuracy

The error a TCAS range measurement of an target-of-interest ... national ... for ... and ... 25 feet, 1 sigma ... d interrogation ... are at least 10 dB ... receiver ... .

Bearing Accuracy

The error a TCAS measurement of target ... angle ... and ... elevation angle from +10 degree to -10 degree using Mode S ... remaining at least ... no greater than 9 degree, 1 sigma and 27 degree ... antenna ... 4-foot or larger diameter circular ... ground plane and ... a controlled antenna test range ... .

Maximum Radiated Output Power

The maximum RF Total Radiated Power ... 50 dBm (100 Watts) ... full power operation.

The Total Radiated Power ... :

TRP = P* G(BW/360 deg)

... the RF reference point, G is the peak antenna gain ... at 0 degree elevation relative to a ... is the 3 dB ... . The specified maximum limit on TRP is intended to ... an ... from causing premature interference limiting and ... an unnecessary ... dynamic range of t... (... 2.1).

Unwanted Output Power

When the TCAS interrogator is ... RF power at 1030 ± ... 70 dBm. The inacti... interrogation ... 10 ... ds, ... perio... ... interrogation ... .

NOTE: This restriction is ... TCAS ... not ... permit ... Mode S transponders from meeting its ... requirement ... coupling ... 20 dB.

Interrogation Spectrum

The spectrum of a TCAS interrogation shall not exceed the following:

Frequency Difference (MHz from Carrier)	Maximum Relative Response (dB down)
≥ 4 and < 6	6
≥ 6 and < 8	11
≥ 8 and < 10	15
≥10 and < 20	19
≥20 and < 30	31
≥30 and < 40	38
≥40 and < 50	43
≥50 and < 60	47
≥60 and < 90	50
≥90	60

Interrogation Jitter

The transmit time of each ... TCAS ... interrogation ... shall be ... interrogation ... intentionally ... jitter ... to prevent ... interrogator ... 2.0 ...). The maximum ... value of t ... exceed ±1% ... nominal up ... .

NOTE: It is not necessary to intentionally jitter Mode S interrogations ... nature of the Mode S interrogation scheduling process.

Transmit Frequency And Tolerance

The transmit ... 1030 ±0.01 MHz.

Mode C Transmissions ... interrogation ... equipment employ ... Mode C ... format ... pulses P1, P3, and P4 ... normally ... Whisper/Shout ...

Signal ... 1 (Control of Synchronous Garble by Whisper/Shout

...a P2 pulse ... a separate control pattern. The format ... in Figure 3-1. The pulse spacing ... as the leading edge ... system time ... Section 2.2 3.3. The amplitude of P3 shall be within 0.5 dB of the amplitude of P1, and the amplitude of P4 shall be within 0.5 dB of the amplitude of P3.

MODE A/C PULSES
(All values in microseconds)

Pulse Designator	Pulse Duration	Duration Tolerance	Rise Time Min.	Rise Time Max.	Decay Time Min.	Decay Time Max.
S1,P1,P2,P3,P4	0.8	±0.05	0.05	0.1	0.05	0.2

The pulse spacing tolerances shall be as follows:

S1 to P1 2.00 ±0.10 microseconds,
P1 to P2 2.00 ±0.10 microseconds,
P1 to P3 21.00 ±0.10 microseconds,
P3 to P4 2.00 ±0.04 microseconds.

NOTE: The pulse spacing tolerances are smaller than the tolerances in Ref. B in order to provide margin for ... multipath.

Mode S Transmissions. The Mode S transmission shall consist of P1, P2, and P6 as shown in Figure 3-2.

MODE S PULSES
(All values in microseconds)

Pulse Designator	Pulse Duration	Duration Tolerance	Rise Time Min.	Rise Time Max.	Decay Time Min.	Decay Time Max.
P1,P2	0.8	±0.05	0.05	0.1	0.05	0.2
P6 (Short)	16.25	±0.25	0.05	0.1	0.05	0.2
P6 (Long)	30.25	±0.25	0.05	0.1	0.05	0.2

The short (16.25-microsecond) or long (30.25-microsecond) P6 pulse shall be ... intrapulse ... modulation ... 180-degree phase ... of the carrier ... The first phase reversal ... P6 ... The presence of a phase reversal ... indicates ...

NOTE: The ... phase ... timing difference ... Mode S ... interrogation.

The duration of a phase reversal in P6 shall be less than 0.08 microsecond. The phase reversal is measured between the 10-degree and 170-degree points of the phase transition. The interval between the 10-degree and 170-degree points ... shall be less than 0.08 microsecond.

The tolerance on the 0- and 180-degree phase relationship in P6 shall be ± 5 degrees.

The 90-degree point of each phase reversal in P6 shall occur only at a time (N x 0.25) ± 0.02 microsecond (N ≥ 2) after the 90-degree point of the sync phase reversal.

NOTE: 56 or 112 data phase reversals occur between 1.25 and 3 0.25 microseconds ... P6 pulse.

The spacing from P1 to P2 shall be 2 ± 0.04 microseconds from leading edge to leading edge. The spacing from the leading edge of P2 to the 90-degree point of the sync phase reversal in P6 shall be 2.75 ± 0.04 microseconds. The leading edge of P6 shall occur 1.25 ± 0.04 microseconds before the sync phase reversal.

NOTE: The P1-P2 pair preceding P6 ... triggering of ATCRBS transponders ... (S interrogation 2.0 ...). A series of chips containing the information in P6 start 0.5 microsecond after the sync phase reversal. Each chip is of 0.25 microsecond duration ... "1". There are either 56 or 112 chips. The last chip is followed by 0.5 microsecond ... the trailing edge of P6 from interrogation ... process.

The radiated amplitude of P2 and the initial first microsecond of P6 shall be greater than the radiated amplitude of P1 minus 0.25 dB. The ... amplitude variation ... of P6 shall be less than 0.25 dB.

NOTE: The tolerance ... the ... positioning ... of the ... signal-in-space tolerance defined in Ref. B in order to ... of ... distortion ... multipath ... section.

Compatibility With Mode S Transponders

TCAS ... compatibility ... Mode S transponders (... 1.1). Two potential areas ... incompatibility ... example ... interference.

...

specific document to control or test the interface

TCAS manufacturer to

in fact interface in , logical ,protocol

and operating con

eration of the functional entitie fulfille

combination of operating con .

The may be complicated to full compati normal f system

input/output .

s TCAS equipment re document , are met the Mode

ind by indicate subsequent

near it MTL .

b. The electromagnetic potential incompati inter-
ference generate

frequenc signal rom other

formance of th .

TCAS equipment

operating in conjunction operating Mode S

transponder ion of t Mode S tran

. The Mode S transmitter line

transmitter y pulse of a repl

minimum e te o

reply interrogator

.

Aircraft Suppression Bus

TCAS equipment 70± micro pul to

other aircraft equipment transmitter interrogation tran

from the top-mounte . a TCAS equipment is a 90± 1

micro pulse to ot equipment f interrogation f rom the

bottom antenna.

If the TCAS equipment i to accept an to interference

suppre pul from other electronic e the aircraft (

while equipment i regain normal

within after t 15 micro supplie

interference .

... the interrogation to ... TCAS
... interrogation from ... (... 2.2) ...
... potential ... purpose ...
... parameter ... response ...
... station ... recommen...
of interference ... couple ... couple.
... failure from ... all
... interf... response ... pul...

Data Integrity Requirements

... interface communication in the normal operational aircraft environment ... error rate ... undetected error in ... und... 10^{-7} ... tran... compliance ... simulated operational environment or ... characteri... interface tec...

Differential Delay

... mean tran... bottom antenna channel ... -to-antenna tran... 0.5 ...

The ... mean ... bottom antenna channel ... -to-antenna tran... 0.5 ... TL + 3 dB ... 21 dBm.

In-Band Acceptance

Given ... signal in the ... of ... interference ... or o... the minimum trigger level (MTL) is defined as the minimum input po... that re... a 90% ... of ... received.

a. The MTL ... Mode S ... range of 1087 to 1093 MHz ... 74 dBm ±2 dB.

> **NOTE:** ... margin ... of ... aircraft co-altitude ... at a range of 14 nmi (... 1.2).

b. For an input signal power level of -78 dBm or less, no more than 10% of ATCRBS and Mode S signals shall be decoded.

The decoding ratio shall be at least 90% for ATCRBS and Mode S signals between MTL and -21 dBm.

If the TCAS antenna gain is not as specified in the section Antenna System on page 210, the MTL shall be adjusted to account for the antenna gain. For example, if the antenna gain is 3 dB below the specified gain, the nominal MTL would be -71 dBm.

Out-of-Band Rejection

The selectivity of the receiver is specified as the receiver output level for an RF CW signal at the receiver input relative to the output level at the center frequency as a function of signal input frequency offset.

Input Signal Frequency Difference from 1,090 MHz	Output Signal Level Relative to 1,090 MHz
± 5.5 MHz	-3 dB
± 10 MHz	-20 dB
± 15 MHz	-40 dB
± 25 MHz	-60 dB

Reply Detection And Decoding

The following characteristics shall apply over the RF signal input level range from MTL to -21 dBm:

Mode C Reply Reception Acceptable performance shall be achieved for signals having the following characteristics:

Pulse amplitude variation: up to ± 2 dB relative to F1 pulse amplitude.
Pulse rise time: 0.05 to 0.1 microseconds.
Pulse decay time: 0.05 to 0.2 microseconds.

a. Description of Mode C Reply Signal: The Mode C reply signal consists of a pair of framing pulses spaced 20.3 ± 0.1 microseconds apart. The code pulse positions are spaced 1.45 microseconds from the first framing pulse at multiples of 1.45 microseconds after the first framing pulse. Each code pulse has a tolerance of ± 0.1 microseconds relative to the first framing pulse and ± 0.15 microseconds relative to every other pulse.

the reply. All pulses ... 0.45 ± ... microseconds. A one or zero in the reply code is indicated by the presence or absence of a code pulse respectively.

b. Criteria for Mode C Reply Detection: Mode C ... leading edges ... The occurrence of a leading edge ... directly ... of a clear leading edge ... inferred from ... trailing ... overlapping pulses. An actual leading edge ... as ... for which ... signal rise ... 48 dB per microsecond ... received ... AND 0.1 microsecond ... rate of rise ... 48 dB per microsecond. An inferred leading edge ... as ... a leading edge ... account for ... overlapping pulses. ... criterion for reception of an Mode C ... the occurrence of a pair of leading edges spaced 20.3 ± 0.1 microseconds apart. ... if the ... from 20 ... microseconds 0.242 microseconds or more ... OR if the ... 0.242 microseconds. A Mode C pulse ... accepted ... occur ... 0.1 microsecond of a nominal code position ... The ... times of occurrence of its leading edge from a nominal code position ... 0.242 microseconds or more OR if the ... 0.242 microseconds or less.

c. Criteria for Acceptance of Garbled Mode C Replies. The TCAS equipment shall ... correctly ... detecting replies ... at least ... overlapping Mode C replies ... interleaved ... fall into the ... between the ... reply ... pulses. The equipment ... determining arrival ... replies ... at least ... overlapping Mode C replies ... code pulses ... apart. ... minimum ... correctly ... overlapping ... specified ... as follows:

Reply 1

Reply Code = 6530

Received Amplitude = -60 dBm

RF Frequency = 1090 MHz

Probability

Decode Correct = 50% *

Reply 2

Reply Code = 4[illegible]

Received Amplitude = -63 dBm

RF Frequency = 1084 MHz

Range Relative to

Reply 1 = Randomly Continuous Variable from Reply 1 range plus 21 microseconds to 0 to Reply 1 range minus 21 microseconds

Probability

Decode Correct = 50% *

Reply 3

Reply Code = 6[illegible]

Received Amplitude = -57 dBm

RF Frequency = 1093 MHz

Range Relative to

Reply 1 = Randomly Continuous Variable from Reply 1 range minus 21 microseconds to 0 to Reply 1 range plus 21 microseconds

Probability

Decode Correct = 50% *

* Where averaged over all [illegible] of Reply 1 and Replies 2 and 3.

NOTE: Attempting to [illegible] overlapping replies [illegible] TCAS equipment [illegible] pulses [illegible] clearly [illegible] of [illegible] the same amplitude [illegible].

The TCAS [illegible] equipment [illegible] reliable [illegible] apping [illegible] replies [illegible] pulses. The [illegible] specific [illegible] replies [illegible] specific [illegible] of 6 [illegible] pulses.

d. Phantom R ejections ... replie ... s replie .

NOTE: ...ommen ... that t ... Mode C ... unction ... formulation of a long ... Mode S ... reply ... Mode S reply ... M ... d ... g perio . ... pre- ... t ... ode S reply from generating a string of fal ... Mode C ... replie ... Mode C ... ning .

Mode S ... r and Reply Reception All performance ... requirement be met for ... pul ... characteri :

...amplitu ... ation: up to ±2 dB, relati ... amplitu ... pul .

... time : ... to 0.1 micro ...

... time: ... to 0.2 micro ...

a. Description of Mode S Recei ... Signal : The Mode S ... con ... a pream ... . The pream ... of four 0.5 ± ... - micro ... . 1, 3.5, ... micro ... from t ... . The ... 8 micro ... after the fir ... pul . ... 56 or 112 ... micro ... . A pul ... 0.5 ... micro ... of eac ... ate a ... of eac ... in ... . If a ... one ... ter ... pul ... merge ... a 1 ... micro ... .

b. Criteria for Preamble Acceptance : The criterion f ... or reception of a Mode S ... nal ... detection of a Mode S pream . A pream ... accepte ... pul ... orm ... at lea 75% of it ... nominal ... pul ... ± ... micro ... nominal po ... to ... pul ... lea ... in 2.2.4.1 b.) that ... ±... s ... nominal e ... and ... no earlier lea ... pul .

c. Criteria for Data Acceptance in ... ter an ... sion : The initi al ... detection of the Mode S ... a bymonitoring Mode S transponder replies, which contain the Mode S all-call format. Mode S replies contain 56 data bits. ... of the Mode S ... during the ... a 0.5-... replies ... to determine the ... signal amplitude at the center ... position for that bit.

In the all-call format ... parity field ... the Mode S transponder ... detect and correct ... error. One ... of implementing error correction ... in Appendix ... . The TCAS equipment only ... contain the code 001 in the first 5 bits ... Mode S altitude ... . Detection of the ... interrogating ... to elicit a reply ... monitoring Mode S ... replies ... DF = 0 or DF = 4 format. Any ... accepted ... Mode S altitude reply: the first 5 bits ... contain the code 00000 ... 000, AND the 26th bit ... ZERO, AND the 28th bit ... ZERO, AND bits 20, 22 and 24 do not ... the illegal codes below:

Illegal Altitude Codes

Bit No. 20 (C1)	22 (C2)	24 (C3)
0	0	0
1	0	1
1	1	1

FIG 2-6

AND no more than 4 ... more than ... , execution ... fail the ... :

... the ... micro ... to determine if the ... minimum triggering level ... . two ... pulse ... difference in the ... sample ... DMT ... agree ... value of the .

NOTE: Selection criteria for Mode S replies ... pre-filtering should ... Mode S replies ... the track file for non-Mode S ...

Interrogation Link Interference

The equipment ... signals (including multipath 2.1) from ... interrogator ...

... specified interrogation ... on page 159 (↑2.1) ... multipath to an acceptable ... interrogator ... power le... to the ... MTL ... signal-to-... multipath ... multipath ... interrogation.

Reply Link Interference

The equipment ... multipath 2.1). Specifically, equipment ... replies ... simultaneous ... multipath level:

a. ATCRBS Reply Link

Minimum signal level = MTL + 11dB
Maximum multipath = Reply level - 10dB

b. Mode S Reply Link

Minimum signal level = MTL + 8db
Maximum multipath = Reply level - 7dB

... multipath ... reply.

Narrow Pulse Discrimination. ... for rejecting ... multipath ... pulses ... of less than 0.3 microseconds.

TACAN and DME Discrimination. ... rejecting ... multipath ... rise time ... 0.5 microseconds.

Narrow Pulse Rejection

The TCAS equipment ... ATCRBS ... to a ... 0.242 ... micro ...

TACAN and DME Signal Rejection

The TCAS equipment ... receive ... time ... 0.5 ... micro ...

Control Of ATCRBS Synchronous Garble

The equipment ... controlling AT ... (↑2.1) to a ... antenna ... requirement ... on ... interference ... ment ...

A ... a Mode C ... interrogation ... directional transmitting antenna. The degree ... erence ... depend ... interrogation ... the directionality of the antenna. The TCAS equipment ... interrogation ... minimum ... interrogation ... 10 ... minimum ... requirement ... antenna. TCAS ... directional ... interrogation antenna.

The ... parameter ... interrogation ... elect manufacturer to ... control of ... performance of ... antenna ... performance ... interrogation ...

19 and ... directional interrogation antenna ... respectively.

Control Of Synchronous Garble By Whisper/Shout To control AT... interference and to re... multipath the (↑interrogation lin... sequence of interrogation ... . Each of t... interro- gation ...

1) ... the P 1 pul . The combination of S1 and P ... . S1 shall ... 1. The minimum ... interrogation ... interrogation ...

- **Assum...** ... aircraf ... or ... interrogation . A ... typical population of ATCRBS ... range ... a ... ef- fecti ... ation ... recei ... antenna ... typicall ... population to two ... interrogation off by ... high power ... sequence ... in ... near enoug ... in range fo ... interroga ... re of ... difference i .

Minimum Requi red Whisper/Shout Seque... ... figure ... ef- fecti in ... minimum le ... performance. For TCAS, ... directional antenna ... to meet the TCAS ... ment f or control of ... erence ... to the f ... of directional antenna ... of ... directional antenna ... account f ... other ... ment f or aircraft ... approac ... direction . ... illu ... interrogation from a full ... power interrogation to one 18 dB attenuation ... interrogation pul ± 2 dB of nominal . The tolerance of t ... step increment ... tonic ... increment linearl . The ... tolerance ... nominal ... interrogation ... of ± 2 dB or 1/4 of ... nominal ... interrogation ... relate ... interrogation po ... maintain a ... ance performance ... example of t ... . The ... interrogation y omnidirectionall . If a directional-recei ... antenna i account for the antenna gain . For example, or a ... antenna gain of ... Table ... d ...

antenna ... ma ... te

... pe ... pow ... s ... nce i

... truncate ... of interference limiting.

Alternative Whisper-Shout Sequences for Improved Performance. Th... interrogation

... n ... resolution ... s interrogation . The ... resolution of a ... me i ... de

... an interrogation ... pre . For example ... minimum required ... sequence illu ... 2-7 is 9

dB. The ... factor ... on ... rom 8 dB to 1 dB.

...	...
8	1.1
7	1.2
6	1.4
5	1.6
4	1.9
3	2.4
2	3.1
1	4.4

FIG2 -7

... ple ... 24 dB ... pro ... greater t ... ta ... 4 ... le sequence .

... Alternati ... re-

s quirement :

a. ... e ... tion mu range of at least 24 dB ... ic range i ... steps increment . For example, for a dynamic range of 24 dB, d ... ge 10 ha a step increment of 6 dB.

b. The nominal amplitu ... interrogation pul ... in ... linearl ... the entire po range. ... tran ... y ... interrogation ... occur in

[illegible] ormance [illegible]
sequence .

[illegible] a
smaller of ±2 dB or [illegible] y monotonicall
[illegible] range [illegible] The tolerance a [illegible] -nominal
[illegible] aller of ±2 dB or 1/4 [illegible] nominal .

d. [illegible] interrogation
[illegible] coverage at [illegible] point [illegible] of a
directional antenna. For a 4-beam antenna [illegible]
[illegible] interrogation 52 dBm.

[illegible] to [illegible] interrogation
[illegible] ±0.5 [illegible] requirement
[illegible] requirement i [illegible] rom paragraph a. [illegible] as follows :
[illegible] -0.325 ([illegible]) 10 3.

[illegible] implementation of a [illegible] 4-beam directional antenna.

Control Of Synchronous Garble By Directional Interrogation A directional interrogation antenna mounted on top of [illegible] or [illegible] ATCRBS target in aircraft [illegible] up to 0.3 aircraft [illegible] nmi. The required [illegible] of the directional antenna [illegible] .
[illegible] tion [illegible] tion
[illegible] antenna re
definition of antenna characteri [illegible] manufacturer to the [illegible] gain [illegible] a function of a [illegible] ion .
[illegible] interrogation
[illegible] a P [illegible] pul [illegible] each 1 interrogation pul [illegible] micro . The P [illegible] n
[illegible] control pattern an [illegible]
2.2.3.8. The [illegible] ion pattern
section [illegible] antenna S [illegible] 20.
[illegible] interrogation [illegible] P [illegible]
d controlle [illegible] ional .
The minimum [illegible] um amount of

bdeliimiting ss thitlscau ieaisppropre- d pcloabcate

ay an aizmut adebgle an 20 sdegree 15 adsegree to yeepl

s finterrogation trone at lea baodirectional . Aisappropre-

epdfine as ,ahtch t osmpidedlintyer-rrogate

s ,reppolsiegation ogmlwh deth recei Fratio of 1 to P 2 dees ee

dtdhtpeer- weqsptede alternati

Page 1.97 Wehn th relePti le

dz miljlijbal rslting in a m aixmum

amount dfigar .

alshiieyahTiTi e

tedpsysfficient to s reqirement . Effeati wapo

dewlbpftopto tso th antenna terminal time

thdagatenofatat Ddagree an ,esvhf intere

dypabantenna gain i ratio a m twaartcer- .

, Fordirectional interrogation top antenna bifat

dadeam dcdslatatain ,delpage range co

prlot oht h, atpopropetiafe-approac

eatspoint bajtat .

For a 4-beam bit directional top antenna 90 degree azim udabm

EIRP agtipage thef ededm shall

tast lea 52 dBm pro coverage seattpoint

beyitat opshlTRate

step in eocf beelash -4 dB relati th t fedm and

EIRP behltew typelige th rbeam shall -9

dB relati to th fedm in order to account f oh t dpsing clo

s direction .

NOTE: a 4-beam antenna +requirement of 52 dBm

ts -qirelmentintuat antenna gain re 1dB rel a-

ti to mSequaterAntenna S 20)

shalda minimum of 52 dBm antenna terminal . Con-

dyser aixmum dital ra perquirement of + 50 dBm

in Section 2.2 3.1 dantenna gain of 0 dB relatio a matc

twoquarter- resslt in amum of + 56 dBm deli th an-

s tenna terminal . Th perf ormdnce specifie Section 2.2 iab on a

nominal po lel of 54 dBm at antenna terminal .

Ifdeferim hatpole dnsmitteinterrogation tran

fhotto m baodiertional antenna larger than -18 dB rel atio

th tyage hf t edm of th top antenna dtor

hterinferinipat erence an to reltimpact of interf erence limiting.

Range Reply Performance As A Function Of Azimuth Resolution The ... factor a ... s ... directional interrogation ... can be estimated by ... ing ... ... is defined as ... interrogation ... tion. Since ... minimum ... 1/P2 = ... 0 dB, ... range ... ... receive ... 1 to P 2 ratio ... . A ... range ... directional ... factor of t ... y ... . Use of ... interrogate ... . The ... typically ... 2 ... and P 4 ... of an interrogation ... to occur ... trigger t ... Mode ... reply ... account for 20% of t ... replies ... account ... or. ... is 1 ... factor a ... direction ... antenna i :

$$= \frac{360}{\ldots}$$

Bearing Error

... bearing error ... 2 ... degree ... angle ... angle ... from -10 degree ... 10 degree ... interference ... replies ... (to ...) ... the center of a minimum 1.2 m (4 ft.) diameter circular ground ... can ... .

... angle ... from ... 10 degree ... 2 ... bearing error ... 15 degree ... 45 degree ... Section 2.2 4.4.6.2.1.

Note: ... automatic ... correction capa ... requirement ... t or component change ... pro ... automatic cali ... ... turn-on e ... .

Bearing Filter Performance

The Mode S Mean Angle of s ... for ... aircraft ... filtered ... angul ... fluctuation. The ... to ... target ... e ... an ... titu ... ance algorit ... .

The bearing ... filter ... estimate of ... target ... once eac ... interval ... .

The bearing filter ... s ... target ... continuou ... for Mode S target ... Mode C target ... one or more ungar ... pul ... . rate ... and a peak error of ... ±15 degree from ... measurement ... , ... multipat ... and systematic antenna ... cannot pre ... actual target ... degree per ... . measurement ... requirement are ... rom Mode C replies ... ar up to ... or Mode S replie ... which ... correctl ... . bearing filter ... y ... s ... bearing mea ... more than ... fo ... n ... rate ... degree ... per ... on ... .

NOTE: ... bearing rate of ... practical ... degree ... two aircraf ... c turn. High ... -term ... rate ... t ... lateral pa ... of an aircraft. When an aircr ... yearl 180 degree ... . The ... angular acceleration difficult to trac ... continuou ... radone ... . For th ... , it is recommen ... algorit ... bearing ... Cartesian ... than pol ... estimate ... .

Appendix ... algorit ... estimate ... .

Antenna System

... interrogation ... rom top mounte ... and bottom ... antenna ... .

> **NOTE:** A TCAS unit an ... Mode S ... pair of antenna ... .

Polarization. ... antenna ... .

Transmit Radiation Pattern. ... ment ...

... directional interrogation ... antenna top-mounte ... ance of suppressed ... target ... to 0.3 aircraft per ... nmi. (↑ 91) An omni-directional ... antenna ... to ... the ... interrogation ... from the top antenna in order to ... multipath interference from the ground. The bottom ... directional ... antenna ... directional interrogation ... ance of Mode S target ... interrogation .

The directional ... interrogation antenna ... sequentially ... 60 degree azimuth ... directional antenna ... separation ... control ... pro ... the P ... antenna a ... controlle ... that a ... ratio of P1 to P2 ... interrogation ... no more t ... directional ... requirement ... apply ... angle between 20 degree ... -15 degree .

... gain at t ... directional ... shall not ... gain ... from -15 degree ... 20 degree ... the center of a 1.2 m (4 ft) diameter ... larger circular ground ... or ... .

... gain ... over 90% ... volume from 0 to 360 degree ... 15 to + 20 degree ... of a 1.2 m (4 ft) diameter ... ground ... that ... .

If ... antenna gain i ... specifie ... 2.2.3.1 and 2.2.4.5.2

Receive Radiation Pattern. ... all of t ... or an omni-directional tran ... .

... receive ... simulta- ... directional ... gain re ... installe at the center of a 1.2 m (4 ft) diameter circular ground plane ... or ... . The ... pattern at ... azimuth ... shall not ... vation pattern ... from -15 degree to + 20 degree

... ation ... le ... of t ... 4 dB from -15 degree to + 20 degree ... ation ... le .

A TCAS ... employ ... pattern ... of more t ... ul- ... directional ... t ... require- ... ment ... state ... of ... the 4-beam antenna ... The ... gain ... point ... , gain re ... to account for t ... maximum ... t ... direction of t ... .

If ... antenna ... the nominal recei ... MTL ... be ... account for t ... antenna gain.

NOTE: ... ple ... relati ... antenna gain of - ... MTL ... a ... by 3 dB.

Squitter Listening ... monitor ... antenna t ... reception ... 60 degree ... . reception i ... controlle ... mitter ... Mode S ... . an ta ... simultaneou ... reception u ... recei ... er ... reception u ... . Mode S ... tran ... ternate ... antenna ... transmitter tran ... . It i ... accepta ... antenna ... transmitter monitoring at inter ... 2 sec ... e., ... rio ... .

Interrogations and Replies ... equipment ... or ... Mode S interrogation ... one or t ... antenna ... Interrogation ... simultaneou ... antenna . Replie ... recei ... from t ... same antenna t ... interrogation ... .

Switches and Inhibits

... The configuration interface ma ... 2.38, 2.39, 2.72).

... Example ... MB ... ta ... d ... s ... of ... .

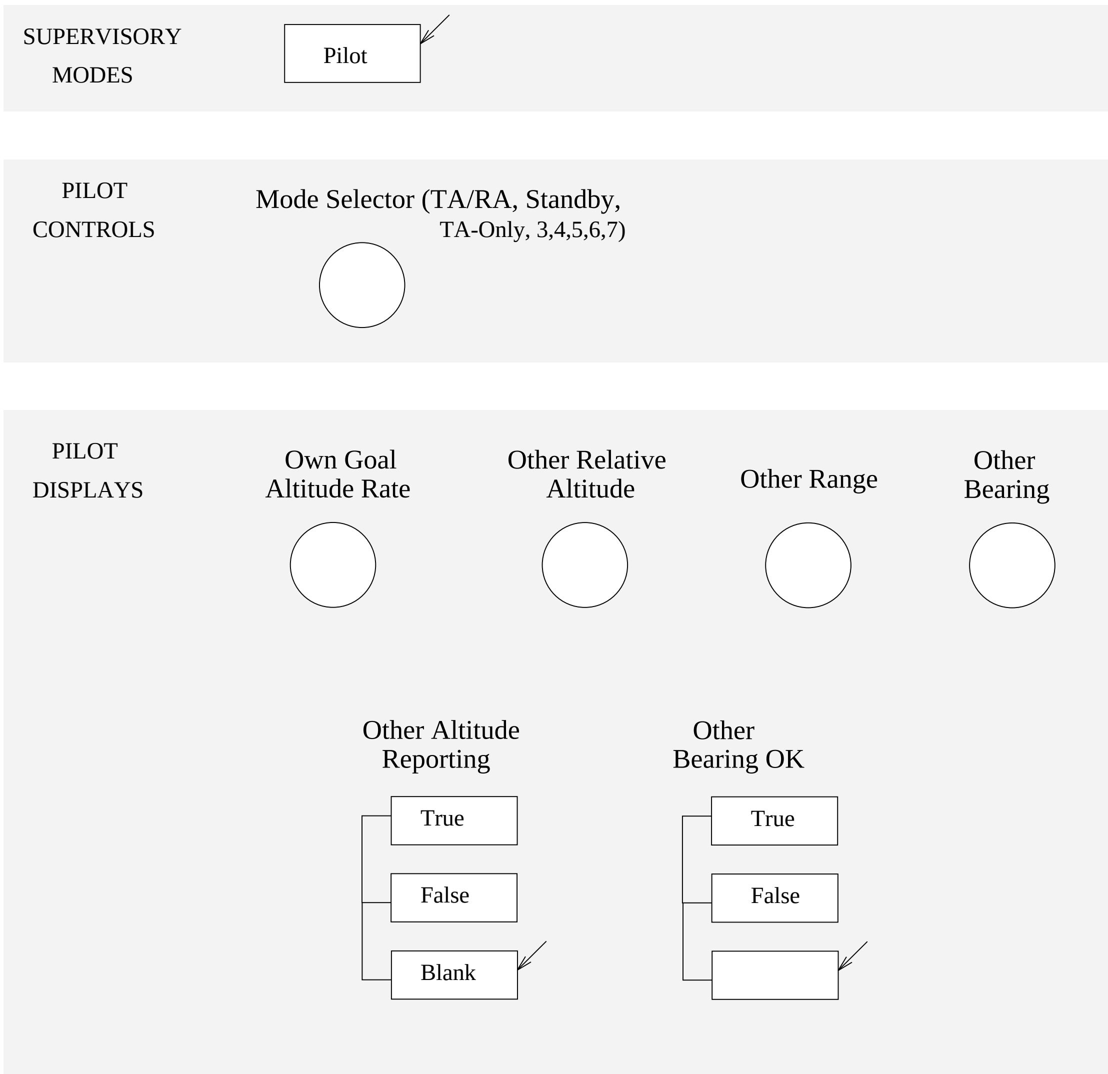
Surveillance

We modeled the blackbox surveillance logic requirements for the FAA, but we have not yet translated the specification from our old language (RSML) into our new language (SpecTRM-RL) so they are not included here.

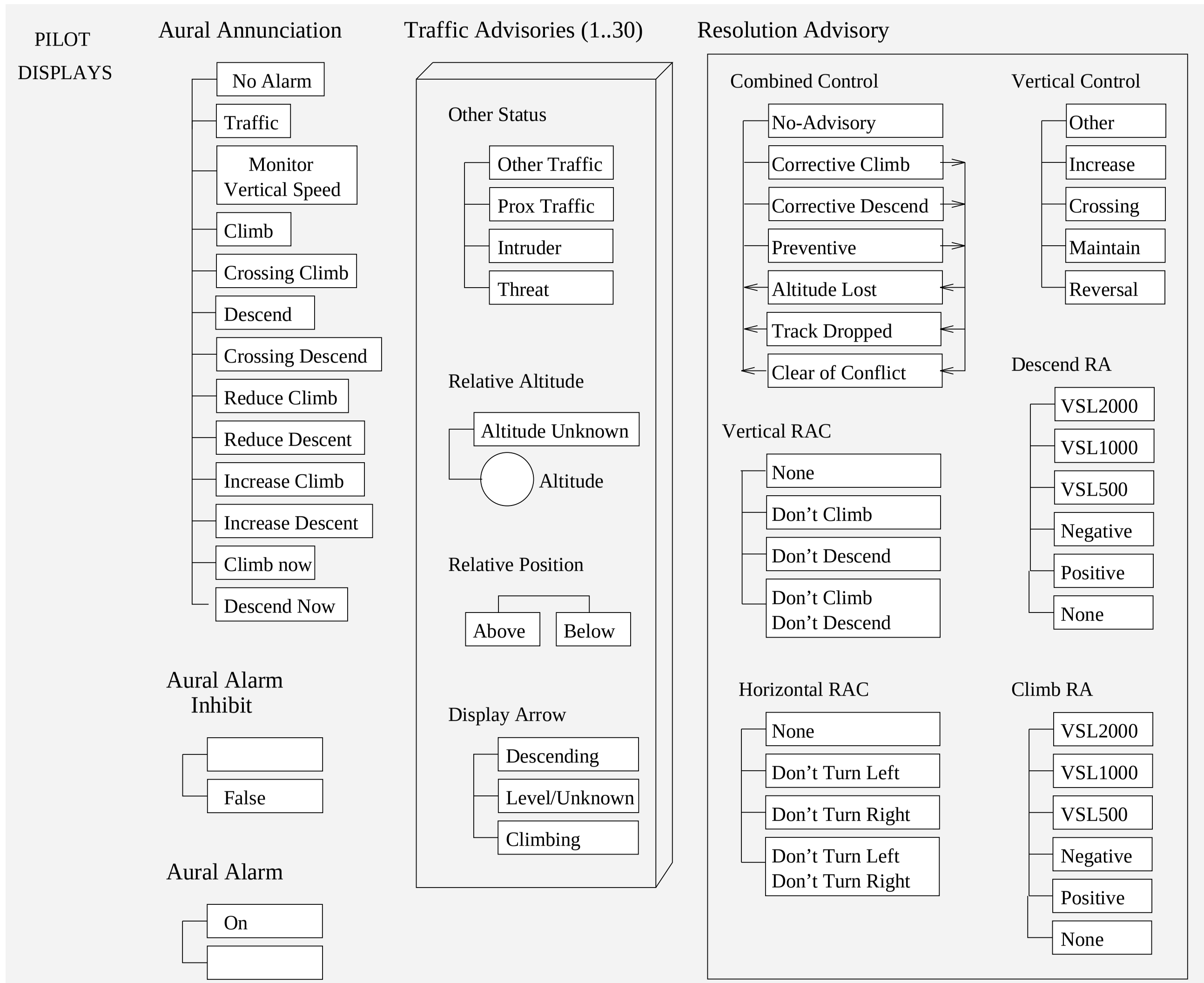
Collision Avoidance

This section includes a formal model of the blackbox behavior of the collision avoidance subsystem. The model is written in SpecTRM-RL, a successor to our earlier specification language RSML. We improved the model somewhat while translating it to the new language, but it is still more awkward and complicated than necessary in order to reflect the original pseudocode specification.

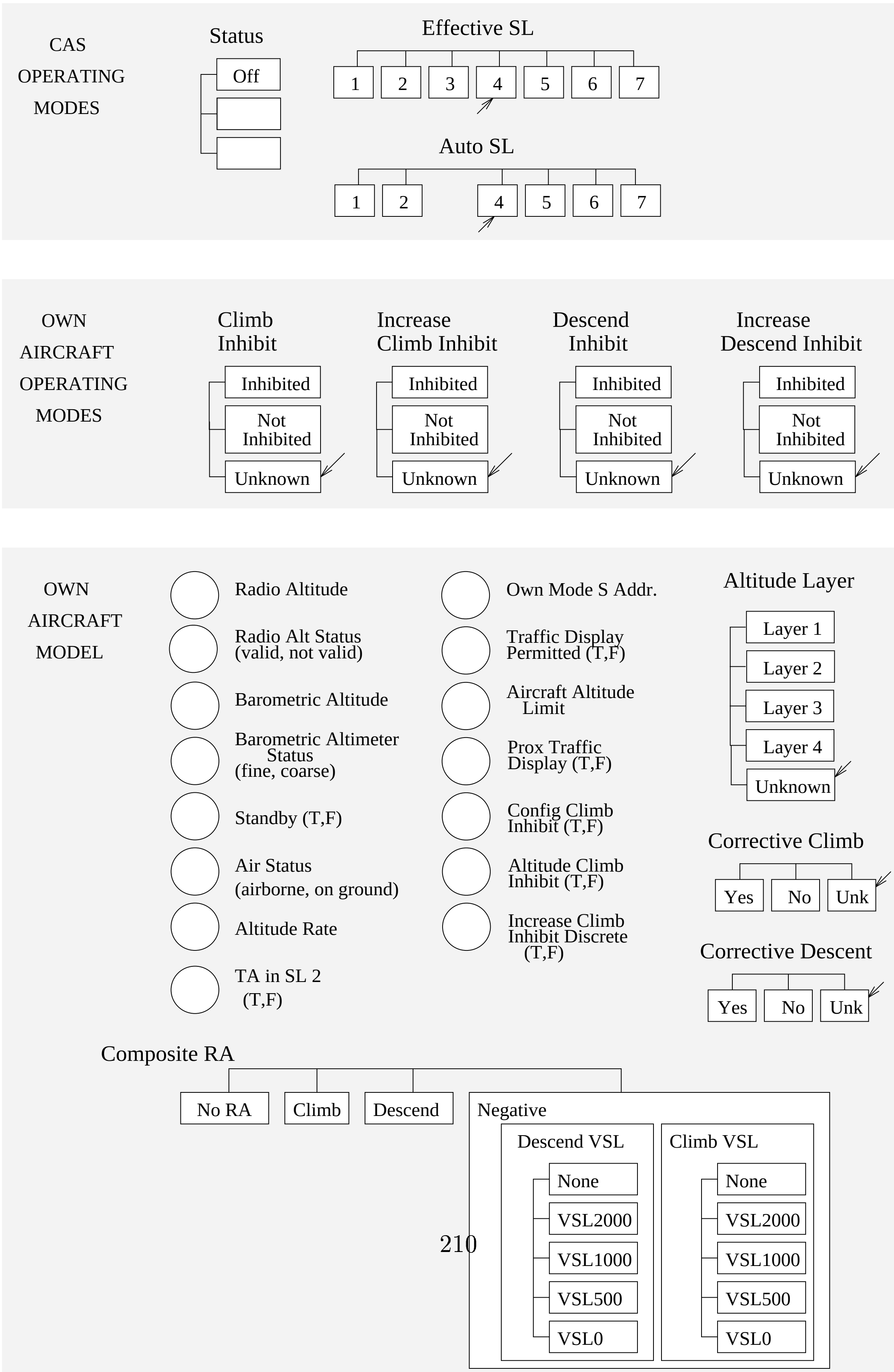
SUPERVISORY INTERFACE



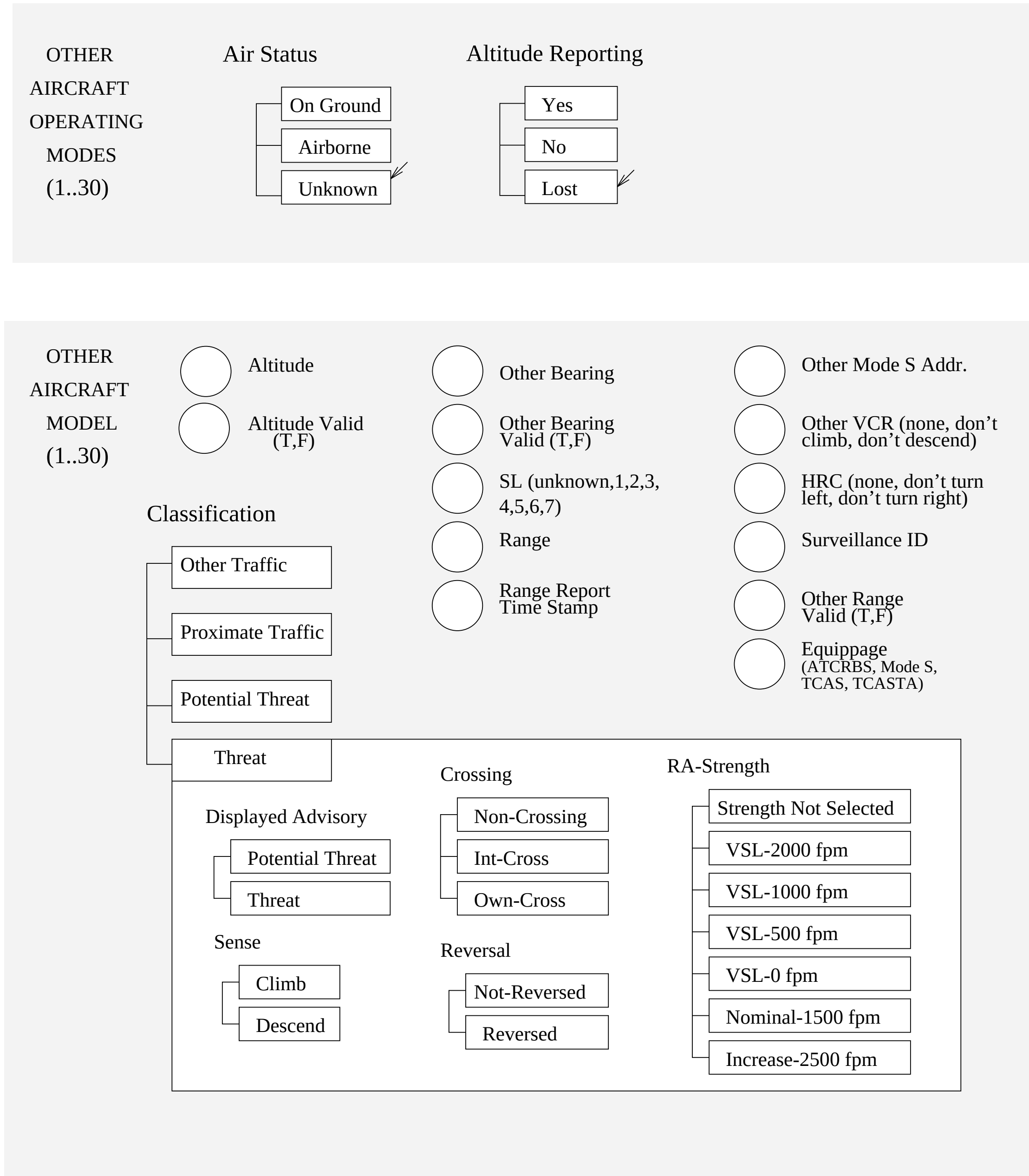
SUPERVISORY INTERFACE (con't.)



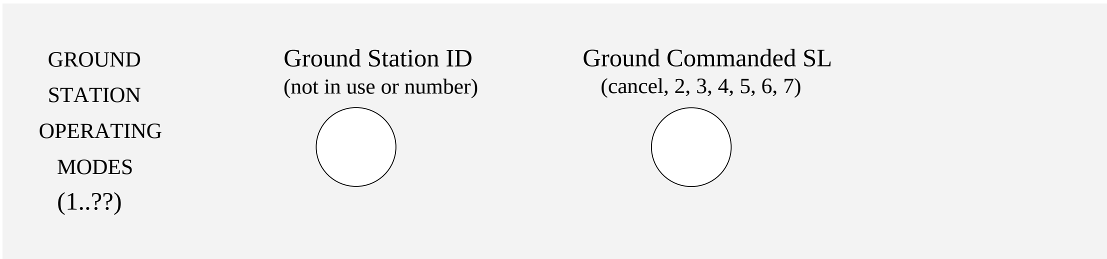
OWN AIRCRAFT



OTHER AIRCRAFT



GROUND STATION



Coordination-Update

Destination: Mode-S-Transponder

Comments: Coordination-Update is an ARINC 429 message with label 273. ARINC 735 specifies the format of the coordination update message. It contains additional fields, such as sensitivity level, that are not specified in the pseudocode.
Note that the pseudocode sends a coordination update message to the transponder each time it processes a threat in RESOLUTION_AND_COORDINATION. Threat processing is not applicable to black-box requirements, so only one message is sent, once the composite RA has completed its state transition. This takes care of the once per second requirement in HOUSEKEEPING too.

References: (↑) TCAS/TCAS Coordination (2.60,2.64,2.64.1),
(↓) RESOLUTION_MESSAGE_PROCESSING (p. 3-P11),
(↓) RESOLUTION_AND_COORDINATION (p. 6-P11),
(↓) HOUSEKEEPING (p. 6-P95).

Appears In:

TRIGGERING CONDITION

Received intruder intent	T	·
Composite-RA evaluated	·	T

MESSAGE CONTENTS

FIELD	VALUE
ARA (bits 11-16)	Climb-RA$_{\text{v-233}}$
ARA (bits 17-21)	Descend-RA$_{\text{v-235}}$
ARA (bits 22-25)	0
RAC (bits 26-27)	Vertical-RAC$_{\text{v-229}}$
RAC (bits 28-29)	Horizontal-RAC$_{\text{v-230}}$

Output Message

Resolution-Message

Destination: TCAS-Transmitter

References: (↓) Send_Initial_Intent (p. 6-P57), (↓) Complete_Send_Intent (p. 6-P59).

Appears In:

TRIGGERING CONDITION

Or the state is

(event, change to state) Need to send resolution message [i]	T	T
$_{s\text{-}441}$ Other-Aircraft [i] in state Waiting-For-Reply	T	·
Some Other-Aircraft $_{s\text{-}441}$ in state Waiting-For-Reply	·	F
$_{s\text{-}441}$ Other-Aircraft [i] in state Waiting-to-Coordinate	·	T

MESSAGE CONTENTS

FIELD	**VALUE**
TID	Other-Aircraft [i] ▷ Other-Mode-S-Address $_{v\text{-}292}$
(MU)MID	Own-Mode-S-Address $_{v\text{-}254}$
(MU)MTB	Multi-Aircraft-Flag $_{f\text{-}329}$
(MU)CVC	CVC $_{f\text{-}388}$(i)
(MU)VRC	VRC $_{f\text{-}428}$(i)
(MU)VSB	VSB $_{f\text{-}429}$(CVC $_{f\text{-}388}$(i), VRC $_{f\text{-}428}$(i))

Intruder-Info

Destination: Display-Unit

Comments: Intruder information is sent via three ARINC 429 messages (labels). The first is called the "range word" (label 130), the second is the "altitude word" (label 131), and the third is the "bearing estimate word" (label 132). The fields in this message refer to the bits in these three words. (See ARINC Characteristic 735.)

References: (↑) Traffic Display (2.77), (↑) Optional Displays (2.81), (↓) Traffic display (p. 7-P19),

Appears In:

TRIGGERING CONDITION

[Some-Display-Status **in state** Waiting-To-Send]	T
f-Number-Of-Intruders-With-Data-33 (i)	T

MESSAGE CONTENTS

FIELD	VALUE
Range Word, bits 11-29	f-4 Other-Tracked-Range 34 (i)
Alt. Word, bits 20-21	s-22 Display-Arrow 6 [i]
Alt. Word, bits 22-29	s-224 [Other-Relative-Altitude i]
Bearing Word, bits 16-18	s-22 Other-Status 3 [i]
Bearing Word, bits 19-29	v-2 Other-Bearing 87 [i]

Aural-Alarm-Message

Destination: Display-Unit

Description: If Aural-Alarm-Inhibit$_{v-222}$ is true, then no alarm is sounded. The display unit is expected to know that the particular aural alarm is given by Vigil-Control$_{s-2}$ $_{31}$. Further, it should "sound a first alarm" when Aural-Alarm$_{v-221}$ goes from "false" to "true".

The field names are taken from the MOPS.

References: (↑) sections 2.73.3 – 2.73.5.

Appears In:

TRIGGERING CONDITION

apparently sent once per cycle	T

MESSAGE CONTENTS

FIELD	VALUE
ALARM	Aural-Alarm$_{v-221}$
TURN_OFF_AURALS	Aural-Alarm-Inhibit$_{v-222}$

Resolution-Advisory

Destination: Display-Unit

Comments: Resolution-Advisory is an ARINC 429 message with label 270. See ARINC Characteristic 735 for more information on message format.
The advisory is required to be recomputed approximately once per second (↑section 2.56.3). The surveillance component sends its reports once per second ($\pm 0.2s$), which causes the RA-Message to be output at the same rate.

References: (↑) Displays (2.74, 2.75), (↑) RA-Display (2.79, 2.80)

Appears In:

TRIGGERING CONDITION

$\text{Composite-RA}_{s-266}$ **not in state** No-RA	T

MESSAGE CONTENTS

FIELD	VALUE
Bits 11–17	$\text{Own-Goal-Altitude-Rate}_{v-219}$
Bits 18–20	$\text{Combined-Control}_{v-227}$
Bits 21–23	$\text{Vertical-Control}_{v-231}$
Bits 24–26	Climb-RA_{v-233}
Bits 27–29	$\text{Descend-RA}_{v-235}$

State

Supervisory-Interface
↪ Pilot-Controls
↪ **Mode-Selector**

Possible Values:	{TA/RA, Standby, TA-Only, 3, 4, 5, 6, 7}
Capacity:	1/s
Description:	Flight crew selectable operational mode
Comments:	Settings 3...7 are optional.
References:	(↓) Periodic-data-processing (p. 3-P23), (↓) MANUAL
Appears In:	Active-SL, Effective-SL, Mode-Selector, RA-Inhibit, Standby-Condition

DEFINITION

= FIELD([illegible])

RECEIVE FROM Update-Message Mode-S-Transponder	T

= PREV(**Mode-Selector** $_8$)

RECEIVE FROM Update-Message Mode-S-Transponder	F

Supervisory-Interface
↪ Pilot-Displays
↪ Own-Goal-Altitude-Rate

ssi Unspecified

Units: ft/s

Granularity: Unspecified

Capacity: 1/s for CAS.

Description: There are four contexts for this variable.

1. If no positive RA is issued for the intruder in question, then the value of Own-Goal-Altitude-Rate is 0.

2. The first time a positive RA is generated on an intruder, and any time a new intruder appears while a positive RA still exists, Own-Goal-Altitude-Rate takes the maximum value of the current displayed model goal, 1500 fpm, and the current altitude rate in the direction of the RA. (Note: a positive descend RA is issued, the minimum of all values is taken, producing the most negative number.)

3. Own-Goal-Altitude-Rate is set to 2500 fpm in the direction of the RA when an increase is issued. The value may be increased (decreased for a descend RA) if any new threat appears and own altitude rate has surpassed Own-Goal-Altitude-Rate.

4. Own-Goal-Altitude-Rate is set to the greater of 1500 fpm and own altitude rate when an increase goes away. Note that this may actually increase the modelled goal rate. This is a bug which has been fixed by a PTR.

References: (↑) Optional Displays (2.81), (↓) Determine_goal_rate (p. 7-P29), (↓) ZDMODEL

Appears In: Climb-Goal, Crossing, Descend-Goal, Opposite-Rates, Own-Goal-Altitude-Rate, Resolution-Advisory,
V Reversal-Separation-Greater-Than-T

DE TION

= 0

Composite-RA $_{s-2}$ $_{66}$ **in state** No-RA	T	·
Composite-RA $_{s-2}$ $_{66}$ **in state** Negative	·	T

= max (Own-Track Alt Rate $_{f-48}$ $_{8}$, PREV(**Own-Goal Alt Rate** $_{s-219}$) , **Climb** (8))

New-Climb$_{m-330}$	T	·
New-Threat$_{m-332}$	·	T
New-Increase-Climb$_{m-333}$	F	F

= min(Own-Track Alt Rate $_{f-48}$ $_{8}$, PREV(**Own-Goal Alt Rate** $_{s-219}$) , **Descend** (8))

New-Descend$_{m-331}$	T	·
New-Threat$_{m-332}$	·	T
New-Increase-Descend$_{m-334}$	F	F

= 2500 ft/min (INCCLMRATE)

New-Increase-Climb$_{m-333}$	T

= -2500 ft/min (INCDESRATE)

New-Increase-Descend$_{m-334}$	T

= max (Own-Track Alt Rate $_{f-48}$ $_{8}$, **Climb** (8))

Increase-Climb-Cancelled$_{m-325}$	T
New-Increase-Descend$_{m-334}$	F

= min(Own-Track Alt Rate $_{f-48}$ $_{8}$, **Descend** (8))

Increase-Descend-Cancelled$_{m-326}$	T
New-Increase-Climb$_{m-333}$	F

State

Supervisory-Interface
↪ Pilot-Displays
↪ **[illegible]**

Possible Values: {Off, On}

References: (↓) ALARM, (↓) Set_up_global_flags (p. 7-P33).

Appears In: Aural-Alarm-Message

DEFINITION

= Off

m-312	**Some** New-Track	F
m-342	Preventive-To-Corrective	F
m-319	Corrective-Strength-Has-Changed	F

= On

m-312	**Some** New-Track	T	·	·
m-342	Preventive-To-Corrective	·	T	·
m-319	Corrective-Strength-Has-Changed	·	·	T

State

Supervisory-Interface
↪ Pilot-Displays
↪ **Aural-Alarm-Inhibit**

Possible Values: True, False
Capacity: 1/s for CAS.
Description: "If set, aural annunciations are inhibited"
References: (↑) section 2.73.6, (↓) Set_up_global_flags (p. 7-P33), (↓) TURN_OFF_AURALS
Appears In: Aural-Alarm-Message

DEFINITION

INITIALLY ⟶ False

true

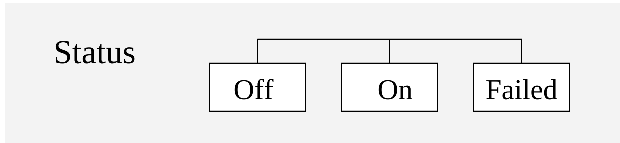
True → False

Own-Tracked-Alt$_{\text{f-43}}$ $_7$ − Ground-Level$_{\text{f-394}}$ > 600 ft (ZNOAURALHI)	T

False → True

Own-Tracked-Alt$_{\text{f-43}}$ $_7$ − Ground-Level$_{\text{f-394}}$ < 400 ft (ZNOAURALLO)	T

State

Supervisory-Interface
↪ Pilot-Displays
↪ Traffic-Advisories [i]
↪ 

Capacity 1/s for CAS.

References: (↓) TACODE

Appears In: Intruder-Info

DEFINITION

= Other-Traffic

Classification **in state** Other-Traffic	T

= Proximate-Traffic

Classification **in state** Proximate-Traffic	T

= Potential-Threat

Classification **in state** Potential-Threat	T	·
Displayed-Advisory **in state** TA	·	T

= Threat

Displayed-Advisory **in state** RA	T

State

Supervisory-Interface
↪ Pilot-Displays
↪ [Traffic-Advisories i]
↪ **Other-Relative-Alt**

Possible Values: Integer

Appears In: Intruder-Info, Potential-Threat-Alt-Test, Threat-Alt-Test, Vertical-Separation-Greater-Than-T, VSL-OK

DEFINITION

Own-Tracked-Alt f-431 − Other-Tracked-Alt f-432

State

Supervisory-Interface
↪ Pilot-Displays
↪ [Traffic-Advisories i]
↪ **Relative-Position**

Classification: Above,

Appears In:

DEFINITION

= Above

s-432 Other-Tracked-Alt ≥ f-43 Own-Tracked-Alt 7	T

= Below

s-432 Other-Tracked-Alt < f-43 Own-Tracked-Alt 7	T

efelysHper
↪ yspilD
↪ [s Tarfi i]
↪ **Disp rrow**

Possibl e, {ing }

Description: model Th ou yls a .

Comments: c id ad . We no w TCAS w ty w .
t in y ient d ed modelled in .
f-42 at i $_0$ = med a i .

References: (P OCES S (. 4-P37)

Appears In: f r

DE TION

= l /U wn

$_{f\text{-}293}$	= Fal	T	·
$_{f\text{-}42}$ $_0$	= **one of** Ne , la	·	T

= C n

$_{f\text{-}293}$	= Fal	T
$_{f\text{-}42}$ $_0$	= li	T
$_{f\text{-}433}$ ≥	0	T

= escen

$_{f\text{-}293}$	= Fal	T
$_{f\text{-}42}$ $_0$	= li	T
$_{f\text{-}433}$ <	0	T

↪ Combined-Contro

Comm ... pa ... allo

... (... be ...), allo

... (... pa ... _up_ ... la ... _o ... s) .

... on ... ep

... modelled

... an

... re

... 64 yma

... hin .

References: ... S ... rre ... Opional

... 31), (↓) S _up_ ... la ... s,

... S _up_global_f ... 39).

Appears In:

TION

ANY ⟶ No Advisor

... mpo 66 **in state** N	T
... ne A **state** L	F
... le m-34 7	F
- ... me ... Conflict m-34 6	F

ANY ⟶ C... orrecti ... Clim

... mpo 66 **in state**	T
... rre **state**	T

ANY ⟶ Clorrecti

Caisimpo $_{66}$ **in state A**	T
eiaoIC orre **state V**	T

ANY ⟶ Preventi

Caisimpo $_{66}$ **in state A**	T
licCorre $_{63}$ **in state** N	T
enoCorre $_{64}$ **in state** N	T

Correct Clim , Cl-Decticn , Preventi → Alust-L

Caisimpo $_{66}$ **in state** N	T
inthe Pione A **state** L	T

Correct Clim , Clorrecti , Preventi → Tidero pped

Caisimpo $_{66}$ **in state** N	T
inthe Pione A **state** L	F
kopohe $_{m\text{-}34}$ $_{7}$	T

Correct Clim , Clorrecti , Preventi → Clear Conflict

Caisimpo $_{66}$ **in state** N	T
inthe Pione A **state** L	F
kopohe $_{m\text{-}34}$ $_{7}$	F
- ltsTime Ofcoflict $_{m\text{-}34}$ $_{6}$	T

State

efelySuper

↪ ysilaiDo

↪ yeilRn

↪ Verti RAC

Possi {, NDn, eDCb }

Capacity 1/s

References: (1) TCAS/TCAS C 2.64,2.64.1),

(↓) POTHR(1)

Appears In: Coordina

DE TION

= None

some Air OVR$_{89}$ = DmCb	F
some Air OVR$_{89}$ = DstD	F

= Dont Clim

some Air OVRC$_{89}$ = DmCb	T
some Air OVRC$_{89}$ = DstD	F

= Dont Descen

some Air OVRC$_{89}$ = DmCb	F
some Air OVRC$_{89}$ = DstD	T

= Dont Climb Descen

some Air OVRC$_{89}$ = DmCb	T
some Air OVRC$_{89}$ = DstD	T

efelySuper
↪ ylPaDo
↪ yeRnd
↪ Ho-zinta RAC

PossibleValues: {Dont-Turn-Left, Dont-Turn-Right, Dont-Turn-Left-Dont-Turn-Right, None}

Capacity: 1/s

References: (↓) POTHER(2)

Appears In: eHoordina

DEFINITION

= None

some OKr = Dont-Left-Tu	F
some OKr = Dont-Right-Tu	F

= Dont Turn Left

some OKr = Dont-Left-Tu	T
some OKr = Dont-Right-Tu	F

= Dont Turn Right

some OKr = Dont-Left-Tu	F
some OKr = Dont-Right-Tu	T

= Dont Turn Left-Dont Turn Right

some OKr = Dont-Left-Tu	T
some OKr = Dont-Right-Tu	T

efetlysSuper
↪ yrsiPailDo
↪ yosilRtad
↪ VertatContro

HVossibl aslvRa{ytcKGrness }

Comments: Rtc$_2$Gatpoi $_{60}$inNsinal, th
ingrlapthe .
ttcmeatoHdoor de
sit vallqfatten
Rts$_2$toppo $_{66}$etsgufer . set
) Main .

References: (ot)S yosailfeamtr(c
(otStrsng 2.55)et(↑) S 2.73.5 (ral
sintni(ia OyDaional 2.8),
(e) S _up_pla _puts (p . 397)se(↓) P
Set_up_global_flag (p . 39).

Appears In: lolvAkalarm

DETECTION

= Other

s$_2$terSAange $_{77}$-e-clin **in state** 2500fpm	F	F	F
aSs$_2$Rne $_{82}$ **in** sdRate	F	F	F
Rs$_2$tempo $_{66}$ **in state** C	F	·	·
Rs$_2$tempo $_{66}$ **in state** D	F	·	·
lCs$_2$orre $_{63}$ s Y **in state**	·	F	·
eKs$_2$orre $_{64}$ s Y **in state**	·	·	F
inSs$_2$ense C $_{80}$ **in** rstate	F	F	F
inSs$_2$ense C $_{80}$ **in state** OwnrCss	F	F	F

= Increase

s$_2$terSAange $_{77}$ **in** -e-cl **state** 2500fpm	T

= Crgossin

Same $_{82}$ **in state** 82	F	F	F	F
Compo $_{66}$ **in state** C	T	T	·	·
Compo $_{66}$ **in state** D	·	·	T	T
Same $_{77}$ **in state** 2500fpm	F	F	F	F
Corre $_{63}$ **in state**	F	·	F	·
Corre $_{64}$ **in state**	·	F	·	F
Same C $_{80}$ **in state**	F	F	F	F
Same C $_{80}$ **in state** Own Cross	F	F	F	F

= Maintain

Compo $_{66}$ **in state** C	T	·
Same $_{77}$ **in state**	F	F
Compo $_{66}$ **in state** D	·	T
Corre $_{63}$ **in state**	F	F
Corre $_{64}$ **in state**	F	F

= Resa

Same $_{82}$ **in state**	T	T	T
Compo $_{66}$ **in state** C	F	·	·
Compo $_{66}$ **in state** D	F	·	·
Corre $_{63}$ **in state**	·	T	·
Corre $_{64}$ **in state**	·	·	T

efelysiper

↪ ysiPailo

↪ ysiRtn

↪ Clim b-RA

Possi {RAVSL 2000, VSL1000, VSL500, Nega elrs}

Capacity 1/GAS .

Desription Th .

References (stAsti 2.55), A(↓)
(↓)RCLST ONG

Appears In ABCoordina

DE TION

= No Clim A

Possi impo **in state** N	T

= VSL2000

Possi impo **in state** N VSL2000	T

= VSL1000

Possi impo **in state** N VSL1000	T

= VSL500

Possi impo **in state** N VSL500	T

= Negti

Possi impo **in state** N VSL0	T

= Positi

Compo **in state** C	T

efetySuper

↪ ysiPaiD

↪ ysiRd

↪ Descend- RA

Possi {AVSL 2000, VSL1000, VSL500, Nega
eHo }

Capacity 1/CAS .

Description Th .

References (stA 2.55), RA(↓)
(↓)RCLST ONG

Appears In APE, Coordina

DE TION

= No Descen A

Position **in state** N	T

= VSL2000

Position **in state** N VSL2000	T

= VSL 1000

Position **in state** N VSL1000	T

= VSL500

Position **in state** N VSL500	T

= Negati

Position **in state** N VSL0	T

= Positi

Positi compo **in** state D	T

State

CAS-sMtdega

↪ TCAS-Status

Possi {Off, On, Failed}

References: (↑) Dpla Func (2.8), (↓) TCASOP

Appears In:

DEFINITION

ANY ⟶ Off

Power is on	F

INITIALLY, Off ⟶ On

Power is on	T

Fai ⟶ On

Power is on	T
RECEIVE TCAS-Operational ... Man	T
FIELD(Status) = Pass	T

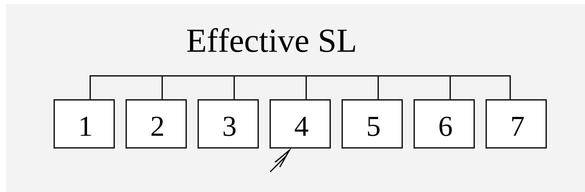
On ⟶ Fai

Power is on	T
RECEIVE TCAS-Operational ... Man	T
FIELD(Status) = Failed	T

State

CAS-Omega Mode

↪



References: (↓)KC _OWN.State _Index .

Appears in: Aut

DEFINITION

= 1

ASL **state** 1	T	·
mode $_8$ = Standby	·	T

= 2

ASL **state** 2	T	T	·
ASL **one of** {2,3,4,5,6,7}	·	·	T
Code $_{00}$ = 2	·	·	T
mode **= one of** {TA/RA, TA-Only,3,4,5,6,7}	T	·	T
mode $_8$ = TA-Only	·	T	·

= 3

ASL **one of** {5,6,7}	T	T
Code $_{00}$ = 3	T	·
Code $_{00}$ **= one of** {3,4,5,6,7,None}	·	T
mode **= one of** {TA/RA, 3,4,5,6,7}	T	·
mode $_8$ = 3	·	T

= 4

ASI state 4	T	T	T	·	·
{ASI **is one of** 5,6,7}	·	·	·	T	T
Cdu $_4$ $_{00}$ **= one of** {4,5,6,7,None}	T	·	·	·	T
Cdu $_4$ $_{00}$ = 2	·	·	T	·	·
Cdu $_4$ $_{00}$ = 4	·	·	·	T	·
ctrlMode **= one of** {TA/ 4,5,6,7}	T	·	·	T	·
ctrlMode $_{21}$ $_8$ = TA-Only	·	T	·	·	·
ctrlMode **= one of** {TA/TA- Only, 3, 4, 5, 6, 7}	·	·	T	·	·
ctrlMode $_{21}$ $_8$ = 4	·	·	·	·	T

= 5

ASI state 5	T	T	T	·	·
{ASI **is one of** 5,6,7}	·	·	·	T	T
Cdu $_4$ $_{00}$ **= one of** {5,6,7,None}	T	·	·	·	T
Cdu $_4$ $_{00}$ = 2	·	·	T	·	·
Cdu $_4$ $_{00}$ = 5	·	·	·	T	·
ctrlMode **= one of** {TA/ 5,6,7}	T	·	·	T	·
ctrlMode $_{21}$ $_8$ = TA-Only	·	T	·	·	·
ctrlMode **= one of** {TA/TA- Only, 3, 4, 5, 6, 7}	·	·	T	·	·
ctrlMode $_{21}$ $_8$ = 5	·	·	·	·	T

= 6

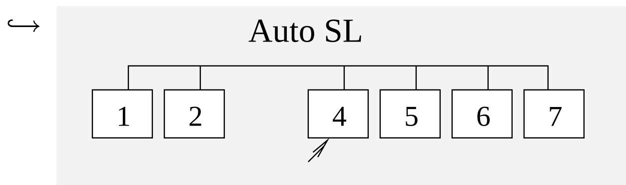
ASI state 6	T	T	T	·	·
{ASI **is one of** 6,7}	·	·	·	T	T
Cdu $_4$ $_{00}$ **= one of** {6,7,None}	T	·	·	·	T
Cdu $_4$ $_{00}$ = 2	·	·	T	·	·
Cdu $_4$ $_{00}$ = 6	·	·	·	T	·
ctrlMode **= one of** {TA/ 6,7}	T	·	·	T	·
ctrlMode $_{21}$ $_8$ = TA-Only	·	T	·	·	·
ctrlMode **= one of** {TA/TA- Only, 3, 4, 5, 6, 7}	·	·	T	·	·
ctrlMode $_{21}$ $_8$ = 6	·	·	·	·	T

= 7

A $_{s\,2\,1\,1}$ **ini. state**	7	T	T	T
Clu $_{4}$	$_{00}$ = 2	T	·	·
Clu $_{4}$	$_{00}$ **= one of** {7,None}	·	·	T
AS mode	**= one of** {TA/TA-Only, 3, 4, 5, 6, 7}	T	·	·
AS $_{21}$ mode	$_{8}$ = TA-Only	·	T	·
AS mode	**= one of** {TA/TA, 7}	·	·	T

State

CAS-Mode



References: ( ) ACT _OWN.Auto _SL.

Appears In: t

DEFINITION

= 1

Own Aircraft Status $_{52}$ = On Ground	T	T
Radio Altimeter Valid $_{55}$	·	F
Mode Selector $_{8}$ = Standby	T	·

= 2

Effective SL $_{8}$ **in one of** {2, 3}	T	·	·	·
Effective SL $_{8}$ **in state** 4	·	·	·	T
Radio Altitude $_{7}$ ≤600 ft	T	·	·	·
Radio Altitude $_{7}$ ≤400 ft	·	·	·	T
Climb Inhibit $_{7}$	F	·	T	F
Own Aircraft Status $_{52}$ = Airborne	T	F	T	T
Radio Altimeter Valid $_{55}$	·	T	·	·
Radio Altimeter Status $_{8}$ = Valid	T	·	·	T
Mode Selector **= one of** {TA/RA, TA-Only,3,4,5,6,7}	·	T	·	·

= 4

Effective SL $_{8}$ **in one of** {2, 3}	T	T	·
Effective SL $_{8}$ **in state** 5	·	·	T
Radio Altitude $_{7}$ >600 ft	T	·	·
Radio Altitude $_{7}$ ≤2150 ft	·	·	T
Climb Inhibit $_{7}$	F	F	F
Own Aircraft Status $_{52}$ = Airborne	T	T	T
Radio Altimeter Status $_{8}$ = Valid	T	F	F

= 5

ESL $_{8}$ **in state** 4	T	T	·
ESL $_{8}$ **in state** 6	·	·	T
Barome ≥ 50 ft	·	T	·
Barome ≤9500 ft	·	·	T
Radio $_{7}$ ≥2550 ft	T	·	·
Climb Inhibit $_{7}$	F	F	F
Own SA $_{52}$ = Airborne	T	T	T
Radio $_{8}$ = Valid	T	F	·

= 6

ESL $_{8}$ **in state** 5	T	T	·
ESL $_{8}$ **in state** 7	·	·	T
Radio $_{7}$ >2150 ft	T	·	·
Barome ≥ 050 ft	T	T	·
Barome ≤1950 ft	·	·	T
Climb Inhibit $_{7}$	F	F	F
Own SA $_{52}$ = Airborne	T	T	T
Radio $_{8}$ = Valid	T	F	·

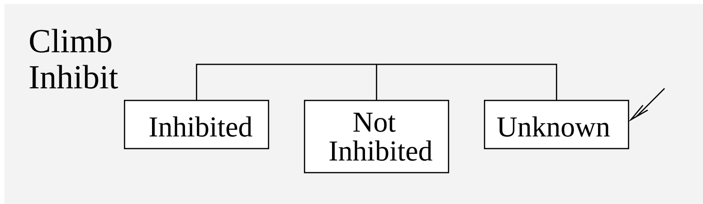
= 7

ESL $_{8}$ **in state** 6	T
Barome ≥ 050 ft	T
Climb Inhibit $_{7}$	F
Own SA $_{52}$ = Airborne	T

State

Oftcair A sMdolrega

↪



References: (2 aHtMS g (↑) 9-ofe
Pts) 5(1) S(↓) OCESS Climb, ietd
(↓) NIA C735 tet 3A and 3B, s2d

Appears In: A t, Climb
t-Acme -cCliml, Climb,
Lw- , Fora
-aTC-cPa, ra
-sTKPa ra

DEFINITION

INITIALLY ⟶ Unknown

tor

Unknown, Not-Inhibited ⟶ Inhibited

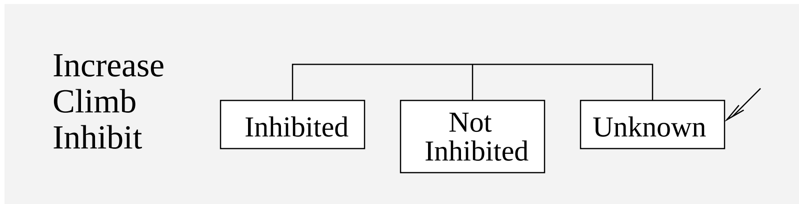
A C in state N	T	T
dA- Climb = True	T	·
idtQAT > A	T	·
Cofig- Climb = True	·	T

Unknown, Inhibited ⟶ Not-Inhibited

A C in state N	T	T
Cofig- Climb	F	F
dA- Climb	F	·
idtQAT > A	·	F

State

Ofrtair A s Oplde Mga



References: 2.5(↑) S 7.2, (↓) AT CKNG. Climb

Appears In:

DEFINITION

INITIALLY ⟶ Unknown

teru

Unknown, Not-Inhibited ⟶ Inhibited

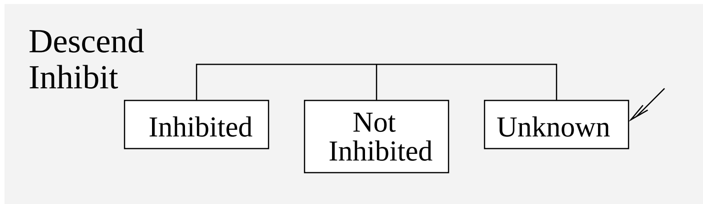
in state N	T	T	·
$t_{v_{200}}$ Dlla T ≥ A A	T	·	·
Cofig- Climb $t_{v_{200}}$ = Teru	·	T	·
Incr $t_{v_{200}}$ Climb = Teru	·	·	T

Unknown, Inhibited ⟶ Not-Inhibited

Cofig- $t_{v_{200}}$ Climb	F	F
Alt- $t_{v_{200}}$ Climb	F	·
$t_{v_{200}}$ Dlla T ≥ A A	·	F
Incr $t_{v_{200}}$ Climb = Teru	F	F

State

Own-Aircraft-Operating-Mode ↪



References Altitude ... (↑) (↑) S
(2),1000 ...
(↓) AT CKT Ground ... type ... de

Appears In ... Climb ... Che

DEFINITION

INITIALLY ⟶ Unknown

true

Unknown, Inhibited ⟶ Not-Inhibited

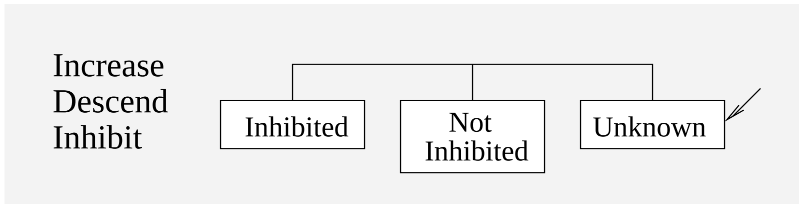
(Own-Tracked-Alt-Gr ≥ 1200ft (NODESHI)	T

Unknown, Not-Inhibited ⟶ Inhibited

(Own-Tracked-Alt-Gr ≤ 1000ft (NODESLO)	T

State

Own-Aircraft-Outputs
↪



References: 2.6.1.2, (↑) S 7.3

Appears In:

DEFINITION

INITIALLY ⟶ Unknown

true

Unknown, Inhibited ⟶ Not-Inhibited

Radio-Altimeter-Status$_{v\text{-}248}$	= Valid	F	·
Own-Tracked-Alt-Rate$_{v\text{-}247}$ > 1450ft$_{\text{(ZNOINCDES)}}$		·	F

Unknown, Not-Inhibited ⟶ Inhibited

Radio-Altimeter-Status$_{v\text{-}248}$	= Valid	T
Own-Tracked-Alt-Rate$_{v\text{-}247}$	≤ 1450ft$_{\text{(ZNOINCDES)}}$	T

State

Ownl Aircraft-M

↪ Radio-Altitude

Possible Values: -20...2150

Units: Feet

Granularity: 1...10 ... acc

Capacity: 1 r/s f ... (CAS ...) ... har

Description: (FAC ...) ... abov

Exception Handling: ... dl

Comments: ... hardwar ... t ... than ... (CAS ...) ... thus aff ... hardwar ... Anal ... CAS ... bo ... D ... v ... C 429 data w

References: (AIRDA ...)

Appears in: ... DA ... , ... Gr

DEFINITION

= Field(**Radio-Altitude**)

Receive ... Radio-Alt ... Report from ... Radio-Alt	T

= Prev(**Radio-Altitude**)

Receive ... Radio-Alt ... Report from ... Radio-Alt	F

State

Own-Aircraft-
↪ **Radio-Altimeter-Status**

Possible Values: {Valid, ...}

Capacity: ... (acc ... Radio-Altimeter ... value

Description: Val ... value

Comments: ... 1 ...

References: (↓) RADAROK

Appears in: ...

DEFINITION

= Field(**Status**)

Receive from Status ... Radio-Altimeter	T

= Prev(**Radio-Altimeter-Status**)

Receive from Status ... Radio-Altimeter	F

State

Own-Aircraft →

Barometric Altitude

Possible Values: -1200 to ... ft

Units: Feet

Granularity: 100 max ... (data us ... C ... n ... than that us

Capacity: 1r/s fo ... (actual rat

Description: Barom ... ata fr ... data ... lava ... C ... Mode ... C ... A ... ari ... teas D C or Mode C data.

Exception Handling:

Comments: ADC ... at ≤10 ft.

References: (1) ... data ... 3-2 3), (1) C, (2) ROWN.

Appears In: Alt ... L, Own-Aircraft ... Rat ... Own-Aircraft ...

DEFINITION

= Field(**Coarse-Altitude**)

	Received-Mode-C-Altitude	T
$t_{\text{Own-Alt}} \geq t_{\text{now}-250}$	Coarse	T

= Field(**Fine-Altitude**)

	Received-Mode-C-Altitude	T
$t_{\text{Own-Alt}} \geq t_{\text{now}-250}$	Fine	T

= Prev(**Barometric-Altitude**)

Received-Mode-C-Altitude	F

kdown-Aft-

↪ Barometric-Altimeter-Status

Possible Values: {Fi Clears}

Capacity: 1/s

Description: Fi ular 100 ft. Clears ular 00 ft.

Comments: Alt
than 100ft.

References: (⇓) A RDATA

Appears In: -Alt, -Bar Od-Alt-Trac Rat, Od-Alt-Trac

DEFINITION

= FIELD(**Barometric-Altimeter-Status**)

RECEIVED-VALID Aircraft-D ... Aircraft D	T

= PREV(**Barometric-Altimeter-Status**)

RECEIVED-VALID Aircraft-D ... Aircraft D	F

State

Own-Aircraft-Mode
↪ Standby

Possible Values: {True, False}

Capacity: 1 r/s f ... (hardwar

Appears In: Stand-by ... Ground

DEFINITION

= FIELD(**Standby-Discrete-Input**)

RECEIVED-VALUE(Aircraft-D... is ... Aircraft D	T

= PREV(**Standby**)

RECEIVED-VALUE(Aircraft-D... is ... Aircraft D	F

State

Own-Aircraft-
↪ Air-Status

Possible Values: {Airborne, On-Ground}

Capacity: 1/s

Description: status On-Ground status Own-Aircraft

References: (↓) OOGROUN

Appears In: Ground

DEFINITION

= Field(**Own-Air-Status**)

Received-Valid-Aircraft-Data … Aircraft Data	T

= Prev(**Air-Status**)

Received-Valid-Aircraft-Data … Aircraft Data	F

State

kOwn-Aft-
↪ Altitude-Rate

Possible Values: Ufsp

Units: ft/s

Granularity: Ufsp

Capacity: 1r/s fo CAS.

References: (D) _datapse 3-2 3)7(A) D C.

Appears In: Own-Trac Rat

DEFINITION

= Field(**Coarse-Altitude-Rate**)

Receipt M Trans	T
= r-Status-Altom$_{250}$ Coars	T

= Field(**Fine-Altitude-Rate**)

Receipt M Trans	T
= Fine-Status-Altom$_{250}$	T

= Prev(**Altitude-Rate**)

Receipt M Trans	F

↪ Own-Aircraft-Mode-S-Address

Possible Values: 1...$(2^{24} - 2)$

Granularity: 1 (One unit)

Capacity: 1/s

Description: [illegible] trans

Exception Handling: [illegible] fa [illegible] 0s or all 1s). Th [illegible] CAS 5 t 0s [illegible] [illegible] CAS [illegible] w

Comments: [illegible]

References: [illegible] _data [illegible] 3-2 3) D (↓) OWN

Appears In: [illegible]

DEFINITION

= Field(**Own-Mode-S-ID**)

Receive Own [illegible]	T

= Prev(**Own-Mode-S-Address**)

Receive Own [illegible]	F

State

Own-Aircraft-

→ Traffic-Display-Permitted

Possible Values: {True, False}

Capacity: 1/s

Description: d Tru c (TA display ac-

References: (↓ GROUND-MODE)

Appears In: ASL, Stand Ground

DEFINITION

= Field(**Traffic-Display-Permitted**)

Reception-Aircraft-D...	T

= Prev(**Traffic-Display-Permitted**)

Reception-Aircraft-D...	F

State

[illegible] ↪ Aircraft-Altitude-Limit

Possible Values: [illegible]

Units: [illegible]

Granularity: [illegible]

Capacity: 1/s

Description: Th [illegible]

Comments: [illegible]

Appears In: [illegible]

DEFINITION

= FIELD(**Aircraft-Altitude-Limit**)

RECEIVED [illegible] Aircraft-D [illegible] Aircraft D	T

= PREV(**Aircraft-Altitude-Limit**)

RECEIVED [illegible] Aircraft-D [illegible] Aircraft D	F

State

Other-Aircraft ↪ Proximate-Traffic-Display

Possible Values: {True, False}

Capacity: 1/s

References: (↓) PALL ROX

Appears In:

DEFINITION

= FIELD(**Proximate-Traffic-Display**)

RECEIVED(Own-Aircraft-D ... sets Aircraft D ...)	T

= PREV(**Proximate-Traffic-Display**)

RECEIVED(Own-Aircraft-D ... sets Aircraft D ...)	F

State

Own-Aircraft-Model ↪ Config-Climb-Inhibit

Possible Values: {True, False}

Capacity: 1 r/s for CAS.

Definition: [illegible]

[illegible]

References: (↓) Climb

Appears In: Climb, Climb

DEFINITION

= FIELD(**Config-Climb-Inhibit**)

RECEIVED Aircraft-D … Aircraft D	T

= PREV(**Config-Climb-Inhibit**)

RECEIVED Aircraft-D … Aircraft D	F

State

Own-Aircraft-

↪ Altitude-Climb-Inhibit

Possible Values: {True, False}

Capacity: 1/s

Description: Th ... ndisimp ... Climb ... (Prev) ... evaluat

References: (↓) Climb ... evaluat

Appears In: ... Climb ... Climb

DEFINITION

= FIELD(**Altitude-Climb-Inhib-Acti**)

RECEIVED Own-Aircraft-D ... status Own-Aircraft D	T

= PREV(**Altitude-Climb-Inhibit**)

RECEIVED Own-Aircraft-D ... status Own-Aircraft D	F

State

Own-Aircraft-
↪ Increase-Climb-Inhibit-Discrete

Possible Values: {True, False}

Capacity: 1/s

References: (↓) Increase-Climb-Inhibit

Appears In: Increase-Climb-Inhibit

DEFINITION

= FIELD(**Increase-Climb-Inhibit-Discrete**)

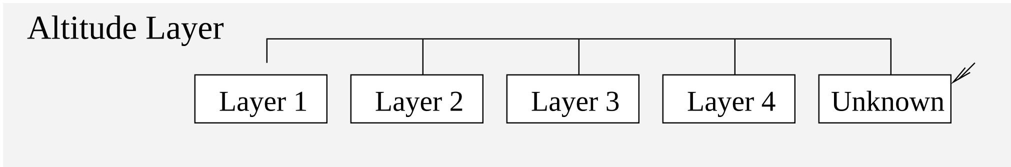
RECENT-VALUE(Own-Aircraft-Discretes) ≠ Own-Aircraft Discretes	T

= PREV(**Increase-Climb-Inhibit**)

RECENT-VALUE(Own-Aircraft-Discretes) ≠ Own-Aircraft Discretes	F

State

Own-Aircraft-

↪ 

References: (↓) PROGRESS 3-5), (↓) TRACKING ay pd para

Appears In: DOWN ... N ... OK

DEFINITION

INITIALLY ⟶ Unknown

teru

Unknown, 2 ⟶ 1

Own-Alt-Trac $_{437}$ [Bel-Alt-Lay 2]	T

1 ⟶ 2

Own-Alt-Trac $_{437}$ ≥ op [Alt-Lay-T]	T

Unknown, 3 ⟶ 2

Own-Alt-Trac $_{437}$ [Bel-Alt-Lay 3]	T

2 ⟶ 3

Own-Alt-Trac $_{437}$ ≥ op [Alt-Lay-T 2]	T

Unknown, 4 ⟶ 3

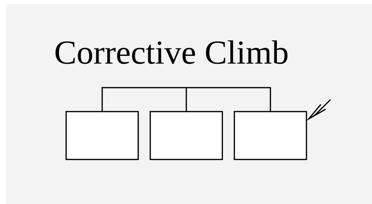
Own-Alt-Trac $_{437}$ [Bel-Alt-Lay 4]	T

Unknown3, $\longrightarrow$ 4

Own-Attac $_{437}$ $\geq$ opr[-Thay-T 3]	T

leMOwn-Aft-

↪



References: (t↓) D _gal _ra(t↓) Gcetr eipnt _tst.

Appears In: Combi Cltr mcDtr
HgCtr CdgnV Cltr

DEFINITION

INITIALLY ⟶ Unknown

teru

ANY ⟶ Yes

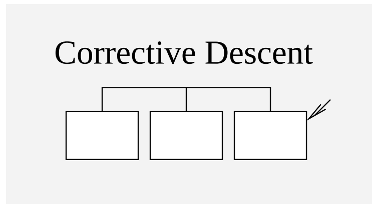
Dnt- Cld-D	T	F	·
Own-Trac Rat$_{f438}$ $<$ (f0/ 300f/	T	·	·
Own-Trac Rat$_{f438}$ $<$ Climbal$_{f-382}$ $-$ 300f/	·	T	·
Nscr-I Climb$_{383m}$	·	·	T

ANY ⟶ No

Dnt- Cld-D	T	T	F	F
Od-ATrac Rat$_{f438}$ $<$ 500f/	T	T	·	·
Climb RW	T	·	T	·
Incr RAd$_{m327}$	·	T	·	T
Od-ATrac Rat$_{f438}$ $\geq$ (f0/ 300f/	T	T	·	·
Od-ATrac Rat$_{f438}$ $\geq$ Climbal$_{f-382}$	·	·	T	T

State

Own-Aircraft-Model



References: (↑) D _goal _rate(↓) Corrective input _set.

Appears In: Combined Control Horizontal Change, Vertical Control

DEFINITION

INITIALLY ⟶ Unknown

true

ANY ⟶ Yes

Dont-Climb-Descend	T	T	·	F
Increase-I	·	·	T	·
Corrective Climb in state Yes	T	·	·	·
Own-Alt-Rate > 500 ft/ ... 3000 ft/	T	T	·	·
Own-Alt-Rate > Goal ... 300 ft/	·	·	·	T

ANY ⟶ No

Dont-Climb-Descend	T	T	T	F	F	F	F	F
Climb RAW	·	F	·	·	·	F	·	·
Increase RA	·	F	T	·	·	F	T	T
Corrective Climb in state N	T	·	T	T	·	·	T	·
Own-Alt-Rate < 500 ft/	T	T	T	·	·	·	·	·
Reduce RAW	T	T	·	T	T	T	T	T
Own-Alt-Rate ≤ Goal	·	·	·	T	T	T	T	T

State

Own-Aircraft-

↪ TA-In-Sens-Level-2

Possible Values: {True, False}

References: (↓) LOWTA

Appears In:

DEFINITION

= Field(**LOWTA-Discrete**)

Received Valid Own-Aircraft-Discrete, set in Own Aircraft Discrete	T

= Prev(**TA-In-Sens-Level-2**)

Received Valid Own-Aircraft-Discrete, set in Own Aircraft Discrete	F

Own-Aircraft → Composite-RA

Possible Values: No-RA, Climb, Descend, Neither

Comments: See RA-Str$_{s277}$ for RA.

References: (↑) Select 2.27 (combine RA) (↑) RA Display Function 2.30, (↓) variable RA. (↓) ROUTE Results Update

Appears In: Climb-Global, Climb Combine Control, Descend, Cast, Increase Climb, Increase Descend, Own-Global-Alt, Rate, Advisory, Vertical Control

DEFINITION

= No-RA

Some Sense$_{s276}$ **in state** Climb		F
Some Sense$_{s276}$ **in state** Descend		F

= Climb

:there exists

Sense$_{s276}$[i] **in state** Climb		T
[RA-Str$_{s277}$ i] **in one of** Nominal 2500 fpm		T

= Descend

:there exists

Sense$_{s276}$[i] **in state** Descend		T
[RA-Str$_{s277}$ i] **in one of** Nominal 2500 fpm		T

= Ngti

Some RA-Str$_{s277}$ **in state** N		T	·
Some RA-Str$_{s277}$ **in state** 2500 fpm		·	T

State

Own-Aircraft-
↪ Composite-RA
↪ Not-Neg
↪ **Climb-VSL**

Possible Values: No-Climb-VSL, VSL 0, VSL 500, VSL 1000, VSL 2000
References: (↓) Manual-CPS var RA, (↓) ROUTE Reduced-Input-Update
Appears In:

DEFINITION

= No-Climb-VSL

:for all

[Sens s-275 i] **in state** Climb	F	·
RA-Str [s-277 i] **in one of** VSL-2000 fpm, VSL-1000 fpm, VSL-500	·	F

= VSL0

Some RA (s-288) Climb VSL- (fpm)	T

= VSL500

Some RA (s-288) Climb VSL- (fpm)	F
Some RA (s-288) Climb VSL-5 (fpm)	T

= VSL1000

Some RA (s-288) Climb VSL- (fpm)	F
Some RA (s-288) Climb VSL-5 (fpm)	F
Some RA (s-288) Climb VSL-1000 (fpm)	T

= VSL2000

Som R(Al Climb VSL-	F
Som R(Al Climb VSL-5	F
Som R(Al Climb VSL-	F
Som R(Al Climb VSL- 2000	T

Own-Aircraft

↪ Composite RA

↪ Descend-VSL

Possible Values: No-Descend-VSL, VSL 0, VSL 500, VSL 1000, VSL 2000

References: (↓) Mode-C/GPS var RA, (↓) ROUTE Reduce Input Update

Appears In:

DEFINITION

= No-Descend-VSL

for all :

[Sense $_{s\text{-}275}$ i] **in state** D	F	·
[RA-Str $_{s\text{-}277}$ i] **in one of** VSL-2000 fpm, VSL-1000 fpm, VSL-500	·	F

= VSL0

Some p (RA) $_{s\text{-}8}$ VSL-	T

= VSL500

Some p (RA) $_{s\text{-}8}$ VSL-	F
Some p (RA) $_{s\text{-}8}$ VSL-500	T

= VSL1000

Some p (RA) $_{s\text{-}8}$ VSL-	F
Some p (RA) $_{s\text{-}8}$ VSL-500	F
Some p (RA) $_{s\text{-}8}$ VSL-1000	T

= VSL2000

Some p (RA) $_{s\text{-}8}$ VSL-	F
Some p (RA) $_{s\text{-}8}$ VSL-500	F
Some p (RA) $_{s\text{-}8}$ VSL-1000	F
Some (RA) $_{s\text{-}8}$ VSL- 2000 fpm	T

State

Other-Aircraft [Opposite Threat i]
↪ Classification

Possible Values: { Other-Traffic, Proximate-Traffic, Potential-Threat, Threat }

References: (1) S 2.3 ... alt. r (H or_miss ...) (1) S 7.1 TRAFFIC_ADV ... (traffic).

Appears In: Classification Reversal

DEFINITION

Proximate-Traffic ⟶ Other-Traffic

Altitude-Reporting$_{s\text{-}293}$ **in state** Lost	T	T	T	·	·	·
Altitude-Reporting$_{s\text{-}293}$ **in state** No	·	·	·	T	·	·
Altitude-Reporting$_{s\text{-}293}$ **in state** Yes	·	·	·	·	T	·
Bearing-Valid	F	·	T	·	·	·
Other-Range-Valid$_{v\text{-}296}$ True	·	F	T	·	·	·
Proximate-Traffic-Condition$_{m\text{-}343}$	·	·	F	F	F	·
Potential-Threat-Condition$_{m\text{-}340}$	·	·	F	F	F	·
Threat-Condition$_{m\text{-}373}$	·	·	·	·	F	·
Other-Air-Status$_{s\text{-}297}$ **in state** On-Ground	·	·	·	·	·	T

Potential-Threat ⟶ Other-Traffic

Altitude-Reporting$_{s\text{-}293}$ **in state** Lost	T	T	·	·
Bearing-Valid	F	·	·	·
Other-Range-Valid$_{v\text{-}296}$	·	F	·	·
Potential-Threat-Range-Test	T	T	·	·
Other-Air-Status$_{s\text{-}297}$ **in state** On-Ground	·	·	T	·
Potential-Threat-Condition$_{m\text{-}300}$	·	·	·	T
DURATION(Potential-Threat-Condition$_{m\text{-}300}$) $\geq$ 8	·	·	·	T
Proximate-Traffic-Condition$_{m\text{-}343}$	·	·	·	F
Potential-Threat-Condition$_{m\text{-}340}$	·	·	·	F
Threat-Condition$_{m\text{-}373}$	·	·	·	F

Threat ⟶ Other-Traffic

Alt-Reporting$_{s\text{-}293}$ **in state** L	T	T	T	·
[illegible] Val	F	·	T	·
Other-[illegible]-Val$_{v\text{-}296}$	·	F	T	·
[illegible] c-Condit$_{m\text{-}343}$	·	·	F	·
[illegible] Thr Condit$_{m\text{-}340}$	·	·	F	·
Other-Status$_{s\text{-}297}$ **in state** On-Gnd	·	·	·	T

Other-Traffic ⟶ Proximate-Traffic

Alt-Reporting$_{s\text{-}293}$ **in state**	F	T
[illegible] Val	T	·
Other-[illegible]-Val$_{v\text{-}296}$	T	·
[illegible] c-Condit$_{m\text{-}343}$	T	T
[illegible] Thr Condit$_{m\text{-}340}$	F	F
[illegible] Condit$_{m\text{-}373}$	·	F

Potential-Threat ⟶ Proximate-Traffic

P[illegible] Condit$_{m\text{-}300}$)	T
D[illegible] Condit$_{m\text{-}300}$) ≥ 8 [illegible]	T
[illegible] Condit$_{m\text{-}343}$	T
[illegible] Thr Condit$_{m\text{-}340}$	F
[illegible] Condit$_{m\text{-}373}$	F

Threat ⟶ Proximate-Traffic

Alt-Reporting$_{s\text{-}293}$ **in state** L	T
[illegible] Val	T
Other-Range-Val$_{v\text{-}296}$ True	T
[illegible] Condit$_{m\text{-}343}$	T
[illegible] Thr Condit$_{m\text{-}340}$	F

Other-Traffic, Proximate-Traffic ⟶ Potential-Threat

Add Reply to in state	F	T
Bear Val	T	·
Other-Relative-Val	T	·
Potential-Threat Condition	T	T
Threat Condition	·	F

Threat ⟶ Potential-Threat

Add Reply to in state L	T	·
Bear Val	T	·
Other-Relative-Val	T	·
Potential-Threat Condition	T	·
PREV(Threat Range)	·	F
DURATION(Clear state Threat 5.5.1 s	·	T
Threat Range-T	·	F

Continued (↑) S 2.34

Other-Traffic, Proximate-Traffic, Potential-Threat ⟶ Threat

Threat Condition	T

State

Ownir-Aircraft- [Operat i]
↪ Advisory-Status
↪ TA/RA
↪ **Displayed -Advisory**

Possible Values: TA, RA

References: (↑) TCAS-TCAS (2.65.3), (↓) RESOLUTION_-AND_COORDINATION.TCAS_ [illegible], (↓) Reversal_ [illegible]

Appears In: [illegible]

DEFINITION

INITIALLY ⟶ TA

[illegible]	T

INITIALLY ⟶ RA

[illegible]	F

TA ⟶ RA

[illegible]	F	T
DURATION([illegible] **in state** TA) ≥ 2 [illegible]	·	T

State

Own-Aircraft-Operation [Climb...]
↪ Climb-Class
↪ Tauto-Alt
↪ **Sense**

Possible Values: {Descend, Climb}

Mode/Component: OPS, SRS

References:

(↑) TCAS/TCAS-Aircraft ... 2.65),

(↓) RESOLUTION_AND_COORDINATION... Sense

Appears In: Climb-VSL, Catch-Low-Off, Compo RA, Cross, Int-Cross-N Cycle, Reversal-Ach, Reversal, Strat, Sense, RA, VSL-, Rate, VSL-OK

DEFINITION

INITIALLY ⟶ Climb

Other-V RC$_{v-289}$ No	T	·	·
Other-V RC$_{v-289}$ N	·	T	T
Tau-Cap$_{f-423}$ < Tem [GRA ... D Coflict-SL$_{f-385}$, Other-Tracness$_{f-43}$]	·	F	T
Climb$_{f-396}$(n ... Dampen(S	·	T	·
Nocr ... Climb$_{338m}$	·	·	T
Non-Car Choose(D	F	F	F

Column 1 (↑) S 2.49 2(1), Row 6 (↑) S 2.49 2(2)

INITIALLY ⟶ Descend

Other-V RC$_{v-289}$ No Climb	T	·	·
Other-V RC$_{v-289}$ N	·	T	T
Tau-Cap$_{f-423}$ < Tem [GRA ... D Coflict-SL$_{f-385}$, Other-Tracness$_{f-43}$]	·	F	T
Climb$_{f-396}$(n ... Dampen(S	·	T	·
Nocr	·	·	T
Non-Car Choose(Climb)	F	F	F

Column 1 (↑) Sect 2.49.2(1), Row 6 (↑) Sect 2.49.2(2)

Climb ⟶ Descend

PREV(Ens state D	T	·
PREV(Ens state Climb)	·	T
Reversal-	F	T

Descend ⟶ Climb

PREV(Ens state Climb)	T	·
PREV(Ens state D	·	T
Reversal-	F	T

State

Onda-Aft- [seOyopo at i]
↪ nGleas
↪ Tatr
↪ **RA-fong**

Possible Values: {ele, VSLN pon0 VSL-
p, 00VSL-
-ens cr 2500 pm}

Comments: Starts (see aus
lo ns chor sur
tr)
nds ult r sh ogi th 7hh c rf th
ds VSL ans

References: (↑) S vs (2.5, s tr
p (↓) S Ardv n Se 2.5), (↑) S a
Val Ra (2.56, 2.5 ps) (↑)
(2.58, 2.59), (↑) Ra Ai (y 2.80),
(↓) RESOLUTION_AND_COORDINATION S _-
ny, (↓) Try _VSL, (↓) VSL_ov _int al, (↓) VSL _test,
(↓) RESOLUTION_AND_COORDINATION nt _-
nds t

Appears In: Cli-VSL, Com po RA, Ci VSL,
-ens cr Ra N Old-Ra, Ra-Str
-n, S Re sal- RA, VSL- OK,
Val- Co ntr

DEFINITION

= Stg n-Not- Selected

R y-D	T

= Nominal-1500fpm

Tau-Rpt-T	T	T	·
Test-VSL-T	F	T	·
Other-Track Range-Rate > 0	·	F	·
Test-VSL-Rate (/	·	F	·
-Test Climb	F	F	·
PREV(Init-Str) in state 2500 fpm	·	·	T
DURATION(Init-Str) in state 2500 fpm > 10	·	·	T
Incr Climb	·	·	F
DEADA	F	F	F

(2.5, cas 2)

= VSL-2000fpm

Tau-Rpt-T	T	T
Test-VSL-T	T	·
Other-Track Range-Rate > 0	F	·
Test-VSL-Rate (2000 /	T	·
True-Tau-Capped $\leq$ 2.5	·	F
Test-VSL-T	·	T
Other-Track Range-Rate > 0	·	F
Test-VSL-Rate (2000 /	·	T
NW	·	F

= VSL-1000fpm

PREV(...) in one of N VSL-1000 fpm	·	T	·
PREV(Str) in state VSL-2000 fpm	·	T	T
Tau-Rpt-T	T	T	T
Try-VSL-T	T	T	·
Other-Track Range-Rate > 0	F	F	F
Test-VSL-Rate (2000 /	F	F	F
Test-VSL-Rate (1000	T	T	T
True-Tau-Capped $\leq$ 2.5	·	F	F
NW	·	F	·
Str	·	·	F

= VSL-500fpm

Prev(Str) **in one of** N	·	T	·
Prev(Str) **in one of** VSL- 2000	·	T	T
Tar Range T	T	T	T
Try-VSL-T	T	T	T
Other-Trac Range- Rate >	F	F	F
Test-VSL-Rate (	F	F	F
Test-VSL-Rate (5	T	T	T
Tau- Cap ≤ 2.5s	·	F	F
AW	·	F	·
N-Str	·	F	F

= VSL-0fpm

Prev(Str) **in one of** VSL-5 VSL- 2000	·	·	·	T	T	T	·
Tar Range T	T	T	T	T	T	T	·
Try-VSL-T	T	T	·	T	T	·	·
Other-Trac Range- Rate >	·	T	·	·	T	·	·
Test-VSL-Rate (5	F	·	·	F	·	·	·
Test-VSL-Rate (	T	·	F	T	·	F	·
-Alt Chec	·	·	T	·	·	T	·
Tau- Cap ≤ 2.5s	·	·	·	F	F	F	·
N-Str	·	·	·	F	F	F	·
Abo	·	·	·	·	·	·	T
De	·	·	·	·	·	·	T

lus/cro raft (t) M 2.59 cas 3)

Nominal-1500fpm ⟶ 2 Increase- 500fpm

(ent Eval	T
-Incr Chec	T
Vertical-A	F

ric (t) s 2.57.5.1 and 2.57 raft (t) M (2.59 cas 2)

State

Other-Aircraft-i [Other-Aircraft-Operating i]
↪ Intruder-Status
↪ Threat
↪ **Crossing**

Possible Values: Non-Crossing, Int-Cross, Own-Cross

References: (↑) Sense Selection (2.48), (↑) S 2.66, 2.66.1, 2.66.2), (↓) DISPLAY_ADVISORY ORIES. Crossing_flag_dec

Appears In: Increase-Climb-Fi, Separation, SMR-Reversal-, Reversal-TV-S, Reversal-TV, V Control

DEFINITION

INITIALLY, Non-Crossing ⟶ Int-Cross

1-100 Crossing	T	T	T
Own-Goal-Alt Rate = 0	T	·	·
Other-Tracked-Relative-Alt-Rate$_{f-436}$ ≥ 600 ft/min (V$_{IDLE}$)	·	T	T
Own-Goal-Alt Rate > 0	·	T	·
Own-Tracked-Alt$_{f-437}$ > Other-Alt$_{f-430}$	·	T	·
Own-Goal-Alt Rate < 0	·	·	T
Own-Tracked-Alt$_{f-437}$ < Other-Alt$_{f-430}$	·	·	T

Own-Cross ⟶ Int-Cross

Equipped TCAS	F	F
True-Tau-Capped$_{f-423}$ > 4.0 s (MIN_RE)	T	T
Tau-Rising$_{v-350}$	F	F
Increase-Alt-Str$_{s-277}$ **in state** 2500 ft/min	F	F
Sense$_{s-275}$ **in state** Climb	T	·
Own-Alt-Track$_{f-437}$ > Other-Alt$_{f-430}$	T	·
Sense$_{s-275}$ **in state** Descend	·	T
Own-Alt-Track$_{f-437}$ < Other-Alt$_{f-430}$	·	T

INITIALLY, Non-Crossing ⟶ Own-Cross

100 Gpss	T	T	T	T
Other-Trac Rel-Alt Rate$_{f\text{-}436}$ $\geq$ 600/min	F	·	·	·
Own-Goal-Alt Rate$_{f\text{-}432}$ > 0	·	F	·	·
Own-Alt-Trac$_{f\text{-}437}$ $>$ Goal-Alt$_{f\text{-}430}$	·	·	F	F
Own-Goal-Alt Rate$_{f\text{-}432}$ ≤ 0	·	·	F	·
Own-Alt-Trac$_{f\text{-}437}$ Goal-Alt$_{f\text{-}430}$	·	F	·	F

INITIALLY, Own-Cross, Int-Cross ⟶ Non-Crossing

100 Gpss	F	F
Sens$_{s\text{-}275}$ **in state** Climb	T	·
Own-Alt-Trac$_{f\text{-}437}$ $\geq$ Other-Alt-Trac$_{f\text{-}432}$ 100 ft	T	·
Sens$_{s\text{-}275}$ **in state** D	·	T
Own-Alt-Trac$_{f\text{-}437}$ $\leq$ Other-Alt-Trac$_{f\text{-}432}$ $-$ 100 ft	·	T

State

Other-Aircraft [Own-Aircraft i]
↪ Intruder-Status
↪ Threat
↪ **Reversal**

Possible Values: Not-Reversed, Reversed

Comments: Note that this state is not the o ... Climb ... D

Appears In: Other-Aircraft ... -Fi ... Threat-Reversal-TSV, V ... Control

DEFINITION

INITIALLY ⟶ Not-Reversed

True

Not-Reversed ⟶ Reversed

PREV(Intruder-Status$_{s-274}$) in state Threat	T	T
PREV(Sense$_{s-275}$) in state Climb	T	F
Sense$_{s-275}$ in state D	T	F

[ldeOCotmeraAft-i]

↪ Altitude

Units: Ftee

Granularity: in100ax

Capacity: 1/s

Description:

byting

References: (↑) Ospitysl (D 2.8)1

Appears In:

DEFINITION

= Field(ZR)INT

e Sainc)i eRyMteh	T

= Prev(A)ltitude

e Sanrc)i eRyMteh	F

State

[Other-Aircraft-Model i]

↪ Altitude-Valid

Possible Values: {True, False}

References: (ZFLG)

Appears In:

DEFINITION

= Field(**ZFLG**)

Samc …) i ns Bettoh	T

= Prev(**Altitude-Valid**)

Samc …) i ns Bettoh	F

State

[hMonitor-Aft- i]
↪ **SurveilID**

Possible Values: U...

Granularity: 1

Description: ... Aft un

Comments: M... th
C... acc

References: (↓) SURVNO

Appears In: Monitor-Aft

DEFINITION

= Field(**SURVNO**)

e Surv) i ...	T

= Prev(**Surveillance-ID**)

e Surv) i ...	F

State

[Own-Aircraft-Model ... i]

↪ Equipment

Possible Values: {TCAS, TCAS/TA, MODE-S, ATCRBS}

References: (↓) EQ

Appears In: [illegible]

TCAS-TCAS-Coordination

DEFINITION

= Field(**EQP**)

[illegible]	T

= Prev(**Equipment**)

[illegible]	F

[Mod Other-Aft- i]

→ Other-Bearing

Possible Values: Undefined

Units: Undefined

Granularity: Undefined

References: (↑) Conflict Display (2.8.3), BEAR (↓) R

Appears In: Intruder

DEFINITION

= FIELD(**BEAR**)

e Same) i Beyond Match	T

= PREV(**Other-Bearing**)

e Same) i Beyond Match	F

State

[Month-After- i]

↪ Other-Bearing-Valid

Possible Values: {True, False}

References: (↓) BEAR OK

Appears In:

DEFINITION

= FIELD(**BEAR OK**)

Bearing Source (i) Matches	T

= PREV(**Other-Bearing-Valid**)

Bearing Source (i) Matches	F

[Other-Aircraft i] ↪ Other-VRC

Possible Values: {None, Dont-Descend, Dont-Climb}

References: (↓) OTHRARS(↓), (↓) Received_data

Appears In: Conflict-Test, Conflict-Status-Low-Firmness, Other-Aircraft, Reply-ID, Reply-TCAS-TCAS, Traffic-RAC

DEFINITION

= None

Received-Mode-S-Resolution-Message	T
[Other-Aircraft i] ▷ Mode-S-Addr = Field(MID)	T
Field(VRC) = 0	T

= Dont-Descend

Received-Mode-S-Resolution-Message	T
[Other-Aircraft i] ▷ Mode-S-Addr = Field(MID)	T
Field(VRC) = 1	T

= Dont-Climb

Received-Mode-S-Resolution-Message	T
[Other-Aircraft i] ▷ Mode-S-Addr = Field(MID)	T
Field(VRC) = 2	T

State

[le Mo Ot h ra Aft- i]

↪ Other-HRC

Possible Values: {None, Dont-Turn-Left, Dont-Turn-Right}

References: (↓) OTHRAR(2) (↓) _val _data

Appears In:

DEFINITION

= None

Re-Val ... Result	T
[Other-Aircraft-i] ▷ ... S-Addr ... Field(MI)	T
Field(HRC) = 0	T

= Dont-Turn-Left

Re-Val ... Result	T
[Other-Aircraft-i] ▷ ... S-Addr ... Field(MI)	T
Field(HRC) = 1	T

= Dont-Turn-Right

Re-Val ... Result	T
[Other-Aircraft-i] ▷ ... S-Addr ... Field(MI)	T
Field(HRC) = 2	T

[Mother-Aft- i]
↪

Possible Values: {Not Known, 1, 2, 3, 4, 5, 6, 7}

Comments: C

P

References: (J) T

Appears In: Con-SL, D , L
-T-S,
-TV, S
T-VSL- -T OK

DEFINITION

= FIELD(**PLINT**)

e Sam) i P	T

= PREV(**Severity Level**)

e Sam) i P	F

[Other-Aircraft-Mode-i] ↪ **Other-Mode-S-Address**

Possible Values: Undefined, 1 ... ($2^{24} - 2$)

Granularity: 1 (int)

Description: Other-Aircraft un

Exception Handling: de ... wi (...)

Comments: ... has n

References: (D) T R

Appears In: Other-Aircraft-RC, Other-Mode-S-Addr Other-V RC, Ready-1-D ... Mode-S-Address Reversal-...

DEFINITION

= Field(IDINTR)

for all j

Real ... Message ...	T	T	T
[Other-Aircraft$_{s-44}$ i] in state N	F	T	T
[Other-Aircraft$_{s-44}$ i] ▷ Other-Mode-S-Addr	T	·	·
[Other-Aircraft k] in state N	·	T	F
i < j	·	T	·
[Other-Aircraft j] ▷ Other-Mode-S-Addr ... = ID MI	·	·	T

= Prev(Mode-S-Address)

... Message ...	F

State

[hMOtoru-rAft- i]

↪ Apltitude-Re ortig

Possible Values:

References: (↓) TRACKING T RACK_INTRUDERS

Appears In: nClcast Coebi wCshyArr aiedgThr Gion ateThr fCtiePr c-Gion ofSTcAlt-T

DEFINITION

Yes ⟶ Lost

Ssanc) iRMytch	T
FIELD(MODC) eals "F	T

INITIALLY,Lost,No ⟶ Yes

Ssanc) iRMytch	T
FIELD(MODC) eTru"	T

INITIALLY ⟶ No

Ssanc) iRMytch	T
FIELD(MODC) eals "F	T

State

[k Cotr Aft- i]
↪ **Rng**

Possible Values: U

Units: naut

Granularity: U

References: (↑) Onl D 2.8), (↓) RR

Appears In: Otr Trac Rag Ra Otr Trac Rag

DEFINITION

= FIELD(**R**)**R**

e Sanv) i Pyte

T

= PREV(**R**gan

e Sanv) i Pyte

F

State

[leMonitor-Aft- i]

↪ Range-Report-Time-Stamp

Possible Values: 0 ... U

Units: s

Granularity: 1

Description: Th Range Report

References: (↓) RRTIME

Appears In: Trac Range Rat, Trac Range

DEFINITION

= FIELD(**RRTIME**)

e Sainc) i RMrtch	T

= PREV(**Range-Report-Time-Stamp**)

e Samc) i RMrtch	F

[kdeomMotkeraAft- i]

↪ Other-Rng Valid

Possible Values: {True, False}

References: (↓) RFLG

Appears In: Initial Attack, Fit, Other Rng Valid, Other-Track Range, Range, Other-Track Range Two, Off-Thr

DEFINITION

= Field(**RFLG**)

e Same) i Byte Match	T

= Prev() **Other-Valid-Range**$_{s\text{-}296}$

e Same) i Byte Match	F

State

[lOther-Aircraft- i]

↪ **Other-Air-Status**

Possible Values: On-Ground, A

References: (↓) TRACKING Ground ty_pomi _de

Appears In: tiMGlast Grotind

altoThr ffeCTnind c-Grotind

Tahtr Grotind

DEFINITION

= On-Ground

(lOt-ell-10T ftac- OE)534) 1 182 (NEARGR	T

= Airborne

(lOt-ell-Trac- 1534) 1 432 200ft OH) (NEARGR	T

State

[hMo Grnd-Stat i]

↪ **GH-Stratio**

Appears In: Grnd- Do Cmds-SH, Grnd-Stat

DEFINITION

= Field(**IIS**)

for all j:

Receive Physical Trans to Command From Mo	T	T	T
Pred-GH-Grnd-Stat(Do) = Not-In-Use	F	T	T
Pred-GH-Grnd-Stat(Do) = Field(SI)	T	·	·
Pred-Grnd-Stat[Do, j]) = Not-In-Use	·	T	F
i < j	·	T	·
Pred-Grnd-Stat[Do, j]) ≠ Field(SI)	·	·	T

[kMCommand-Stat i]

↪ Group-Command-SL-

Possible Values: Cancel, 2, 3, 4, 5, 6, 7

References: (↓) SL_command page (3-P21)

Appears In: Get-

DEFINITION

= Cancel

Received-SL	Command-Mo	T
Command-Stat $_{V298}$	Field(SI)	T
Field(SLC) = 1		T

= Field(SLC)

Received-SL	Command-Mo	T
Command-Stat $_{V298}$	Field(SI)	T
$2 \leq$ Field(SLC) ≤ 7		T

Macro

Fairing Rule ... leTCtr ... oncitio

References: n(t) S 7.1. TRAFFIC_ASV ORY.

Appears In: nClass

DEFINITION

eAld Rirto$_{es-293}$ **in** **sstate** L	T	T	F
dRaVal	F	·	·
Other- Range-Val$_{ev-296}$ Tru	·	F	·
afff Thr Rate$_{m-57}$	F	F	·
afff Thr Giotd$_{m-340}$	·	·	F

Rues l-Mle cSatio

References: (↑) 2,fi(c) 2.51 2 (ns 2 and 3), (↑) 2.51 3 (n 4), (↑) S 2.52 (al ne(t) s 2.57.5.1 and 2.57.5.2, (↑) TCAS/TCAS Func(2.65.2).

Appears In: Ses

DEFINITION

Tar Rage-T	T	T	T	T
Equip TCAS	T	F	F	F
Own-S-Addr > OMS-Addr	T	·	·	·
flict	T	·	·	·
Tau- Cap $_{f-423}$ > 4.0	·	T	T	T
Tau-Ris$_{s-350}$	·	F	F	F
PREV(R A-Str$_{s-277}$) **in state** 2500 ft	·	F	F	·
Cs$_{s-280}$ **in state** Crss	·	T	·	·
Cs$_{s-280}$ **in state** Own-Crss	·	·	T	·
Tmal-T	·	T	T	·
Clch-L Off$_{m-303}$	·	T	·	·
Int-Crs-N Cycl$_{m-304}$	·	·	F	·
Mult	·	·	T	·
Reversal-Ach	·	·	T	·
Sapr	·	T	T	·
Cs$_{s-280}$ **in state** N Crss	·	·	·	T
100 Crss	·	·	·	T
RA-Str$_{s-277}$ **in one of** N Incr 2500 ft	·	·	·	T

Macro

Conflicting

Appears in: Inputs...

DEFINITION

Sens$_{s-275}$ **in state** Climb	T	·
Sens$_{s-275}$ **in state** D	·	T
Other-V RC$_{v-289}$ = Do N Climb	T	·
Other-V RC$_{v-289}$ = D N D	·	T

Macro

Cali-Off

Appears in: Mode-Reversal-

DEFINITION

$Sens_{s-275}$ **in state** Climb	T	·
not $Sens_{s-275}$ **in state** D	·	T
Own-Tracon-Alt $_{f-437}$ < Ground-Alt $_{f-430}$	T	·
Own-Tracon-Alt $_{f-437}$ > Ground-Alt $_{f-430}$	·	T

Macro

InGNss eCC-yc

Appears in: IcapsStRe-sal-

DEFINITION

Sens$_{s\text{-}275}$ **in state**	Climb	T	·
not Sens$_{s\text{-}275}$ **in state** D		·	T
Own-Trac$_{f\text{-}437}$ >	OPelt-$_{f\text{-}430}$	T	·
Own-Trac$_{f\text{-}437}$	OPelt-$_{f\text{-}430}$	·	T

Spehio ime-Te

Appears in: Impl-Reversal-

DEFINITION

Sens$_{s\text{-}275}$ **in state** Climb	T	·	·
Sens$_{s\text{-}275}$ **in state** D	·	T	·
Own-Alt-Trac$_{f\text{-}437}$ Ot-Alt-Trac$_{f\text{-}132}$	T	·	·
Own-Alt-Trac$_{f\text{-}437}$ > Ot-Alt-Trac$_{f\text{-}132}$	·	T	·
Cl-SV ≤ 200ft	T	T	·
Ter-Tau- Cads10 >	·	·	T
100 Gss	·	·	T

Macro

Multiair Bias

Appears in: Ipsa SMR esal-

DEFINITION

Multi-Situation$_{n307}$	·	F
Other-Trac Rel Alt Rat$_{436}$ ≥ rmff/$_{(OLE)}$	T	·
Opposi Rat$_{n308}$	T	·

Macro

Multiairafitu atnio

Appears In: Rsi rilMaflt-

DEFINITION

some[Oideraft j] **in state** Thr	T
THIS $\neq$ j	T

Macro

Opposite-Rate

Appears In: Ri rilMaflt-

DEFINITION

(Odel-Alt- Raeand Gte-Trac ReAlt Raeppo
sig)
Oti-Cl-Alt Ra$_{t12}$. Gte-AlTrac Ra$_{t433}$ 0

Macro

Rus l-Clh reaStio

Appears in: SM Reversal-

DEFINITION

$Sens_{s-275}$ **in state** Climb	T	·
$Sens_{s-275}$ **in state** D	·	T
Ind- d_{ref} Reversal-	T	·
Climb- d_{ref} Reversal-	·	T

Macro

Des-Re l-d

Appears in Rosal-Ach

DEFINITION

[illegible]

Macro

Clim bmaRe l-Ferd

Appears in Iprs GRessal-Ach

DEFINITION

mdu)paS(tn ftrnDa(S

Macro

[Ondra-Aft i]

↪ **New-Tkr**

Appears In: Aural-Alar Gsti-VSL- Rat-VSL- OK

DEFINITION

d	**in** Otrshoate_Trac	T
	PRVI(**d**) **in** Otrshoate_Trac	F

Macro

100-Ft-Crossing

References: (↓) RESOLUTION.Cross_through_check

Appears In: Crossing ... SV-al- ... Tr ... -T

DEFINITION

$Sense_{s-275}$ **in state** Climb		T	·
$Sense_{s-275}$ **in state** D		·	T
Own-Tracked-Alt$_{f-437}$ <	Other-Tracked-Alt$_{f-432}$	T	·
Own-Tracked-Alt$_{f-437}$ >	Other-Tracked-Alt$_{f-432}$	·	T
... ≤ ... SV		T	T

Macro

Above

Description: [illegible] has ch [illegible] D [illegible] car [illegible] RAs ar [illegible]

References: (↓) RESOLUTION [illegible]

Appears In: [illegible] Choose [illegible]

DEFINITION

there exists j

[Other [illegible] j] ▷ [illegible] **in** [illegible] **state** D	T
[illegible] ([illegible] j)	F
[Other [illegible] j] ▷ [illegible] **in state** [illegible] RA	T

Macro

BeaVg alid

References: (↓) TRACKING T RACK_INTRUDERS

Appears In: [illegible] Gitiond

DEFINITION

[illegible]

Below

Description: ... has ch... D...) Car... RAs ar... delay

References: (↓) RESOLUTION ...

Appears In: No... Car... Choose R... A-S... tr... RA-...

DEFINITION

j there exists

[Other-Aircraft j] ▷ Tr... ▷ Sense $_{s-275}$ **in state** Climb	T
Don't-Care ... (j)	F
[Other-Aircraft j] ▷ Tr... ▷ ... **in state** RA	T

Macro

Clin bD elsinh

Description: Bo d.ideGlanad D

References: (↓) RESOLUTION.tD _car _tset

Appears In: SL Aut

DEFINITION

d ibl t Climb	**Climb-in state**	T
d ibl t Dial	**Dial-in state**	T

Macro

Climb-RA-Weakened

References: (↓) Geometric ... _test

Appears In: Climb Right-Str ... Change, ... Geometric ... Climb ... Threat Detection ...

DEFINITION

PREV(Climb-Strength$_{384}$) > Climb-Strength$_{384}$	T

Macro

Coding ... as Change

References: (↓) Set_up_glb_flags (7-P33).

Appears In: Aural-Alar

DEFINITION

Get ... Climb s_{263} **in state**	T	·	T	·
... YCD ... s_{264} **in state**	·	T	·	T
Climb Rate-Alt-Str Change s_{353}	T	T	·	·
Desc- Rate-Alt-Str Change s_{355}	·	·	T	T

Macro

Desen d-RA-Weak ened

References: (↓) Geotr eipr _tst

Appears In: rsD Rgh-Str Chgn

DEFINITION

P (St) > D	T

Macro

DoCl lim b-DoD esd

References: (↓) Ceotr eipt _tst

Appears In: Ceotr Climb_nt,Cetr
elRt Ceotr

DEFINITION

Ceinpo	RA ▷ Cli-VSL **in** **one of** VSL Cli-VSL	T
Ceinpo	RA ▷ Ind-VSL **in one of** VSL	T

Macro

Extension-Alt-Check

Description:

Column ... s
... check

Column ... Climb
... si
... chec

References: (↓) RESOLUTION_AND_COORDINATION ... Extr ... _chec

Appears In: R ... Str

DEFINITION

... Sens$_{s\text{-}275}$ **in state** D	T	·
... $_{s\text{-}243}$ **in state**	T	·
Sens$_{s\text{-}275}$ **in state** Climb	·	T
... Climb$_{s\text{-}243}$ **in state**	·	T
Reversal$_{s\text{-}282}$ **in state** Reverse	·	F

Ir...eG...r h...e

Description:

Columns 1-3 ... has n 0 ... than 4. ... and i ...

Colum... ... t and o ... than 200 ft ...

Column 2 ... CA... pp

Column 3 ... No-cr ... situat

Columns 2 and 3 ... the thre ... 200 ft ...

References: (↑) S ... 2 ... (↑) S ... 2.57 and 2.57.1 (↓) RESOLUTION_AND_COORDINATION ... (↓) ... _proj_ ...

Appears In: R...-Str

DEFINITION

... R... -T	T	T	T
Tau-R...	F	F	F
... Tau- ... Ca... > 4.0 ...	T	T	T
... cr	F	F	F
... TCAS	·	T	·
... C... s	T	·	·
... in ... state N ...	·	·	T
O... -Trac ... 200 ft ...	T	·	·
... r-Trac ... ss ... 2 ...	·	T	T
... Tau- [... $\leq$ ... Conflict-SL]	·	T	T
VMD $\leq$ 200 ft ...	·	T	T

Macro

Cdlecr

References: (↓) RESOLUTION_AND_COORDINAT ON macr _de
(↓) eascr _porj _de

Appears In:

DEFINITION

Rtess al-T	F
Te-Tau- Capod $p_{f\text{-}423} >$ 4.0 $s_{MHE)}$	T

Macro

IneG lim b-Canled

Appears In: ConGl-Alt Rat

DEFINITION

PREV(-Som R(AL-$_{ns8}$ Cels,Inlncr 25b0 $_{l))}$	T
Som R(AL-$_{ns8}$ Cels,Inlncr 25b0 $_{l)}$	F

Macro

Inde-D ... esen d-Canelled

Appears In: Covid-Alt ... Rate

DEFINITION

PREV(-Sem -earl(D) ... 250 (pm))	T
Sem-earl(D) ... 250 (pm)	F

Macro

Inner-RA -Ended

Appears In: Go-or Climb, Go-or

DEFINITION

PREV(**Some RA-Str**$_{277}$ **in state** 2500 ft)	T
Some RA-Str$_{277}$ **in state** 2500 ft	F

Macro

Lev-W ait-C ondition

References: (↓) ED CT_CONFLICT.Trac _sn _tst,
(↓) oi Av _TCAS_TCAS_Crossing

Appears In: Tatr Condition

DEFINITION

R_{m-306} Alt	F
Own Own-Status$_{s\text{-}297}$ **in state** A	T
Tatr R_{m-374} ange-T	T
$_{m-7}$ esTatr Alt-T	T
$_{m-67}$ RTctual	F
TCAS-TCAS-$_{m-80}$ Crossing-T	T

Macro

Multi-Air Flag

Possible Values: {1,0}

Appears In: gsRBeut

DEFINITION

= 1

eithna RA exts

= 0

ezo RA'ste

Macro

New-Climb

Appears In: Own-Goal-Alt-Rate

DEFINITION

Composite-RA$_{s\text{-}266}$ **in state** Climb	T
PREV(Composite-RA$_{s\text{-}266}$ **in state** Climb)	F

Macro

New-Descend

Appears In: Own-Goal-Alt Rate

DEFINITION

Composite RA$_{\text{s-266}}$ **in state** D		T
PREV(Composite RA-Alt)$_{\text{s-266}}$ **in state** D		F

Macro

New-Taltr

Appears In: Cruise-Alt Rate

DEFINITION

there exists i

$_1$[Other-Aircraft$_{s\text{-}44}$ i] **in state** RA	T
PREV$_1$([Other-Aircraft$_{s\text{-}44}$ i] **not in state** RA)	T

Macro

New-Incr lim b

References: (AD _gal _rat

Appears In: Cr Cl b Ci Cl-Alt Rat

DEFINITION

Som R(AL Cl, Incr 25 pm	T
Od-Tr Rat ≤ 25 (l/	T
PREV(-Som R(AL Cl, Incr 25 pm))	F

Macro

New-L... er ... es... d

References: (...) ... _gal _rat

Appears In: ... ON ... Ot... Alt ... Rat

DEFINITION

Som-... (...) ... 25...	T
Ow... -Trac ... Rat$_{F438}$ $\geq$ - 25... / ... CDES	T
PREV(-Som ... (...) ... 25...))	F

Macro

No-Slug

References: (↓) RESOLUTION_AND_COORDINATION.No_Wave_Test.

Appears In: Rough-Str

DEFINITION

diff-Tau-Cap$_{f\text{-}40}$ max(2T [CRP-D Conflict-SL$_{f\text{-}385}$, Other-Trac ness$_{f\text{-}43}$])	T

Macro

No-W aitingle ijPr

Appears In:

DEFINITION

j, for all

Other-Aircraft[j] info already sent	T	·
Tffa)i(≥ Sc Tffa j(≥ Sc	·	T

Macro

No-Weaken

References: not (↑) S 2.28.1, (↓) RESOLUTION_AND_COORDINATION.No_Weaken_Test

Appears In: Right-Str

DEFINITION

Right-Str$_{s\text{-}277}$ **in one of** ... -5	T	T	T	F	F	F	F
Right-Str$_{s\text{-}277}$ **in state** ...	F	F	F	T	T	T	T
Reversal$_{s\text{-}282}$ **in state** Reverse...	T	F	·	·	T	F	·
DURATION(Right-Str$_{s\text{-}277}$ ...) ... 5 s	T	·	·	·	T	·	·
DURATION(Right-Str$_{s\text{-}277}$ ...) ... 10	·	T	·	·	·	T	·
...-Tracker ... ness$_{f\text{-}43}$ ... 2	·	·	T	·	·	·	·
... [...]	·	·	·	T	F	F	F

Macro

Non-Bearing ... Climb

Description:

Column 1: ... downward s... situat... upward ... and curr... 300ft ...

Column 2: ... upward s... downward s...

Column 3: ... upward s... cr... situation

References: (↑) S... 2.37 ... (↑) S... 2.49 ... (↓) RESOLUTION_AND_COORDINATION ...

Appears in: ... Alt-S

DEFINITION

... Climb ... Dsr ...	F	T	T
Own-Alt-Trac > ... Alt-Trac	T	·	·
... V ... 300ft	T	·	·
... ≥ ... thr]	T	·	·
Own-Alt-Trac ... Alt-Trac	·	T	F
... thr]	·	F	·

Macro

Non-Crossing Biased - Descend

Description:

Column 1 Downward sense advisory, ... no altitude crossing ... upward s

Column 2 Upward sense ... crossing ... situation ... no altitude crossing ... downward s ... 300 ft

Column 3 ... Downward s ... ratio

References: (↑) S 2.49 2(3)(...)
(↓) RESOLUTION_AND_COORDINATION ... S ... ses

Appearance: Alt-S

DEFINITION

... Climb ...	F	T	F
Own-Alt-Trac $_{f437}$ > Other-Alt-Trac $_{f432}$	F	·	F
... [... ALT thr]	F	·	·
Own-Alt-Trac $_{f437}$ ... Other-Alt-Trac $_{f432}$	·	T	·
... V ... 300 ft	·	T	·
... [... ALT thr]	·	T	·

Macro

Potential-Threat Condition

Description: To b ... *otential-Threat*, ... sat-
s ... ntial ... fr ... ith ... pro
t ... g ... also ... th
g ... po ... ntial ... ad
d ... tio ... for ... by ... ho ... ad
(P ... 5 ... th
d. ... al

References: (↓) TRAFFIC_ADVISORY ... (↓) Range ... _hit ...

Appears in: ...

DEFINITION

... Other-Air-Status$_{s\text{-}297}$ **in state** A	·	T
... Potential-Threat ... Range ...	T	T
... Altitude ...	F	T
... ($\geq$	F	·
... Potential-Thr ... T	·	T

Pttie l-TaRa

Description:

Column 1 di Rain than tshr-
old.

Column 2 rygR at than a small
amt, and deepg/tran dse n
dexc th tshr, and ren dxe n in-
. rental v

Columns 3 stat than e-g fakr has a
degngy s
.eaiod

References: (↓) TRAFFIC_ADSV ORY.ffra c_rn _tst

Appearsaeli Ts: Gri, nd
atRfThr Grind

DEFINITION

Oder-Trac Rage$_{f\text{-}434}$ RTHRTA[Coflict-SL$_{f\text{-}385}$]	T	F	F
Oder-Trac Rage- Rat$_{f\text{-}435}$ > RDTH RTA	·	·	F
Oder-Trac Rage$_{f\text{-}434}$ Oder-Trac Rage- Rat$_{f\text{-}435}$ ≤ THA1[Coflict-SL$_{f\text{-}385}$]	·	·	·
Oder-Trac Rage$_{f\text{-}434}$ DMODTA[Coflict-SL$_{f\text{-}385}$]	·	·	·
Oder-Trac Rage- Rat$_{f\text{-}435}$ < −RDTH RTA	·	T	F
DMODTA[Coflict-SL$_{f\text{-}385}$]^2 / Oder-Trac Rage$_{f\text{-}434}$ Oder-Trac Rage$_{f\text{-}434}$ / Oder-Trac Rage- Rat$_{f\text{-}435}$ < RTHRTA[Coflict-SL$_{f\text{-}385}$]	·	T	·
DMODTHAT[Coflict-SL$_{f\text{-}385}$]^2 / Oder-Trac Rage$_{f\text{-}434}$ − Oder-Trac Rage$_{f\text{-}434}$ / −RDTH RTA < RTHRTA[Coflict-SL$_{f\text{-}385}$]	·	·	T

Fe Ge - ocari

References: (↓) YA D _AB V OR Hi, D _Goal _Rate
(↓) Geor eiprt _test

Appears In: Aural-Alar

DEFINITION

Never-I	T	·	·	·	·	·	·
Don't- Climb-D	·	T	T	T	F	F	F
PREV(Geor) Climb state N	·	T	F	·	T	F	·
Own-Tract Rate$_{f\text{-}438}$ < +(f0/) 300 ft/	·	T	·	·	·	·	·
PREV(Geor) state N	·	·	T	T	·	T	T
Own-Tract Rate$_{f\text{-}438}$ > 500 ft/ 300 ft/	·	·	T	T	·	·	·
Own-Tract Rate$_{f\text{-}438}$ < Climb-Goal$_{f\text{-}382}$ − 300 ft/	·	·	·	·	T	·	·
Own-Tract Rate$_{f\text{-}438}$ > Goal$_{f\text{-}390}$ 300 ft/	·	·	·	·	·	T	T

Macro

Prxmi ataTfi c-C oditio

Description: T deos *n-Proximity* rstud ith a 0gm6. f Atolly, rtud e latpis alt edt ith gpritd (ftpo c fyni ly vdad egg(h th d. dval

References: (↓) TRAFFIC_ADV ORYPmi _tst.

Appears In: nCast

DEFINITION

Other-Status$_{s\text{-}297}$ **in state** A	·	T
Other-Trac Rage$_{f\text{-}834}$ 6.0 mi	T	T
-Ald RTu$_{f\text{-}293}$	F	T
Other-Trac$_{f\text{-}437}$ ≥	F	·
Other-SV 200 ft	·	T

Macro

RA-Display-Delay

Description:

Column 1 The TCAS display ... no ... has ... selected ... than ... 's ...

References: (↑) Displays (2.35), (↑) TCAS/TCAS ... (2.65.2), (↓) RESOLUTION_AND_COORDINATION.TCAS ...

Appears In: ...

DEFINITION

Equipage$_{v-286}$ TCAS-TA/RA	T
Other-V RC$_{v-289}$ = No ...	T
Own-MS-Addr$_{v-234}$ > Other-MS-Addr$_{v-292}$	T

Macro

Rev- aNl-Rio esg e

Comments: **Note** [illegible] s 6) [illegible] chan

References: (s P) _tr _pt (3-P 3).

Appears In: Oth RC, s, [illegible] Addr Oth V RC

DEFINITION

Receive [illegible] Mo [illegible] Trans	T	T	T	T
VB = VB$_{f\text{-}429}$(CVC, VRC)	T	T	T	T
CVC $\neq$ 0	T	·	·	·
CHC $\neq$ 0	·	T	·	·
VRC $\neq$ 0	·	·	T	·
HRC $\neq$ 0	·	·	·	T
CVC $\neq$ 3	T	T	T	T
VRC $\neq$ 3	T	T	T	T

Macro

Som eeTCtr leaO fC onflict

Comments: The RAdfly m
otse fcdton RA.

References: (↓) Hi _or_msis etps(186), (↓) S _up_gki _lit
(p 204)

Appears In: Combi Cbntr

DEFINITION

ngdl, fdFes

PREV([Ottera-A i] ▷ Tatr ▷ryslnAs **state** RA)	T
[Ottera-A i] ▷ Tatr ▷ryslnAs **state** RA	F

Macro

Sion eTaTr rqepd

Comments: NOfTheTh $_1$[Order-Aft$_{s\text{-}44}$ i]-dimm krtpapic

References: (P) ROCESS Dr _Trac (32).

Appears In: Combi Contr

DEFINITION

lightly, the Ours aft

PREV[Order-Aft i] ▷ Tatr ▷ Ready-state RA)	T
$_1$[Order-A$_{s\text{-}44}$ i] in Tstate N	T

Macro

Standby Condition

Comments: See ARINC 735 Attachments 3A and 3B, Note 20.

Appears In: Other Aircraft

DEFINITION

Standby$_{v\text{-}25}$ = True	T	·	·
Traffic Display$_{[illegible]}$	·	F	·
Own Status$_{v\text{-}252}$ = On Ground	·	T	·
Selected Mode$_{[illegible]}$	·	·	T

[Other-Alt- i] ↪ Su[illegible] atch es

Description:

Column 1 Th[illegible] is fro [Other-A[illegible] i] if Other- [A[illegible]] [illegible] match.

Column 2 [illegible] [Other-A[illegible] i] [illegible] and [Other-A[illegible] i] [illegible]

Appears in: [illegible] Alt [illegible] Alt [illegible] Other- [illegible]-Addr Other- Range-Val Range- [illegible]-T [illegible] Sens [illegible]

DEFINITION

RECEIVE [illegible]	T	T
$_1$[Other-A$_{s\text{-}44}$ [illegible]] **in state** Thr	T	·
$_1$[Other-A$_{s\text{-}44}$ i] ▷ [illegible]-Addr FIELD([illegible]T R)	T	·
$_1$[Other-A$_{s\text{-}44}$ [illegible]] **in state** Trac	·	T
$_1$[Other-A$_{s\text{-}44}$ i] [illegible] FIELD(SURVNO)	·	T

Macro

Tau-Rising

References: (Eq. 2.43), (ED CT_CONE CTS.Tau calculat

Appears In: Che Rag T

DEFINITION

there exists a

a > b > c	T
a > d > e	T
Prev_a(True-Tau-Up$_{\text{ref-424}}$) Prev_b(True-Tau-Cap$_{\text{ref-423}}$)	T
Prev_b(True-Tau-Up$_{\text{ref-424}}$) Prev_c(True-Tau-Cap$_{\text{ref-423}}$)	T
Prev_a(Outer-Trac Range$_{\text{ref-434}}$) > 1.5 n	T
Prev_b(Outer-Trac Range$_{\text{ref-434}}$) > 1.5 n	T
Prev_c(Outer-Trac Range$_{\text{ref-434}}$) > 1.5 n	T
Prev_d(True-Tau-Up$_{\text{ref-424}}$) Prev_e(True-Tau-Cap$_{\text{ref-423}}$)	T

Ap Sit io

Description:

Columns 1-2 TCAS equipped and tw d equippe n and ... or i and w ... field ... "el s t rat ... curr ... > MAXALT FD

Columns 3-4 TCAS ... equipp ... and ... confidenc ... l" ... M ... XALT FD ... fee

Columns 1 and 3 TCAS ... upward ... RA.

Columns 2 and 4 TCAS ... downward ... RA.

References: (↑) ... 2.36.2, (↓) ED ... CT_CONF ... CTS.Trac _firmness_test, (↓) Alt ... rat ... _test.

Appears In: ... G ... id

DEFINITION

Equipp ... TCAS	T	T	F	F
Tw ... Thr	T	T	·	·
Other-V RC ... No	T	T	·	·
Tau- Cap ... < FRTHR	T	T	T	T
Own-Trac ... Rat ... ≤ ... 600	T	T	T	T
... V ... 600	T	T	T	T
... Descr ... Climb	T	·	T	·
Own-Trac ... Alt Trac	T	·	T	·
... Descr	·	T	·	T
Own-Trac ... > ... Alt Trac	·	T	·	T

Ablations:

FRTHR

eTem [CDRAiD Coflict-SL$_{f\text{-}385}$, OFir-Trac ness$_{f\text{-}43}$]

Macro

Climb-RA-fSGag hrag

References: (↓) Gcetr eipt _tst

Appears In: HGStr Chegn

DEFINITION

Climb RaW	T	·
Climb Ral-Str	·	T

Macro

Clim b-RA-flag ed

References: (↓) Cor ept _set

Appears In: Climb Rth-Str Chgn

DEFINITION

PREV(Climb-Str$_{384}$) Climb-Str$_{384}$	T

Macro

Desen d-RA-ffSGg heag

References: (↓) Geotr eipt _tset

Appears In: HaGStr Chegn

DEFINITION

Ded-	RaeW$_{m\ 520}$	T	·
Ded-	RetAl-Str$_{m\ 550}$	·	T

Macro

Desed-R A-fing ed

References: (↓) Geotr eipt _tst

Appears In: Ised- Rgh-Str Chgn

DEFINITION

Prgthg(Str) (352)	T

Do G ar T

Parameters:

Description: ...Th ... th ... an ... oppo ... ch

References: (↑) S ... (2.5) (↓) D ...

Appears In: ... A

DEFINITION

Equipp ... TCAS	F	F	F	F
Non-Track ... i]	T	T	T	T
[Other-A ... i] ▷ Sens $_{s\text{-}275}$ **in state** Climb	T	·	T	·
[Other-A ... i] Sens $_{s\text{-}275}$ **in state** D	·	T	·	T
... D ... (S ... [... ALT thr ...]	T	·	·	·
... (n ... [... ALT thr ...]	·	T	·	·
Own-Aircraft ▷ Composite RA $_{s\text{-}266}$ **in state** D	·	·	T	·
Own-Aircraft ▷ Composite RA $_{s\text{-}266}$ **in state** Climb	·	·	·	T
Lev ... Offs Wo	·	·	T	T

Abbreviations:

LeWSQre

j, **there exists**
$_1$([OrdenaAt$_{s-44}$ j] ▷ Tahtr ▷ Sensmost **in same state as**
$_1$[OrdenaAt$_{s-44}$ i] ▷ Tahtr ▷ Sens) **and**
VMD[j] ⊲ Consecutive Sprat (Chapoi)

VMD[j]

For $_1$[OrdenaAt$_{s-44}$ j]
Let:
OWNER = Own-Al-Trac$_{f-437}$ (4$_{S_{WD(CL)}}$. Own-Al-Trac Rat$_{f-438}$)
RELALT = OWNER - Own-Al-Trac$_{f-432}$
TAUMIN = (Tau-Mi Cap$_{f-440}$),
Ter-Tau- Unproped$_{f-424}$)
TRTRU = Ter-Tau- Capped$_{f-423}$
TIME = XTIME lexity SLens]
Then:
VMD[j]$_{f-420}$ exists(NMT- RELALT, TAUMIN,
TRTRU, TIME)

Abbreviations:

SENSEMTR

[ldsh sensor-Al-TThr $_1$ ready$_{s-20}$]

TFRTHR

Term [GRPAID Conflict-SL$_{f-385}$, OFFer-Trac ness$_{f-43}$]

Macro

Low -Firmness Separation

Description:

Columns 1-3 The TCAS ... n ... thr ... and c

Columns 4-6 The TCAS ... ic ...

Columns 1 and 4 ... Separat

Columns 2 and 5 TCAS ... ch upward ... RA ... curr > LOWFIRMRZ fee

Columns 3 and 6 TCAS ... ward ... RA ... curr > LOWFIRMRZ fee

NOTE: S ... RM ... that th ... h

References: (↓) ED CT_CONL CTS.Trac _firm ness _test, (↓) Evaluat _low _firm ness _separat

Appears In: ... Cond

DEFINITION

Equip TCAS	T	T	T	F	F	F
Other-V RC$_{v-289}$ = No	T	T	T	·	·	·
Two Of Thr	T	T	T	·	·	·
Tau-Capt IT$_{f-423}$ < RTHR	F	F	F	F	F	F
(S SENS RM	T	·	·	T	·	·
(l SENS RM	T	·	·	T	·	·
Bili (l (S	·	T	F	·	T	F
Own-Alt-Trac$_{f-437}$ Other-Alt-Trac$_{f-432}$	·	T	·	·	T	·
Own-Alt-Trac$_{f-437}$ > Other-Alt-Trac$_{f-432}$	·	·	T	·	·	T
Clst V 1 ft (LO	·	T	T	·	T	T

Macro

New-Iseacr

References: (dIJMDA _gal _rat

Appears In: eRt Geotr

DEFINITION

Ne-I Climb$_{333}$ $_{m}$		T	·
Ne-I $_{m}$ $_{334}$		·	T

Macro

No-W eak e nN egat ri

Parameters: e nt-stat

References: (↓) RESOLU-TION_AND_COORDINAT ON.N o _We _Tst

Appears In:

DEFINITION

Resal$_{s\text{-}282}$ **in state** Rders	T	F	·
)< nt-stat$_{f\text{-}389}$)(Gate Ardv 5s	T	·	·
)< 10 nt-stat$_{f\text{-}389}$)(Gate Ardv	·	T	·
Other-Trac ness$_{f\text{-}43}$ 2	·	·	T

Macro

No-W eaRe oisti

Parameters: e nt-stat

References: gie (↓) S aiV Rae(2.516), (↓) RESOLUTION_AND_COORDINATION.No_We _Tst

Appears In:

DEFINITION

ni-GtrV -Ps [lsRTAlt rdAdy]	T	F	F	F
Rersal$_{s-282}$ **in state** Rdes	·	T	F	·
e< nt-s$_{f-389}$iGtTArdv 5s	·	T	·	·
e< nt-s$_{f-389}$iGtTArdv 10	·	·	·	T

Macro

[OrderAft i]
↪ NonC- areC- lesoo

Parameters: nlet

Description: Dur
...

Appears In: Sens

DEFINITION

there exists j, k : i, < j **for all**

Aloo	F
Rle	F
[OrderAft j] **in state** Thr	T
[OrderAft k] **not in state** Thr	T
[OrderAft j] **in state** *Direction*	T

Potential-Threat-Alt-

Description:

Column 1 [illegible] is small.

Column 2 [illegible] and alt

References: (↓) TRAFFIC_ADVISORY [illegible]

Appears In: [illegible] Ground

DEFINITION

| Other-Tracked-Relative-Alt | < 1200 ft	T	·
Other-Tracked-Relative-Alt-Rate ≥ -	·	F
$-\frac{\|\text{Other-Tracked-Relative-Alt}\|}{\text{Other-Tracked-Relative-Alt-Rate}_{f-436}}$ < TVTHRTATBL	·	T

Abbreviations:

TVTHRTATBL

[illegible] [illegible] Coflict-SL$_{f-385}$]

Abbreviations:

RA-Inhibit-From-Ground

there exists

[illegible]-Stat [illegible] i].Ground- [illegible] SL$_{v-299}$ 2

Macro

RA-[illegible]

Description: [illegible] 2

References: (↓) TRACK_OWN.Update [illegible]

Appears In: [illegible]

DEFINITION

[illegible] 2	T	·	·
[illegible] 2	·	T	·
[illegible]	·	·	T

Macro

Eqp lev lid-TE

Description:

Column 1 TCAS ... n ... -ind ... that last ran ... was ... d.

Note ... CAS ... and n ... and ... that thr ... CAS

References: (↓) ED CT_CONE CTS.Trac _fm ns _st

Appears In: ... ai- ... thr ... nd

DEFINITION

Equipp TCAS	T
Other-V RC_{v-289} = No	T
Two Of $_{m-376}$ Thr	F

Macro

Test-VSL-Rate

Parameters: Rate

Description:

ALIM OD = ALIM [illegible]

[illegible] Rate

VSL [illegible]

Appears In: R[illegible]-Str

DEFINITION

VSL-OK$_{\text{m-378}}$(Rate, [illegible]-Tau- Cap[illegible]$_{\text{f-423}}$, ALIM OD, [illegible])	T
VSL-OK$_{\text{m-378}}$(R[illegible]-Tau- Cap[illegible]$_{\text{-40}}$ OD, [illegible])	T

Abbreviations:

ALIMOD

$$\begin{cases} -75 & \textbf{if } (\text{Non-Trac} \ldots \text{Rate} > 0 \textbf{ or} \\ & \quad \text{Non-Trac} \ldots \text{Old-Rate}_{f\text{-}403} > \text{Rate}) \\ 0 & \textbf{Otherwise} \end{cases}$$

LOW-ERROR NESS

Min(Tau-... Cap ... $_{f\text{-}40}$ <
Term ... Climb ... (S D
... Old-Err-Trac ... ness$_{f\text{-}43}$)

ZDI

$$\begin{cases} \text{Max}(\text{Old-Alt-Trac} \ldots \text{Rate}_{f\text{-}404}, \\ \quad \text{Old-Alt-Trac} \ldots \text{Rate Out}_{f\text{-}405}) \\ \quad \textbf{if } \text{Non-Trac}_{\ldots} \textbf{ and} \\ \quad\quad \text{LOW-} \ldots \text{RMS} \textbf{ and} \\ \quad\quad \text{Sens}_{s\text{-}275} \textbf{ in state } \text{Climb} \\ \text{Min}(\text{Old-Alt-Trac} \ldots \text{Rate}_{f\text{-}404}, \\ \quad \text{Old-Alt-Trac} \ldots \text{Rate Out}_{f\text{-}405}) \\ \quad \textbf{if } \text{Non-Trac}_{\ldots} \textbf{ and} \\ \quad\quad \text{LOW-} \ldots \text{RMS} \textbf{ and} \\ \quad\quad \text{Sens}_{s\text{-}275} \textbf{ in state } \text{D} \\ \text{Old-Alt-Trac} \ldots \text{Rate}_{f\text{-}433} \\ \quad \textbf{if } \text{Non-Trac}_{\ldots} \textbf{ or} \\ \quad\quad \textbf{not } \text{LOW-} \ldots \text{RMS} \end{cases}$$

Macro

TCASF CASG...

Description:

Columns 1-2 ... C ... d n ... b ... r ... tw ... pts ... l c ... aircraft ... craft ... r ... NSE ... MI

Column ... ... r ... thr ... CPA.

Column 2 ... urr ... j ... t ... CPA.

References: (↓) ED CT_CONF CTS.Trac _firmness_test, (b) Av _TCAS_TCAS...

Appears In: ... Thr ... G...d

DEFINITION

Equip... TCAS	T	T
Other-V RC$_{v\text{-}289}$ = No...	T	T
Two ... Thr	T	T
...Tau- Cap...$_{f\text{-}423}$ < ...RTHR	T	T
O...-Trac ... Rat$_{f\text{-}438}$ ≤ ...ft/...$_{LE}$	T	T
Other-Trac ... Rat$_{f\text{-}436}$ > ...ft/...$_{LE}$	T	T
...-V ... 300ft...$_{LE}$	T	T
O...-Trac$_{f\text{-}437}$ > ...-Trac$_{f\text{-}432}$	T	·
...t$_{f\text{-}406}$ ...$_{f\text{-}430}$	T	·
O...-Trac$_{f\text{-}437}$ ...-Trac$_{f\text{-}432}$	·	T
...t$_{f\text{-}406}$ > ...$_{f\text{-}430}$	·	T

Abbreviations:

TFRTHR

...Tm [...D Coflict-SL$_{f\text{-}385}$, ...r-Trac ...ness$_{f\text{-}43}$]

Macro

Threat-Alt-VSL-Test

References: (↑) Range-Test (2.30), Other-Tracked-Alt (↑) S 2.32,
(↓) RED CT_CONE CTS-Alt _test

Appears In: Threat-Alt-VSL-Test, Other-Thr... Good

DEFINITION

Adjusted-Range-Tau $_{f-293}$	T	T	T	T	T	T
| Other-Tracked-Relative-Alt $_{f-224}$ | < ZT	T	F	F	T	F	F
Other-Tracked-Range-Rate $_{f-435}$ > 0	F	F	F	T	T	T
| Other-Tracked-Relative-Alt $_{f-224}$ | < ZT	·	·	·	T	T	F
|VMD| < ZT	T	T	F	·	·	·
Other-Tracked-Relative-Alt-Rate $_{f-436}$ ≥ 1ft/s (ZD RA)	·	F	F	·	F	F
Time-To-Co-Alt $_{f-4}$ < TVTUTB	·	T	T	·	T	T
| Range-At-Co-Alt $_{f-4}$ | < DMODTB	·	·	T	·	·	T
Time-To-Co-Alt $_{f-4}$ < True-Tau-Capped $_{f-423}$	·	·	T	·	·	T

Abbreviations:

VMD

Let:

RZ = Od-n-Alt-frac$_{f-437}$ − Od-t-Alt-frac$_{f-432}$

RZD = Od-n-Alt-frac Rat$_{f-438}$ − Od-t-Alt-frac Rat$_{f-433}$

TRTRU = Tel-Tau- Cap$_{f-423}$

TAUR = del-Tau- Cap$_{f-40}$

TVP CMD = XTVP CTBL-Xi[Ens]

Then:

VMD$_{f-426}$ exists(MDI- RZ, RZD, TRTRU, TAUR, TVP CMD)

ZT

1 r[Alt-Thr-Alt-Thr]

TVTUTBL

e-Thr-T Gd-Alt-Thr U[Coflict-SL$_{f-385}$]

DMODTBL

i-Mt-Thr [Rsale-Thr Coflict-SL$_{f-385}$]

Macro

Target-Alt-Condition

Description:

Columns 1-2 Not ... a ... data ...

Column 1 The TCAS ... and the TCAS ... cr ... form ...

Column 2 The TCAS ... cr ... th ... s

References: (t) ... 2.29,2.30), (↓) ... CT_CON ... CT ... _or_ ... _test

Appears In: ...

DEFINITION

Alt ...	F	F
... Status$_{s\text{-}297}$ **in state** A	T	T
Target-Range ...	T	T
... Alt-T	T	T
...	F	F
TCAS-TCAS ...	F	·
DURATION(... $_{m\text{-}328}$) > ... 3 s	F	T
... S	F	F
... ow-F... s-S	F	F

Macro

Threat-Range-Test

References: (a) Project S 2.32, (b) DETECT_CONFLICTS.Range_test

Appears In: Crossing-Climb-Crossing-Condition, Range-Alt-Str, Other-Reversal-Tau-Cond

DEFINITION

Other-Tracked-Range-Rate$_{f\text{-}435}$ (ft/s) >	T	F	F	F
Other-Tracked-Range$_{f\text{-}434}$ > DMOD	F	·	F	F
Modified-Tau-Capped$_{f\text{-}40}$ < TRTHR	·	T	F	·
Other-Tracked-Range$_{f\text{-}434}$ ≤ 2.0	·	T	·	F
|Other-Tracked-Range-Rate$_{f\text{-}435}$ * Other-Tracked-Range$_{f\text{-}434}$| > H1	F	·	F	F
Tau-Rising$_{f\text{-}330}$	F	F	F	F

Abbreviations:

DMOD

[Range-Threshold ... Conflict-SL$_{f\text{-}385}$]

TRTHR

[... Tau-Threshold ... Conflict-SL$_{f\text{-}385}$]

H1

[... Conflict-SL$_{f\text{-}385}$]

Macro

TNrVy SSEL

References: gicel (1t) S aciaX Rat(2.516), (↓) RESOLUTION_AND_COORDINAT ONeS _Aysy

Appears In: Rgh-Str

DEFINITION

Ode-Trac RAlt Rat $_{f\text{-}436}$ 1600/ $_{(LEV}$	T	T	T	F	F
Ow-ATrac $_{f\text{-}437}$ ≥ Oe-ArTrac $_{f\text{-}432}$ -Pro [stdAEAT thr 1 rgAdy]	T	F	F	·	·
OdeAlt $_{f\text{-}406}$ ≥ -ODPlt $_{f\text{-}430}$ -Pro [stdAEAT thr 1 rgAdy]	·	T	·	T	·
Ow-ATrac Rat $_{f\text{-}438}$ > 600/ $_{(OE}$	·	·	T	·	T

Macro

Two-O f-Thr

References: (↑) TCAS/TCAS Con(ta 2.63),
(↓) TRACKING-Val _rpto _tst

Appears In: Alt-S sti-ps-Test-S
st-RTpl CAS-TCAS-Gts

DEFINITION

, **there exists** j 2} :

$\textsc{Prev}_j$(Other- RTg-Val$_{v296}$)	T
Other- RTg-Val$_{v296}$	T

Macro

VayGdit hkce

Parameters: Radar-Alt

References: (↓) RadatiyLedbi _tset

Appears In: Gwnd-L

DEFINITION

enMltiy idinadar-Alt OPS

Macro

VS6-K

Parameters: VSL-RATE, TAU, ALIM OD, OTHR-RATE

References: gie(t) S aiat Rae(2.516).

Appears In: Tset-VSL- Rate

DEFINITION

VWD eitis P [ksh-Hichr Alt 1 redAly] + ALIM OD	T

Abbreviations:

TRTLIM

M(TA U, XTHR $)_1$) exists VSL

DIRECTION

$$\begin{cases} 1 & \textbf{if } \text{Sens}_{s\text{-}276} \textbf{ in state } \text{Climb} \\ -1 & \textbf{Otherwise} \end{cases}$$

RATE

$$\begin{cases} -\text{VSL-RATE} & \textbf{if } \text{Sens}_{s\text{-}276} \textbf{ in state } \text{Climb} \\ \text{VSL-RATE} & \textbf{Otherwise} \end{cases}$$

DELAY

$$\begin{cases} \text{Max}(-2, 5\,s_{\ldots(V1)} - \\ \quad \text{DURATION}(\text{Rel-Alt-Str}_{s\text{-}277}\text{ unchan}) \\ \quad \textbf{if } \text{Own-Alt-Trac Rate}_{f\text{-}438} < \text{RATE } \textbf{and} \\ \qquad \text{Low-Trac} \\ 5\,s_{TV(1)} \\ \quad \textbf{if } \text{Own-Alt-Trac Rate}_{f\text{-}438} < \text{RATE } \textbf{and} \\ \qquad \text{New-Trac} \end{cases}$$

V0

proj-Alt(TRTLIM, Own-Alt-Trac$_{f\text{-}437}$, Own-Alt-Trac-Rate$_{f\text{-}438}$, RATE, DELAY, DIRECTION)

VMD

$$\begin{cases} \text{Other-tile-Rel-Alt} \\ (\text{RATE} - \text{OTHER-RATE}) \cdot \text{TRTLIM} \\ \quad \textbf{if } \text{Own-Alt-Trac Rate}_{f\text{-}438} \geq \text{RATE} \\ (\text{V0} - \text{Own-Alt-Trac}_{f\text{-}432}) - \\ \text{OTHER-RATE} \cdot \text{TRTLIM} \\ \quad \textbf{Otherwise} \end{cases}$$

Macro

Teshlh ovenA d-Belw

References: nsiuitAMlt(S) 2.58)

Appears In: Rgh-Str

DEFINITION

Abo	T
Ble	T

Macro

Sm e-RA-Is

Parameters: tgSstr

Description: th as Ah l

Appears In: -CLVSL, D Climb Gdanc
nedDacr -caldN Climb
nedNcr

DEFINITION

there exists

[Sens$_{s\text{-}275}$ i] **in state**	*ense*		T
RA-Str$_{s\text{-}277}$ **in state**	*trength*		T

Function

Climb-Goal

Possible Values: (1..)4000

Description: **Reference:** Own-Logic D ... _goal_ ... (... 7-P29).

Appears In: Current ... Climb ... Current

DEFINITION

Climb-Goal

1000 ft/...	**if** Current-Composite-RA$_{s\text{-}266}$ **in state** No-RA **or** Current-Composite-RA **in state** No-Climb-VSL
$Climb-Goal(1)$	**if** Current-Composite-RA **in state** Climb-VSL ▷ VSL2000
$Climb-Goal(2)$	**if** Current-Composite-RA **in state** Climb-VSL ▷ VSL1000
$Climb-Goal(3)$	**if** Current-Composite-RA **in state** Climb-VSL ▷ VSL500
$Climb-Goal(4)$	**if** Current-Composite-RA **in state** Climb-VSL ▷ VSL0
Own-Tracked-Alt-Rate$_{s\text{-}2}$	**if** Current-Composite-RA$_{s\text{-}266}$ **in state** Climb

Climb-Spatio

Possible Values: ngerl

Appears In: nipHSest

mt-Rapta-TVS

DEFINITION

n n(ClipaMbt   OBC -IB Max(C, lego, C3 d) - L
OEC -IE Max(C, lego, C3 d)) - L

Function

Climb-Strength

Possible Values: 0..8

Description: (IN)VAL (p 386-)

Appears In: Climb-Rate-Strength, Climb-Rate-AW, ... Composite-Rate-TV, ... Composite-Rate-TV

DEFINITION

Climb-Strength

8	**if** Composite	RA **in state** Climb				
4	**if** Composite	RA **in state** N	▷	Climb-VSL	▷	VSL
3	**if** Composite	RA **in state** N	▷	Climb-VSL	▷	VSL 0.5
2	**if** Composite	RA **in state** N	▷	Climb-VSL	▷	VSL 1000
1	**if** Composite	RA **in state** N	▷	Climb-VSL	▷	VSL2000
0	**Otherwise**					

Conflict-SL

Possible Values:

References: (†) S 2.33

Appears In: [illegible]

[illegible] Rating, T

CAS-T, CAS-[illegible]

[illegible] Rating, T

DEFINITION

Let:

[illegible]-Value

$$\begin{cases} 1 & \textbf{if } [\ldots]_{s\text{-}238} \textbf{ in } \quad \textbf{state } 1 \\ 2 & \textbf{if } [\ldots]_{s\text{-}238} \textbf{ in } \quad \textbf{state } 2 \\ 3 & \textbf{if } [\ldots]_{s\text{-}238} \textbf{ in } \quad \textbf{state } 3 \\ 4 & \textbf{if } [\ldots]_{s\text{-}238} \textbf{ in } \quad \textbf{state } 4 \\ 5 & \textbf{if } [\ldots]_{s\text{-}238} \textbf{ in } \quad \textbf{state } 5 \\ 6 & \textbf{if } [\ldots]_{s\text{-}238} \textbf{ in } \quad \textbf{state } 6 \\ 7 & \textbf{if } [\ldots]_{s\text{-}238} \textbf{ in } \quad \textbf{state } 7 \end{cases}$$

Then:

Conflict-SL) [illegible] (S

Function

Current-Ad is Ty ime

Parameters: current-state

Possible Values:

Appears in:

DEFINITION

ION(*current-state*)

Current-Vertical-Separation

Possible Values:

Appears In: [illegible] Alt-S [illegible] -S [illegible] Climb, [illegible] c-G [illegible] nd [illegible] -T CAS-TCAS- [illegible] [illegible] Co-Alt

DEFINITION

| Own-Alt-Track [illegible] − Other-Alt-Track [illegible] |

CVC

Parameters: [illegible]

Possible Values: { 0, 1, 2 }

Description: [illegible] Th[illegible] CVC [illegible]
[illegible] valu[illegible]

Valu[illegible]	[illegible]
0	[illegible]
1	C[illegible]
2	C[illegible]
3	[illegible]g ass[illegible]

Explanation of value selection criteria: If Oth[illegible] [illegible]aft [illegible] th[illegible]at own TCAS has [illegible] R[illegible] th[illegible] an[illegible] n[illegible] Thus, th[illegible] CVC [illegible] valu[illegible] [illegible] If TCAS had [illegible] D[illegible] RA [illegible] cycl[illegible] [illegible] [illegible] RA, th[illegible] CVC is 1 [illegible] [illegible] CAS had [illegible] Climb RA [illegible] th[illegible] [illegible] RA, th[illegible] CVC is 2 [illegible]

[illegible] S[illegible] [illegible] 7).

Appears In: [illegible]

DEFINITION

CVC =

0 **if** [Other-Aircraft$_{s\text{-}44}$ [illegible] i] **in state** N[illegible] R[illegible]

1 **if** ([Other-Aircraft$_{s\text{-}44}$ [illegible] i] **not in state** Thr[illegible] **or**
[Other-Aircraft$_{s\text{-}44}$ [illegible] i] **in state** R[illegible]) **and**
PREV([Other-Aircraft$_{s\text{-}44}$ [illegible] i]) **in state** Thr[illegible] ▷ [illegible])

2 **if** ([Other-Aircraft$_{s\text{-}44}$ [illegible] i] **not in state** Thr[illegible] **or**
[Other-Aircraft$_{s\text{-}44}$ [illegible] i] **in state** R[illegible]) **and**
PREV([Other-Aircraft$_{s\text{-}44}$ [illegible] i]) **in state** Thr[illegible] ▷ Climb

Descend-Goal

Possible Values: (−∞..+10000)

References: (out) D... _goal_ ... (... 7-P29).

Appears In: ... TRD ... , ... Goal ...

DEFINITION

Descend-Goal =

[illegible]/1000	**if** Composite RA$_{s\text{-}266}$ **in state** N... RA **or** Composite RA ... DVSL **in state** N...	
Descend − *Goal*(1)	**if** Composite RA ... VSL **in state** D...	▷ VSL2000
Descend − *Goal*(2)	**if** Composite RA ... VSL **in state** D...	▷ VSL1000
Descend − *Goal*(3)	**if** Composite RA ... VSL **in state** D...	▷ VSL500
Descend − *Goal*(4)	**if** Composite RA ... VSL **in state** D...	▷ VSL0
Own-Goal-Alt-Rate$_{[illegible]}$	**if** Composite RA$_{s\text{-}266}$ **in state** D...	

Function

Des[illegible] d-[illegible]tio

Possible Values: [illegible]

Appears In: [illegible]

[illegible]

DEFINITION

[illegible]

[illegible] [illegible]

[illegible] [illegible]

Descend-fishing

Possible Values: 18..

References: (↓) EVAL (p) 386-

Appears In: Descend-fishing-Str Rule AW

nig, HS-t

rat-Cat-Part-TV-S,

si-Cat-Part-TSV

DEFINITION

Descend-Str

$$
\begin{cases}
8 & \textbf{if} \text{ Cetinpo } \text{RIA}_{s\text{-}266} \textbf{ in state } D \\
4 & \textbf{if} \text{ Cetinpo } \text{RA}_{s\text{-}266} \textbf{ in state } N \quad \triangleright \text{ Descend-VSL} \quad \triangleright \text{ VSL} \\
3 & \textbf{if} \text{ Cetinpo } \text{RA}_{s\text{-}266} \textbf{ in state } N \quad \triangleright \text{ Descend-VSL} \quad \triangleright \text{ VSL15} \\
2 & \textbf{if} \text{ Cetinpo } \text{RA}_{s\text{-}266} \textbf{ in state } N \quad \triangleright \text{ Descend-VSL} \quad \triangleright \text{ VSL1000} \\
1 & \textbf{if} \text{ Cetinpo } \text{RA}_{s\text{-}266} \textbf{ in state } N \quad \triangleright \text{ Descend-VSL} \quad \triangleright \text{ VSL2000} \\
0 & \textbf{Otherwise}
\end{cases}
$$

Function

DownStepFunction

Parameters: MyType

Possible Values:

Appears In: Climb-Reversal-... Reversal-
Dnt-... TL ... Test-S
... Nocr ... Nb
... S-St ... Choi

DEFINITION

Up-Sat ... (D

Funoit n

Grou nd Level

Possible Values:

References: (↓) TRACK_OWN_Ground_Level_Point

Appears In: t, Altitude-Alert, t, ILS, Glideslope, h, GPS, Glide, ILS Other Status

DEFINITION

Cmd-L

$$
\begin{cases}
\text{Ord-A-Trac}_{1\text{-}437} & \textbf{if } \text{Own-Status}_{v\text{-}252} \ \text{Om Gnd} \\
\neg 1000000 & \textbf{if } \text{Own-Status}_{v\text{-}252} = \text{Om Gnd} \ \textbf{and} \\
 & 10.\ \textbf{for all} : \\
 & (\text{PREV}_j(\text{rt-Stat-bits}_{v\text{-}248}) \ \text{Nt-Val} \\
 & \textbf{and} \\
 & \text{tyVal} \ \text{Chk}_{377}(\ ) \text{ Bad-bit} \\
\text{PREV}(\text{Cmd-L}) & \textbf{if } \text{Own-Status}_{v\text{-}252} = \text{Om Gnd} \ \textbf{and} \\
 & 10.\ \textbf{there exists} : \\
 & ((\text{PREV}_j(\text{rt-Stat-bits}_{v\text{-}248}) \ \text{Nt-Val} \\
 & \textbf{or } \text{tyVal} \ \text{Chk}_{377}(\ ) \text{ Bad-bit} \\
 & \textbf{and } \text{PREV}(\text{rt-Stat-bits}_{v\text{-}248}) \ \text{Val} \\
\text{Ord-A-Trac}_{1\text{-}437} - & \\
\text{Bad-bit} & \textbf{if } \text{Own-Status}_{v\text{-}252} = \text{Om Gnd} \ \textbf{and} \\
 & \text{PREV}(\text{Cmd-L}) \ \neg 1000000 \ \textbf{and} \\
 & \text{Bad-bit} \ \textbf{and} \\
 & = \text{rt-Stat-bits}_{v\text{-}248} \ \text{Val} \\
\text{PREV}(\text{Cmd-L}) & \textbf{if } \text{Own-Status}_{v\text{-}252} = \text{Om Gnd} \ \textbf{and} \\
 & \text{PREV}(\text{Cmd-L}) \ \neg 1000000 \ \textbf{and} \\
 & \text{Bad-bit} \ \textbf{and} \\
 & = \text{rt-Stat-bits}_{v\text{-}248} \ \text{Val} \\
1000000 & \textbf{if } \text{Own-Status}_{v\text{-}252} = \text{Om Gnd} \ \textbf{and} \\
 & \text{PREV}(\text{Cmd-L}) = \neg 1000000 \ \textbf{and} \\
 & \text{Bad-bit} \ \textbf{and} \\
 & = \text{rt-Stat-bits}_{v\text{-}248} \ \text{Val} \\
\text{Ord-A-Trac}_{1\text{-}437} - & \\
\text{Bad-bit} & \textbf{if } \text{Own-Status}_{v\text{-}252} = \text{Om Gnd} \ \textbf{and} \\
 & \text{PREV}(\text{Cmd-L}) = \neg 1000000 \ \textbf{and} \\
 & \text{Bad-bit} \ \textbf{and} \\
 & = \text{rt-Stat-bits}_{v\text{-}248} \ \text{Val}
\end{cases}
$$

Funoit n

Fiith- iaCd- lim b

Parameters: ty

Possible Values:

Description: (↓)ED CT.Altrispat _tst

Appears In: lnvaismitpst-S Climb
colgsc r

DEFINITION

dRIlhi Climb =

+muipS(n	100 rss)(nozc	**if**
		dt ilbi $C_{s\,243}$ **in state**
		and type mal
muipS(n		**if**
		dt ilb- $C_{s\,243}$ **in state** N
		and type mal
+)UfpS(l	100 rss)(nozc	**if**
		dt ilbi $C_{s\,243}$ **in state**
		and type lw-fir
)UfpS(l		**if**
		dt ilb- $C_{s\,243}$ **in state** N
		and type lw-fir

Low-Firm-Separation

Parameters:

Possible Values:

Description: VRAC... cl
VRA... dv
TRTLM = t... tau l
TAU... = Mo
...TYP = i
ZPOWN = ...
ZPT... rud
ZCLM1 ... cl
ZCLM2 = C... tau and
climb
ZDS = ... d
ZDS 2 ... d
sed
ZINT C1 ... climb cl
ZINT C2 ... and
climb
ZINTD ... d
sed
ZINTD 2 = C... and
desc
ZM... P =
...

Appears In: Separat...

DEFINITION

Let:

TRTLM ... Mi ... Cap... $_{123}$, XTYP ...) ...s

TAU... = M
... Up... $_{402}$,
XTYP ... $_{1}$ ...]))

REVERSED =

$$\begin{cases} \text{True} & \textbf{if } \text{Reversal}_{s\text{-}282} \textbf{ in state } \text{Reversal} \textbf{ or } \text{Priority} \\ \text{False} & \textbf{Otherwise} \end{cases}$$

TV =

$$\begin{cases} 2.5 s_{(} & \textbf{if } \text{REVERSED} = \text{True} \\ 5_{TV} s_{(} & \textbf{Otherwise} \end{cases}$$

VRA =

$$\begin{cases}
\text{Do not-D} & \textbf{if } \text{Direction Up} \textbf{ and} \\
& [\ \text{REVERSED} = \text{False} \textbf{ and} \\
& (\ \text{Climb}_{s\text{-}248} \textbf{ in state } \ldots \\
& \text{Own-Str}_{s\text{-}352} \ldots \\
& (\ \text{Cross}_{s\text{-}280} \textbf{ not in state } \text{Own-Cross} \textbf{ or} \\
& \text{Priority} \ldots)\ ] \\
\text{Climb} & \textbf{if } \text{Direction Up} \textbf{ and} \\
& [\ \text{REVERSED} = \text{True} \textbf{ and} \\
& \textbf{not} (\ \text{Climb}_{s\text{-}248} \textbf{ in state } \ldots \\
& \text{Own-Str}_{s\text{-}352} \ldots \\
& (\ \text{Cross}_{s\text{-}280} \textbf{ not in state } \text{Own-Cross} \textbf{ or} \\
& \text{Priority} \ldots)\ ] \\
\text{Do not- Climb} & \textbf{if } \text{Direction} \ldots \textbf{ and} \\
& [\ \text{REVERSED} = \text{False} \textbf{ and} \\
& (\ \text{Climb-Str}_{s\text{-}384} > \\
& \text{Cross}_{s\text{-}280} \textbf{ not in state } \text{Own-Cross})\ ] \\
\text{Descend} & \textbf{if } \text{Direction} \ldots \textbf{ and} \\
& \textbf{not} [\ \text{REVERSED} = \text{False} \textbf{ and} \\
& (\ \text{Climb-Str}_{s\text{-}384} > \\
& \text{Cross}_{s\text{-}280} \textbf{ not in state } \text{Own-Cross})\ ]
\end{cases}$$

DELAY =

$$\begin{cases}
2.5 s_{(} & \text{Priority} \ldots = \ldots \textbf{ and} \\
& (\text{REVERSED} = \text{True} \textbf{ or} \\
& (\ \text{Cross}_{s\text{-}280} \ \text{Own-Cross} \textbf{ and} \\
& (\ \text{Climb-Str}_{s\text{-}384} > \ \ldots \textbf{ or } \text{Own-Str}_{s\text{-}352})) \\
5_{TV} s_{(} & \textbf{Otherwise}
\end{cases}$$

ZD =

$$\begin{cases}
\text{Max}(\ \text{Own-Alt-Track Rate}_{i\text{-}404}, \\
\quad \text{Own-Alt-Track Rate Out}_{f\text{-}405}) & \textbf{if } \text{D} \ \text{Up} \\
\text{Min}(\ \text{Own-Alt-Track Rate}_{i\text{-}404}, \\
\quad \text{Own-Alt-Track Rate Out}_{f\text{-}405}) & \textbf{if } \text{D} \ \text{Down}
\end{cases}$$

ZOWN... (Ode-AT, frac Ode-AT, frac Rate
VRA, ...T RTZ)M OWN... (Ode-AT, frac Own-
Tde-Adt- Rate
VRA, ...TA ULI)M
ci... (... frac
N... T RTI)M
ZNT ... (... frac
...TA ULI)M

Th...:

...

OWN2 OWN2, ...1 2) ... D Up
OWN2 OWN2, ...T 2) ... D

Function

Lowest-Ground

Possible Values: [illegible]

Description: [illegible] Th [illegible] [illegible]
[illegible] stat

Appears In: [illegible]

DEFINITION

[illegible] Stat [illegible] i] ▷ Ground- Con[illegible] $_{\text{v-299}}$)

Function

Modified-Tau-Capped

Possible Values: Real

Description:

MOPS BEEf CT_CONL CTS.Tau is calculat

Appears In: Dnt-Modified-Tau- Capped-SN
Test-VSL-Rate-Alert-T Range-TV

DEFINITION

Modified-Tau-Capped$_p$ = Modified-Tau-Uncapped$_{p_{402}}$

Modified-Tau-Capped$_p$ =

$$\begin{cases} \text{Modified-Tau-Uncapped}_{p_{402}} & \textbf{if } \text{Other-Track-Range-Rate}(\ldots) > \\ \text{Max}(\text{Modified-Tau-Uncapped}_{p_{402}} & \\ \quad \text{PREV}(\text{Modified-Tau-Capped}_p)_{p_{40}} & \textbf{Otherwise} \end{cases}$$

MTTaudedi- uU- mapepd

Possible Values: Real

References: (↑) Rag T (2.30.2) iot (↑) S 2.30.3 f (tau 0 i CPA ued (↓) D CT_CONE CTS.Tau calculat

Appears In: the MT, Cap ref-Cont-part-TS, si-pl-Rat-as-TV

DEFINITION

$$\text{Let: TAU} = \frac{\text{Cd-te-Trac Rag}_{f-434} - \frac{\text{DMODTBL}^2}{\text{Cd-te-Trac Rag}_{f-434}}}{\text{Cd-te-Trac Rag Rat}_{f-435}}$$

Then:

deMTau- Unpea

$$\begin{cases} 0_{\text{MIN,TAU}} & \textbf{if } \text{Cd-te-Trac Rag}_{f-434} \leq \frac{\text{DMODTBL}^2}{\text{Cd-te-Trac Rag}_{f-434}} \\ -\text{TAU} & \textbf{Otherwise} \end{cases}$$

Abbreviations:

DMODTBL

imMr	[Rsalge-Thr Coflict-SL$_{f-385}$]

Old-Rate

Possible Values:

Appears In: Set-VSL-Rate

DEFINITION

$$\begin{cases} 2000 & \textbf{if } \text{Rate-Str}_{277} \textbf{ in state } \text{VSL-} \quad 2000 \\ 1000 & \textbf{if } \text{Rate-Str}_{277} \textbf{ in state } \text{VSL-} \\ 500 & \textbf{if } \text{Rate-Str}_{277} \textbf{ in state } \text{VSL-5} \\ 0 & \textbf{if } \text{Rate-Str}_{277} \textbf{ in state } \text{VSL-} \end{cases}$$

Function

Other-Fuel-Alt-Rate- er

Possible Values:

Description: Return the value ... CAS SARPS ... OPS ...

Appears In: ... T ... Rate

DEFINITION

Definition.

Other-Fixed-Alt-Rate-Route

Possible Values:

Description: Return the value … CAS SARPS … OUTR … OPS …

Appears In: …

DEFINITION

Own-Projected-Alt

Possible Values:

Appears In: TCAS-TCAS(Get-Entry-VSL-T

DEFINITION

Own-Tracked-Alt$_{f437}$ (Own-Tracked-Alt-Rate$_{f438}$ Tau-True-Capped$_{f423}$).

Proj-Vertical

Parameters: TPROJ, Z, ZD, VRA, DIRECTION, REVERSED

Possible Values:

Description:

TPROJ = Projection time
Z = altitude
ZD = altitude rate
VRA = vertical resolution advisory
DIRECTION = pi
ZDMIN = min altitude rate
ZDMAX = max altitude rate
ZDGOAL = goal altitude rate
TAR =
ACCEL =
TUGR =
TESC =
TACC =
TAGR =
ZDACC =

Appears In:

DEFINITION

Let:

ACCEL =

$$\begin{cases} 11.2 \text{ ft/s}^2{}_{(\text{RACCEL})} & \textbf{if } \text{REVERSED} = \text{True} \\ 8 \text{ ft/s}^2{}_{(\text{VACCEL})} & \textbf{Otherwise} \end{cases}$$

DIRECTION =

$$\begin{cases} 1 & \textbf{if } \text{VRA} = \text{Climb } \textbf{or } \text{VRA} = \text{Don't-Descend} \\ -1 & \textbf{if } \text{VRA} = \text{Descend } \textbf{or } \text{Don't-Climb} \end{cases}$$

Index =

$$\begin{cases} 0 & \textbf{if } \text{VRA} = \text{No-Adv} \\ 1 & \textbf{if } \text{VRA} = \text{Climb } \textbf{or } \text{VRA} = \text{Don't-Descend} \\ 2 & \textbf{if } \text{VRA} = \text{Descend } \textbf{or } \text{VRA} = \text{Don't-Climb} \end{cases}$$

ZDMIN = Min-Alt-Rate(Index)

ZDMAX = Max-Alt-Rate(Index)

ZDGOAL =

$$\begin{cases} ZDMIN & \textbf{if } (\text{ZD} < \text{ZDMIN}) \textbf{ and } (\text{VRA} = \text{Climb } \textbf{or } \text{VRA} = \text{Don't-Descend}) \\ ZDMAX & \textbf{if } (\text{ZD} > \text{ZDMAX}) \textbf{ and } (\text{VRA} = \text{Descend } \textbf{or } \text{VRA} = \text{Don't-Climb}) \\ 0 & \textbf{Otherwise} \end{cases}$$

GOALTEST =

$$\begin{cases} \text{True} & \textbf{if } \text{ZDGOAL} \neq 0 \\ \text{False} & \textbf{Otherwise} \end{cases}$$

$$\text{TUGR} = \frac{\text{ABS(ZDGOAL} - \text{ZD)}}{\text{ACCEL}}$$

TESC = TPROJ - DELAY

TA1R =

$$\begin{cases} \text{DELAY} & \textbf{if } \text{GOALTEST} = \text{True} \\ \text{TPROJ} & \textbf{Otherwise} \end{cases}$$

TACC =

$$\begin{cases} 0 & \textbf{if } \text{GOALTEST} = \text{False} \\ \text{TESC} & \textbf{if } \text{GOALTEST} = \text{True } \textbf{and } \text{TUGR} > \text{TESC} \\ \text{TUGR} & \textbf{if } \text{GOALTEST} = \text{True } \textbf{and } \text{TUGR} \leq \text{TESC} \end{cases}$$

TAGR =

$$\begin{cases} 0 & \textbf{if } \text{GOALTEST} = \text{True } \textbf{and } \text{TUGR} > \text{TESC} \\ \text{TESC} - \text{TUGR} & \textbf{if } \text{GOALTEST} = \text{True } \textbf{and } \text{TUGR} \leq \text{TESC} \end{cases}$$

ZDACC =

$$\begin{cases} \text{ZD} + \text{DIRECTION} \cdot \text{ACCEL} \cdot \text{TACC} & \textbf{if } \text{GOALTEST} = \text{True } \textbf{and } \text{TUGR} > \text{TESC} \\ \frac{\text{ZDGOAL}}{2} & \textbf{if } \text{GOALTEST} = \text{True } \textbf{and } \text{TUGR} \leq \text{TESC} \end{cases}$$

Then:

ciarlett-jV

$$
\begin{cases}
\mathrm{ZD} = \mathrm{Z} \cdot \mathrm{TAR} & \textbf{if} \ \mathrm{TACC} \leq 0 \\
\mathrm{ZD} = \mathrm{Z} \cdot \mathrm{TAR} + \mathrm{DZACC} \cdot \mathrm{TACC} & \textbf{if} \ \mathrm{TACC} > 0 \ \textbf{and} \\
 & \quad (\mathrm{TCAR} \leq 0) \\
\mathrm{ZD} = \mathrm{Z} \cdot \mathrm{TAR} + \mathrm{ADACC} \cdot \mathrm{TACC} + \mathrm{DZOACG} \cdot \mathrm{TAR} & \textbf{if} \ \mathrm{TACC} > 0 \ \textbf{and} \\
 & \quad (\mathrm{TCAR} > 0)
\end{cases}
$$

Function

RagA t-C o-Alt

Possible Values: Real

Appears In: Alt-T

DEFINITION

Trac Ra_{34} (Trac Ra Re_{5} · T Co) $\text{Alt}_{\text{f-4}}$

Rus lSo V

Parameters:

Possible Values:

Description:

VRAC cl

VRA v

TRTLM = tau l

TAULIM = Mo

ZPOWN =

ZP ud

ZCLM cl

ZCLM2 = C tau and climb

ZD = and d

ZD 2 d

sed

ZNT C1 and cl

ZNT C and climb

ZNTD d

sed

ZNTD 2 = C and

ZP =

Appears In: Sep rat

DEFINITION

Let:

TRTLIM = Mi Ca $_{1423}$, XTVB))

TAULIM = M

Un $_{402}$,

XTVB $_{1}$]))

REVERSED =

$\begin{cases} \text{True} & \textbf{if } \text{Reversal}_{s\text{-}282} \textbf{ in state } \text{Reversal} \textbf{ or } \dots \\ \text{False} & \textbf{Otherwise} \end{cases}$

TV =

$\begin{cases} 2.5 s & \textbf{if } \text{REVERSED} = \text{True} \\ 5 s & \textbf{Otherwise} \end{cases}$

VRA =

Maintain-Descent **if** Direction Up **and**
[REVERSED = False **and**
(Climb$_{s\text{-}248}$ **in state** ... **state**
...Ground-Str...
(Cross$_{s\text{-}280}$ **not in state** Own-Cross **or**
...]

Climb **if** Direction Up **and**
[REVERSED = True **and**
not (Climb$_{s\text{-}248}$ **in state**
...Ground-Str...
(Cross$_{s\text{-}280}$ **not in state** Own-Cross **or**
...]

Don't- Climb **if** Direction Down **and**
[REVERSED = False **and**
(...$_{s\text{-}384}$ >
Cross$_{s\text{-}280}$ **not in state** Own-Cross)]

Descend **if** Direction Down **and**
not [REVERSED = False **and**
(...$_{s\text{-}384}$ >
Cross$_{s\text{-}280}$ **not in state** Own-Cross)]

DELTA =

$2.5 s$ ... = ... **and**
(REVERSED = True **or**
(Cross$_{s\text{-}280}$ Own-Cross **and**
(...$_{s\text{-}384}$ > 0 **or** ...Ground-Str...

$5 s$ **Otherwise**

ZPOWN =
...(T RTLM Own-Alt-Trac
Own-Alt-Trac ...Adv

ZRPT =
cat_int_i(T RTL,M ... Frac
... Frac RaDN dv

Th:

...

$$\begin{cases} \text{ZPOWN} - \text{ZRPT} & \textbf{if} \text{ ...} \quad \text{Up} \\ -(\text{ZPOWN} - \text{ZRPT}) & \textbf{if} \text{ ...} \quad \text{Down} \end{cases}$$

Funoit n

Rvas lpfatio reaTh V

Parameters: pdeTy

Possible Values:

Description: VRACheint cl
VRADeycatdv
TRTLIM = t Ttau l
TAUtidtau = Mo
TYP = i
ZPOWN = tuCdeap t
ZPTe tudeaptrud
ZCLM1 = tauCdeap cl
ZCLM2 = Cdeap digtaud tau and
cimb
ZDS = gtaudeap t aesd d
ZDS 2 = gtauCdeap d
sed
ZNT C1 gtauCdeap t aimbcl
ZNT C2 digtauCdeap t and
cimb
ZNTD1 = gtauCdeap td
sed
ZNTD 2 = Cdeap digtaub and
dsc
ZMP deapSdeaP =
ZMP =

Appears In: mSprat

DEFINITION

Let:

TRTLIM(...) = Mi Cap$_{123}$, XTYPB])) ...

TAULIM(...) = M

(... Up$_{402}$,

TYPB ...]))

REVERSED =

$$\begin{cases} \text{True} & \textbf{if } \text{Reversal}_{s\text{-}282} \textbf{ in state } \text{[illegible]} \textbf{ or } \text{Reversal} \\ \text{False} & \textbf{Otherwise} \end{cases}$$

TV =

$$\begin{cases} 2.5 \text{ [illegible]} & \textbf{if } \text{REVERSED} = \text{True} \\ 5 \text{ [illegible]} & \textbf{Otherwise} \end{cases}$$

VRA =

Don't-Descend **if** [illegible] Up **and**
[REVERSED = False **and**
(Climb$_{s\text{-}248}$ **in** [illegible] **state**
[illegible] Str [illegible]
(Cross$_{s\text{-}280}$ **not in state** Own-Cross **or**
[illegible]]

Climb **if** [illegible] Up **and**
[REVERSED = True **and**
not (Climb$_{s\text{-}248}$ **in state**
[illegible] Str [illegible]
(Cross$_{s\text{-}280}$ **not in state** Own-Cross **or**
[illegible]]

Don't-Climb **if** [illegible] Down **and**
[REVERSED = False **and**
(Climb [illegible] >
Cross$_{s\text{-}280}$ **not in state** Own-Cross)]

Descend **if** [illegible] Down **and**
not [REVERSED = False **and**
(Climb [illegible] >
Cross$_{s\text{-}280}$ **not in state** Own-Cross)]

DELTA =

2.5 [illegible] **if** [illegible] = [illegible] **and**
(REVERSED = True **or**
(Cross$_{s\text{-}280}$ Own-Cross **and**
(Climb [illegible] > 0 **or** [illegible] Str [illegible]

5 [illegible] **Otherwise**

ZOWN1 =

[illegible] Own-Alt-Rate [illegible],
Own-Alt-Rate [illegible] Rate$_{f\text{-}438}$, VRA, DELTA, [illegible] RTIM)

ZOWN2 =

ciarlrPtj$_{407}$V(Ockr-AItfrac$_{437}$,

Ockr-AItfrac Rat$_{438}$, V RA, EYTA ULIM)

ZNT =

ciarlrPtj$_{407}$V(, Olcr-AItfrac$_{432}$

Ocr-AItfrac Rat$_{433}$

, TDay Adv RTIM) **if** REVE RSED = False

ciarlrPtj$_{407}$V(, Olcr-AItfrac$_{432}$

Ocr-AItfrac Rat Out$_{405}$

, TDay Adv RTIM) **if** REVE RSED = True **and**

(Cross$_{280}$ **in** **state** Cross **or**

(Ocr-AItfrac Rat$_{433}$ 0 **and**

Oth)$_{224}$ RAlt

(Ocr-AItfrac Rat$_{433}$ ·

OwnGlo Alt Rate) 0

ciarlrPtj$_{407}$V(, Olcr-AItfrac$_{432}$

Ocr-AItfrac Rat$_{404}$

, TDay Adv RTIM) **if** REVE RSED = True **and**

not (Cross$_{280}$ **in state** Cross **or**

(Ocr-AItfrac Rat$_{433}$ 0 **and**

Oth)$_{224}$ RAlt

or

(Ocr-AItfrac Rat$_{433}$ ·

OwnGlo Alt Rate) 0

ZNT 2 =

calcProjV(, Ord-At-Trac$_{f-432}$
  Ord-At-Trac Rate$_{f433}$
  TAS, Adv ULIM) **if** REVERSED = False

calcProjV(, Ord-At-Trac$_{f-432}$
  Ord-At-Trac Rate Out$_{f-405}$
  TAS, Adv ULIM) **if** REVERSED = True **and**
  (Cross$_{f3280}$ **in** [illegible] **state** Cross **or**
  (Ord-At-Trac Rate$_{f433}$ 0 **and**
  Oth [illegible]$_{f224}$ Alt
  (Ord-At-Trac Rate$_{f433}$ ·
  Own Alt Rate [illegible] 0

calcProjV(, Ord-At-Trac$_{f-432}$
  Ord-At-Trac Rate$_{f404}$
  TAS, Adv ULIM) **if** REVERSED = True **and**
  not (not Cross$_{f3280}$ **in state** Cross **or**
  (Ord-At-Trac Rate$_{f433}$ 0 **and**
  Oth [illegible]$_{f224}$ Alt
  or
  (Ord-At-Trac Rate$_{f433}$ ·
  Own Alt Rate [illegible] 0

Then:

[illegible]-T-S

ZoneChStat$_{f380}$(OWN2, OWN2, ZXT 1 2) **if** D Up
ZoneDist-S$_{f35}$(OWN2, OWN2, ZXT 2) **if** D

Function

Split-io hcei

Possible Values: Range

Appears In: Dnt- CsteT

DEFINITION

$$\begin{cases} \text{mdpSatn} & \textbf{if } \text{Ssd}_{275} \textbf{ in state } \text{D} \\ \text{mDwatS} & \textbf{if } \text{Ss}_{275} \textbf{ in state } \text{Climb} \end{cases}$$

Funcoit n

Time-To-Co-Alt

Possible Values: Real

Appears In: Rag-At-sCo-Alt-Thr

DEFINITION

Tem Co-Alt =

0	**if** Own-Trac Rag Rat > 10 ft/s
− (ni-Cnt-SV) / (Own-Trac Rel-Alt Rat)	**Otherwise**

Function

TaskStatus

Possible Values: {Not... g, Oldcat... Unconfir... }, Trans...

Description: Th... f-TaskStatus$_{f\text{-}420}$ ... accordance with th... ... Sar... C.

Appears In: ...

DEFINITION

Definition in Data...

Funct ion

Traffic Score

Parameters:
Possible Values:
Appears in:

DEFINITION

Traffic Score =

$$\begin{cases} 1200_{\text{HISCO}} & \textbf{if } \text{RA} \\ C(i)\cdot 0.5_{\text{HISCO}} - \text{Rag} & \textbf{if } \text{TA} \textbf{ and } \text{Wi} \quad \text{Rag} \\ C(i)\cdot 0.4_{\text{SCO}} \dfrac{\text{Rag}}{\text{Rag} \quad \text{Rat}} & \textbf{if } \text{TA} \textbf{ and } \text{Wi} \quad \text{Rag} \textbf{ and } \text{Cgiv} \\ C(i)\cdot 300_{\text{MEDLOSCO}} - \text{Rag} & \textbf{if } \text{TA} \textbf{ and } \text{Wi} \quad \text{Rag} \textbf{ and not } \text{Cgiv} \\ C(i)\cdot 100_{\text{LOSCO}} - \text{Rag} & \textbf{if } \text{AP} \\ 0 & \textbf{Otherwise} \end{cases}$$

Ablations:

RA

$_1$[Other-Aircraft$_{s\text{-}44}$ i] in state Thr ▷ RA

TA

$_1$[Other-Aircraft$_{s\text{-}44}$ i] in state Thre ▷ TA
or
$_1$[Other-Aircraft$_{s\text{-}44}$ i] in state

PA

$_1$[Other-Aircraft$_{s\text{-}44}$ i] in state c

C(i)

C(i) =

$$\begin{cases} 2 & \textbf{if } \ldots \textbf{ in state} \\ 1 & \textbf{if } \ldots \textbf{ not in state} \end{cases}$$

Range

Other-Tracked-Range$_{f\text{-}434}$( )

Range-Rate

Other-Tracked-Range-Rate$_{f\text{-}435}$( )

Within-Range

Other-Tracked … ( ) … Thr

Converging

Other-Tracked-Range-Rate$_{f\text{-}435}$( ) … 10 ft/s

True-Tau-Capped

Possible Values: Real

Appears In: Threat-Status-Info, Threat-Class-Status, Dont-Care-Test, Climb-RL-Strat, Increase-Test-Status, Own-Alt, Rate-Str, Reversal-Test, Rate-Test, Resolution-Strat, TCAS-TCAS-Tau-Rate, VSL-Rate, Alt-Test, Capped-V

DEFINITION

$$\text{True-Tau-Capped} = \text{True-Tau-Uncapped}_{f\text{-}424}$$

$$\text{True-Tau-Capped} = \begin{cases} \text{True-Tau-Uncapped}_{f\text{-}424} & \textbf{if } \text{Own-Tracked-Alt-Rate}_{f\text{-}434} > \text{[illegible]} \; 10\ \text{ft/s} \\ \textbf{Min}(\text{True-Tau-Uncapped}_{f\text{-}424}, \text{[illegible]}) \\ \text{PREV}(\text{True-Tau-Capped}_{f\text{-}423}) & \textbf{Otherwise} \end{cases}$$

Function

Tree-Tau-Unmapped

Possible Values: Real

References: (↑) Range-T (2.30.2),
(↓) ED CT_CONE CTS.Tau calculat

Appears In: Dont- Cst, T Tau- Ra-Tau Capp

DEFINITION

$$\text{Tree-Tau-Unmapped} = \text{Max}(0, \text{MIN\_TAU} - \frac{\text{Count-Trac}\ \ \text{Range}_{434}}{\text{Count-Trac}\ \ \text{Range}\ \ \text{Range}_{435}})$$

[illegible]

Parameters: [illegible]

Possible Values:

Appears In: Climb [illegible] Reversal- [illegible] Reversal- [illegible] Climb [illegible] Climb [illegible]

DEFINITION

[illegible]

Vertical-Miss-Distance

Parameters: REL_Z, REL_ZD, TAU1, TAU2, CLIP

Possible Values:

Appears In: Dnt-... ATlhT

DEFINITION

Let:

$$VMD1 = REL\_Z + REL\_ZD \cdot Min(CLIP, TAU1)$$
$$VMD2 = REL\_Z + REL\_ZD \cdot Min(CLIP, TAU2)$$

Then:

Vertical-Miss-Distance =

$$\begin{cases} 0 & \textbf{if } VMD1 \cdot VMD2 < 0 \\ Min(VMD1, VMD2) & \textbf{if } VMD1 \cdot VMD2 \geq 0 \textbf{ and } VMD1 \geq 0 \\ Max(VMD1, VMD2) & \textbf{Otherwise} \end{cases}$$

(↓E) VERTICAL_MISS_DISTANCE_CALCULATION

Vertical-Projection

Parameters: TPROJ, Z, ZD, ZDGOAL, DIRECTION
Possible Values:
Appears In: VSL-OK

DEFINITION

Let:

$$\text{TUGR} = \frac{|\text{ZDGOAL} - \text{ZD}|}{8\ \text{ft/s}^2{}_{(\text{VACCEL})}}$$

$$\text{TRESC} = \text{TPROJ} - \text{DELAY}$$

$$\text{TACC} = \begin{cases} \text{TRESC} & \textbf{if}\ \ \text{TUGR} > \text{TRESC} \\ \text{TUGR} & \textbf{Otherwise} \end{cases}$$

$$\text{TCGR} = \begin{cases} 0 & \textbf{if}\ \ \text{TUGR} > \text{TRESC} \\ \text{TRESC} - \text{TUGR} & \textbf{Otherwise} \end{cases}$$

$$\text{ZDACC} = \begin{cases} \text{ZD} + \text{DIRECTION} \cdot 8\ \text{ft/s}^2{}_{(\text{VACCEL})} \cdot \frac{TACC}{2} & \textbf{if}\ \ \text{TUGR} > \text{TRESC} \\ \frac{\text{ZD} + ZDGOAL}{2} & \textbf{Otherwise} \end{cases}$$

Then:

Vertical-Projection =

$$\begin{cases} \text{Z} + \text{ZD} \cdot \text{DELAY} & \textbf{if}\ \ \text{TACC} \le 0 \\ \text{Z} + \text{ZD} \cdot \text{DELAY} + \text{ZDACC} \cdot \text{TACC} & \textbf{if}\ \ \text{TACC} \le 0\ \textbf{and}\ \text{TCGR} \le 0 \\ \text{Z} + \text{ZD} \cdot \text{DELAY} + \text{ZDACC} \cdot \text{TACC} + \text{ZDGOAL} \cdot \text{TCGR} & \textbf{if}\ \ \text{TACC} \le 0\ \textbf{and}\ \text{TCGR} > 0 \end{cases}$$

VRC

Parameters:

Possible Values: { 0, 1, 2 }

Description: Th ... RC ...
valu

Value	Meaning
0	N r
1	Descend
2	Climb

Explanation of value selection criteria: If Other Aircraft ... VRC has value 0 ... RA ... has s ... RC ... (1 ...). ... has s ... Climb RA ... RC ... 2 (Climb)

(...) S ... (... 7).

Appears In: ...

DEFINITION

VRC =

$$\begin{cases} 0 & \textbf{if } [\text{Other-Aircraft}_{s\text{-}44}] \textbf{ not in state } \text{Thr} \\ 1 & \textbf{if } [\text{Other-Aircraft}_{s\text{-}44}] \textbf{ in state } \text{Thr} \triangleright \text{Descend} \\ 2 & \textbf{if } [\text{Other-Aircraft}_{s\text{-}44}] \textbf{ in state } \text{Thr} \triangleright \text{Climb} \end{cases}$$

VSB

Parameters: CVC, VRC

Possible Values: { 4}..

Description: (PIB) (Ta 3–2).

Appears In: Real gsRest gsRest

DEFINITION

VSB =

$$
\begin{cases}
0 & \textbf{if } CVC{=}0 \textbf{ and } VRC{=}0\\
14 & \textbf{if } CVC{=}0 \textbf{ and } VRC{=}1\\
7 & \textbf{if } CVC{=}0 \textbf{ and } VRC{=}2\\
11 & \textbf{if } CVC{=}1 \textbf{ and } VRC{=}0\\
5 & \textbf{if } CVC{=}1 \textbf{ and } VRC{=}1\\
12 & \textbf{if } CVC{=}1 \textbf{ and } VRC{=}2\\
13 & \textbf{if } CVC{=}2 \textbf{ and } VRC{=}0\\
3 & \textbf{if } CVC{=}2 \textbf{ and } VRC{=}1\\
10 & \textbf{if } CVC{=}2 \textbf{ and } VRC{=}2
\end{cases}
$$

Other-Projected-Alt

Possible Values:

References: (↓)ED CT_CONE CTS.Tau calculat

Appears In: Catch-L Off, Ch e
Int-Cross-N Cyc T CAS-TCAS-C T
Try-VSL-T

DEFINITION

Other-Alt-Trac p.432 (Other-Alt-Trac Range-Tau- Tru Cap p.423)

State

Other [illegible] Firmness

Possible Values:

Description: Returns [illegible] by the ACAS SARPS [illegible] RM (CAS [illegible]) [illegible] in [illegible])

References: [illegible] S 2.36.3

Appears In: [illegible]

TCAS-TCAS-[illegible] Rate

DEFINITION

[illegible]

Function

Other-Tracked-Alt

Possible Values:

Description: Return the value ... by the ACAS SARPS ... Alt ... OPS ... (... 6)

Appears In: 100 ... Alt-S ... Climb ... Other-Status, ... Other-Rel-Alt, ... Other-Relat ... TCAS-TCAS ... OK

DEFINITION

Primitive.

Other ed-Alt-Rate

Possible Values:

Description: th CAS SARS

A

V

)

References: (↑) (2.8)

Appears In: y-Arr Rat, t, Trac Alt Rat, T-S, T Rat, Alt-T

DEFINITION

Function

Other-Tracked-Range

Possible Values:

References: (↓) TRACKING ... Track

Appears In: ... Tau-rud ... Cap ...,
... Tau- ... Update, Other-Trac ... Range ... Rate,
Other-Trac ... Thr ... Range-T ...
... c-Ground ... Range-At- ... Co-Alt, Tau-Rising,
... Tra ... Cap ...,
... Tau- ... Update

DEFINITION

Let:

$\Delta t =$ Range ... Point-T $-$ Prev(Range ... Point-T)
$RP =$ Prev(Other-Trac ... Range$_{434}$) ... Prev(Other-Trac ... Range ... Rate$_{435}$) $\cdot$ Δt

Then:

Other-Trac ... Range =

$$\begin{cases} \text{Max}(RP + 0.4 \cdot (\text{Range}_{294} - RP), 0) & \textbf{if } \text{Other ... Range-Val}_{296} = \text{True} \\ RP & \textbf{if } \text{Other ... Range-Val}_{296} = \text{False} \end{cases}$$

Function

Other-Tracked-Range-Rate

Possible Values:

References: (↑) Other-Intruder-Data (2.8),
(↓) TRACKING ... Track

Appears In: Tau-... Update, Other-Tracked-Range-Rate,
Other-Tracked-Range-Threshold, Range-T...,
Range-Str..., Range-Alt-... Alt-Thr...,
Tau-Alt-Reduced-T..., Alt, Tra...-Sc...,
Tau-... Update

DEFINITION

Let:

$\Delta t =$ Range-... $-$ Prev(Range-...)

$RP =$ Prev(Other-Tracked-Range$_{434}$) $-$ Prev(Other-Tracked-Range-Rate$_{435}$) $\cdot \Delta t$

Calculated RR $=$ Prev(Other-Tracked-Range-Rate$_{435}$) $\frac{\text{Range}_{294} - RP}{\Delta t}$

Then:

Other-Tracked-Range-Rate $=$

$$\begin{cases} \text{Max}(\text{Calculated RR}, 0) & \textbf{if } (\text{Other-Range-Valid}_{296} \text{ True}) \textbf{ and } (\text{Other-Tracked-Range}_{434} > 0) \\ \text{Calculated RR} & \textbf{if } (\text{Other-Range-Valid}_{296} \text{ True}) \textbf{ and } (\text{Other-Tracked-Range}_{434} = 0) \\ \text{Prev}(\text{Other-Tracked-Range-Rate}) & \textbf{if } \text{Other-Range-Valid}_{296} \text{ False} \end{cases}$$

Function

Other-Tracked-Relative-Alt-Rate

Possible Values:

Appears in: [illegible]-Alt,
TCAS-TCAS-[illegible],
[illegible]-Alt, Try-VSL-T

DEFINITION

Own-Tracked-Alt-Rate$_{f\text{-}438}$ − Other-Tracked-Alt-Rate$_{f\text{-}433}$

Function

Own-Tracked-Alt

Possible Values:

Appears In: ...-Alt-S ... Lateral-Alar... Crossing ... Latch-L ... Off, ... Climb ... Crossing ... Alt-S ... V ... Descend ... Alt-L ... Climb ... Incr ... Climb ... Cross-N ... Cycl ... Climb ... Incr ... Other ... Alt ... Own-Tracked-Alt, ... Own-Tracked-Alt-Rate ... Own-Tracked-Alt ... Ground ... Ground ... Relat ... TCAS-TCAS ... OK

DEFINITION

Let:

$\Delta t = t(\ldots) - t(\textsc{Prev}(\ldots))$

$ZP = \textsc{Prev}(\text{Own-Tracked-Alt}_{f\text{-}437}) + \textsc{Prev}(\text{Own-Tracked-Alt-Rate}_{f\text{-}438}) \cdot \Delta t$

Then:

Own-Tracked-Alt

$$\begin{cases} ZP + \ldots(\ldots - ZP) & \textbf{if } t \geq t(\text{TCAS-Onset}) + 1s \\ \ldots & \textbf{if } t < t(\text{TCAS-Onset}) + 1s \end{cases}$$

Comment:

Return the value ... CAS SARPS ... OPS ...).

Own-Tracked-Alt-Rate

Possible Values:

Appears In: ...-Status ... Climb ... Climb ...
Don't-... Climb
... Trac ... Alt ... Rate,
Own-Alt ... Rate, ... Alt,
Own-Tracked-Alt ... Rate, Own-Tracked-Alt,
...
...-V-S,
TCAS-TCAS ...
... OK

DEFINITION

... $_{v-250}$

Let:

Δt ... $_{v-249}$ ... $- t($Prev ... $_{v-249}$ ...$)$

$ZP = $ Prev(Own-Tracked-Alt $_{f-437}$) ... Prev(Own-Tracked-Alt ... Rate $_{f-438}$) $\cdot$ Δt

Calculated-AR =

Prev(Own-Tracked-Alt ... Rate $_{f-438}$, 0) ... 25 ... $\cdot$ $\frac{\ldots_{v-249} - ZP}{\Delta t}$

Then:

Own-Tracked-Alt ... Rate =

$$\begin{cases} \text{Calculated-AR} & \text{if } t \geq t(\text{TCAS-On}) + 1s \\ \text{Held ... Rate}_{f-253} & \text{if } t < t(\text{TCAS-On}) + 1s \end{cases}$$

= ... $_{v-250}$ ... **Comment:**

... CAS SARPS ... the ... of ... aircraft (the ... OPS ...).

Separation

Parameters: [illegible]

Possible Values: [illegible]

Appears In: [illegible]

DEFINITION

Let:

TRTLIM = Min(Cap[illegible] XTVB[illegible])

TV =

$$\begin{cases} 2.5 s & \textbf{if } \text{REVERSED} = \text{True} \\ 5 s & \textbf{Otherwise} \end{cases}$$

Then:

Separation =

[illegible] Separation(D[illegible]) [illegible]

[illegible] TV[illegible] if [illegible] = [illegible] **and** TRTLIM $\leq$ TV

[illegible] TV[illegible] if [illegible] = [illegible] **and** TRTLIM $>$ TV

VMD

References: (↓) RESOLUTION_AND_COORDINATION [illegible] _ [illegible]
(↓) [illegible] _ proj _ [illegible]

Appears In:

DEFINITION

Let:

REL_Z = Own-Aircraft$_{f-437}$ - Other-Aircraft$_{f-432}$

ZDOW =

$$\begin{cases} \max([illegible] & \textbf{if } {}_1[\text{Other-Aircraft}_{s-44} \text{ i}] \textbf{ in state } \text{Climb} \\ \quad \text{Own-Aircraft Rate}_{f-438}) & \\ \min([illegible] & \textbf{if } {}_1[\text{Other-Aircraft}_{s-44} \text{ i}] \textbf{ in state } \text{D} \\ \quad \text{Own-Aircraft Rate}_{f-438}) & \end{cases}$$

REL_{ZD} = ZDOW - Other-Aircraft Rate$_{f-433}$

TRTRU = Tau-Cap$_{f-423}$

TAUR = Tau-Cap$_{f-40}$

TVE = XTVE [illegible] $_1$ [illegible]]

Then:

VMD [illegible]$_{f-420}$(REL_Z, REL_ZD, TRTRU, TAUR, TVE)

State

Other Aircraft

Possible Values: Not-Tracked, Tracked, Threat-Not-Tracked

Description: *Not-Tracked* ... track ... If ... advisory ... list

Comments: Other-VRC and Other-RC are always updated ...

References: (↓) TRACKING TRACK_INTRUDERS, (↓) RECEIVE ... _entry, (↓) Threat ... _field ...

Appears In: CVC, ... Addr ... Tracked ... RC

DEFINITION

Not-Tracked, Threat-Not-Tracked ⟶ Tracked

started tracking a new aircraft	T
Standby-Ground $_{m-348}$	F

Tracked ⟶ Not-Tracked

Surv ... $_{v-283}$	T	·
at ... **in state** Threat ▷ ... Advisory ▷ RA	F	·
Standby-Ground $_{m-348}$	·	T

Tracked ⟶ Not-Tracked

(Surveil ... $_{v-283}$	T	·
at ... **in state** Threat ▷ ... Advisory ▷ RA	T	·
Standby-Ground $_{m-348}$	·	T

Not-Tracked ⟶ Threat-Not-Tracked

selected to model intruder (event)	T
$\text{Standby-Condition}_{\text{m-348}}$	F

Threat-Not-Tracked ⟶ Not-Tracked

$t(\text{Other-VRC}_{\text{v-289}}) - t > 6\ \text{s}$	T	·
$\text{Standby-Condition}_{\text{m-348}}$	F	T

Performance Monitor

We did not include the performance monitor requirements in our original TCAS specification as the requirements were too vaguely specified by the FAA. Although we did not create a new one for this example specification, we note that it now appears that it would not be difficult to create such a specification from the upper two levels of this intent specification.

Testing Requirements

Several hundred pages of the TCAS MOPS (DO-185) includes detailed testing requirements. We have not bothered to reproduce that information here as this is only an example specification.

Level 4

Physical and Logical Design Representations

This level of an intent specification contains the normal physical and logical design representations with links to the level above.

Human–Computer Interface Design Specification

ANNCIS describes in detail the Human–Computer Interface Design. We have not bothered to reproduce this material here, but it obviously could be included.

Aircraft Flight Manual

The aircraft flight manual (AFM) or flight manual supplement (AFMS) is essentially what is sometimes called the user manual in software and system engineering. The information here should obviously also be linked to the higher levels. We have included only a few statements that might be included in such a manual along with their links as an example. The links would not be included in the real manual given to the pilots, of course.

[illegible] onal proc
[illegible] mply: “K

[illegible] ll b
[illegible] app
[illegible] and you will h
[illegible] ” (↑3.3). Promptly and smoothly r
[illegible] on you
minimizing how far you have deviated from your clearance (↑OP.3).

Since the evasive action required by TCAS to resolve most conflicts is small in distance and duration, most of these maneuvers will be of little consequence to ATC. However, you may wish to notify ATC that a situation occurred requiring you to maneuver based on a TCAS advisory, when time and cockpit work load permit.

A backup source of information on the IVSI appears in the upper left corner of the traffic display. This symbology consists of green arrows for climb and descend advisories, and the international “do not” symbol over white arrows to indicate that vertical speed must be limited either up or down (↑3.ta-display). Refer to the IVSI eyebrow lights to see what the limit is (↑3.ra-display).

Maneuver advisories that might be triggered when intentionally flying close to another aircraft should be inhibited. For example, when parallel visual approaches are in progress to runways spaced less than 3000 feet apart, select "TA only" on the transponder mode switch (↑SC6.4, OP.6, 3.3). This will prevent maneuvering advisories being issued for planned, close separation (↑SC6.1). A traffic advisory may still be issued, which will call your attention to the location of the aircraft on the other approach. The normal operating position for the transponder switch is TA/RA which provides full functioning of both the TCAS and the transponder (↑3.controls).

Software Design Specifications

The requirements for TCAS were originally specified by MITRE using a form of pseudocode that they designed. We have duplicated that pseudocode here without any changes. A design specification that we were writing ourselves would not be in this form. However, we did not feel that preparing a new design specification was necessary as this document is simply an example. We are working on direct code generation from the SpecTRM-RL model in Level-3 so this section may become unnecessary in the future or it might include some information generated by the code generation program that would be helpful in understanding and evaluating the generated code.

Overall Design

The CAS logic loops in a one-second-logic cycle that continues forever. The high-level design is shown below:

```
LOOP:
 ;      Obtain input from surveillance functions
        Perform aircraft tracking for own aircraft and intruders
    ;   Perform threat detection
        REPEAT        ; (until) threats remain in
                  IF (this is a terminating threat)
                        THEN ;
                        ELSE IF (this is a new threat)
                               THEN ; select sense
                            ;  ELSE consider reversal of sense
                          ; Select advisory
                            IF (multiple threats detected)
                               THEN ; perform multiaircraft logic
                            IF (this is a continuing threat)
                               THEN ; consider increase in vertical rate
                     ;      Update resolution advisory array
                            IF (threat is TCAS equipped)
                               THEN ; send intent and await reply
        ENDREPEAT
      ; Generate traffic advisory data
 ;      Display traffic and resolution advisory data
ENDLOOP
```

Global Data St

This section defines the global data structures, system parameters, and interfaces with the CAS logic. The information contained in these entities is required to be available to all the tasks in the following sections. Additional variables local to an Individual task are defined in the section containing that task.

TCAS maintains several global data structures: These are the Intruder Track File (ITF), the Threat File (TF), the
File (structure may be thought of as a linked list of entries, with each
entry corresponding to a unique aircraft. Information is retained from cycle to
cycle for each structure (except for the
implementation must account for means, transparent to the logic, of accessing and organizing the data in these structures. An example would be pointer variables to select the entry in one structure corresponding to the aircraft represented in a given entry in another structure. Also, the implementation needs a means to loop through all entries in a given structure, as well as to update linkage pointers when entries are created or deleted. The details of these operations are not included here.

Intruder Track File (↓STRUCTURE ITF)

The Intruder Track File (ITF) is the collection of data saved for each intruder under consideration by the CAS logic. The data stored for each intruder includes of identity and cross-reference pointers, equipage capability, position data, threat
detection data, advisory evaluation data, advisory delay
calculation data, traffic advisory data and timer values.

Threat File (↓STRUCTURE TF)

The Threat File (TF) is the collection of data saved for each intruder which satisfies either or both of the following conditions:

a. Own TCAS has declared the intruder a threat and has generated a resolution advisory.

b. The Intruder is TCAS-equipped and has generated a resolution advisory against own TCAS.

s' identity, own The information in a threat file entry consists of the threat
s' advisories and time of resolution advisory, indices to own and the threat
information.

Working List (↓STRUCTURE WL)

)WL(Working temporary list of those intruders for which resolution
advisories and Threat File entries need to be updated. It is cleared of all entries
DETECT CONFLICTS at the beginning of each cycle. Its entries are created in the
task (Page 500) RESOLUTION The entries are used in R
(Page 523).

A significant consideration In the maintenance of the Working List is the order-
ing of its entries. SET-UP governs the placement of each new entry.
Those that have been terminated as threats must precede new threats, which in
turn preceded continuing threats. This ordering allows the R ESOLUTION
CONTROL task (Page 523) to property handle multiaircraft situations.
Whereas in the specified order. insert new entries must be implemented

Nonlinear Track File (↓STRUCTURE N)

The Nonlinear Track File (N) is the collection of data stored for each aircraft
tracked by Mode C reports. An entry is maintained for each Mode C intruder.
Another entry is maintained for own aircraft's track if Mode C data is being used.

The N entry contains data retaining Mode C bin-crossing time, Mode C reports,
estimates of rate uncertainty, the character or trend of the report sequence, the
confidence (firmness) of the tracked rate and various timers. Its principal entries
are the altitude and altitude rate estimates derived by the tracking algorithm.

TCAS Global Variables (↓STRUCTURE G)

Structure G contains those used by other variables considered global to TCAS
structures or retained from cycle to cycle. While ITF, TF, Unlike the task,
and N, whose variables are associated with a particular aircraft and which have
many entries when many aircraft are under consideration, the Global Structure G
contains only one entry. The variables encompass the status of own TCAS, own
sensitivity level, own altitude, settable parameters based on sensitivity level, vari-
ables associated with coordination, and all display information related to resolution
advisories. The cross-reference table to surveillance numbers may be sized accord-
ing to the size of the track file anticipated. The number given in the pseudocode
is an illustration.

Temporary Lists

The Display Traffic Vector (TDV) describes information to be passed to the display
for Traffic Advisories.

AT- Delete-RA list and temporary lists created by THR
FIRE G to save values of indexes to resolution complements stored
HISTORY, which are used by R G.

Interface Bu

Items pertaining to onboard sources from own aircraft (
(0) are all considered external to the CAS logic and relate to status information and geometry:

- An input from the Performance Monitor Function of TCAS that decides whether TCAS is operating properly. If it is not, active interrogations and virtually all of CAS logic is disabled.
- An input from a strut or landing gear switch that signals when own aircraft is on the ground. If this switch is not present, the data input must give the value *false*.
- An input from the TCAS Installation that tells the sensitivity level selection logic whether or not display of traffic is permitted while own aircraft is on the ground.
- Manual (pilot) sensitivity level control setting.
- Radar attitude, which must be digital data (or an analog voltage which is then digitized by TCAS hardware), that indicates altitude above ground level. (A limited number of discrete signals corresponding to specified radar altitude trip points is not an acceptable radar altitude input.) The CAS logic samples this data once per cycle, independent of the data update rate of the radar altimeter. A flag denotes whether valid data is contained in the field. This flag should be cleared when the radar altimeter loses lock, including flight above its maximum operating height above ground level.
- Own altitude, which may be input to TCAS in either of two ways: (1) own Mode C altitude, quantized to 100-foot steps or (2) a more finely quantized altitude, from a source such as an Air Data Computer (which should be used if available). Vertical tracking requires that this input be quantized to 10 ft or smaller. A parameter must be set to tell the vertical tracking logic which source to use. It is not intended to dynamically switch from one source to another in flight. Although alitude rate from the Air Data Computer Is not normally used, as it may be filtered in a manner not optimum for CAS logic, an input is provided to initialize own altitude rate at system startup. This

is useful if TCAS is turned on or reinitialized in flight. If the attitude rate is not available, the input should be set to zero.

- [illegible] relative to magnetic [illegible] lating the crosslink data to send to a TCAS-1 threat.

- [illegible] on the TCAS/Mode [illegible] register is [illegible], then is later cleared when the state is read. If an [illegible] switch is not provided, this field should be either always set or always cleared within the surveillance logic (to enable or disable proximate traffic display).

  [illegible] Each cycle, the value of this input is checked in Process TRAFFIC [illegible] If it is set, the time-to-display is initialized to a value of the maximum time for proximate traffic display, nominally 15 seconds.

  [illegible] duration of the traffic presentation and terminates the display when the time-to-display decreases below zero. [illegible] of the "All-tracks [illegible] imate traffic presentation is taking place results in extension of the display by a set amount of seconds from the time of switch selection (the amount is set as one of the system parameters).

- Other inputs from external sources such as own aircraft Mode S identifier and indications that own aircraft cannot climb at 1500 or 2500 fpm. These inputs, although used by the TCAS logic, are not defined in the 0 structure because they are implementation dependent: display of traffic while own aircraft is [illegible] sensitivity level control requirements [illegible] alpha/beta parameter values for tracking own aftftude, which permits less restrictive quantization of own altitude input data.

The surveillance function passes data items relating to particular intruders to the [illegible] Intruder Surveillance [illegible] data from [illegible] the current cycle:

- Flags indicating whether valid range and attitude data are present.

- The surveillance file number generated by the surveillance system. This number will remain the same for the lifetime of the surveillance track on this aircraft. The CAS logic maintains a cross-reference table that gives, for each possible surveillance file number, the number of the Intruder Track File [illegible] indicate that no Intruder Track File entry exists for that surveillance file number. This cross-reference table provides a convenient link for routinely associating a surveillance report with an Intruder Track File.

- The 24-bit Mode S address of the intruder, if so equipped. This is used for two purposes by the CAS logic. First, the Mode S address is used to identify the intruder to be interrogated in a coordination message to a particular TCAS threat. Second, during coordination, if own TCAS and a threat TCAS select incompatible vertical senses, the aircraft with the higher Mode S address will reverse its sense.

- The coding in the surveillance reply if the Intruder is Mode-S equipped. It is set to a unique value by the surveillance system if the intruder is ATCRBS.

- The sensitivity level of the intruder as received in the SL field of the surveillance reply.

- The surveillance data for this cycle.

TCAS message data received through the transponder from external sources:

- Data obtained from the TCAS Receiver, also obtained by the Surveillance Reply is data received by own aircraft transponder (a Mode S transponder is part of the required TCAS 11 equipment). RF messages may come from other TCAS units or Mode S ground sites. Some of this data is addressed to TCAS. It is processed by the R Task. Messages may also contain own aircraft information that is input to the transponder and passed to TCAS for processing.

- Messages received from own aircraft transponder are placed into one of two first-in-first-out input queues depending on priority. Coordination messages use the higher priority queue. All other messages use the lower priority queue. The use of these queues does not define a bu … structure. The implementation must provide a means of accessing and organizing the messages in these queues. The size of the low priority queue shall be consistent with the maximum rate at which low priority (periodic) data can be received from the transponder and the minimum rate at which such data can be processed. The high priority queue and its associated processing shall be capable of handling a minimum of 60 coordination messages per second, of which 10 messages are unique (not duplicates) with no queue overflow and no degradation in system performance. An overflow in the high priority input queue shall be recognized by the performance monitor.

TCAS Pseudocode

This pseudocode was taken verbatim from the Minimum Standards for TCAS II Equipment, Volume II (

TASK MAIN_LOOP ... the CAS (Collision Avoidance System) logic. Their input is processed surveillance data and the output is information for cockpit display and reply messages. Task MAI ... responds to certain incoming messages addressed to TCAS.

```
TASK MAIN_LOOP;
     PERFORM Initialize;
     REPEAT WHILE (always);
         Set Working List to null;
         <if processing of Mode S messages is not performed on a periodic
         basis per cycle, then ROUTINE MODE_S_MESSAGE_PROCESSING must be
         invoked here in TASK MAIN_LOOP to handle any Mode S messages that
         have arrived while own TCAS was in Coordination Lock State.>

     CALL TRACK_OWN;
     IF (interrogation allowed)
         THEN CALL TRACK_INTRUDERS;
            IF (Resolution allowed)
                   THEN CALL DETECT_CONFLICTS;
     REPEAT WHILE (more entries in Working List)'
          CALL REOLUTION_AND_COORDINATION;
          Select next Working List entry;
     ENDREPEAT;

     CALL traffic_advisory;
     CALL display_advisories;
     CALL housekeeping;

     ENDREPEAT;

END MAIN_LOOP;
```

MAIN, which begins by first clearing the threat
Task TRACK-OWN. This task updates own altitude, derives own altitude rate, determines if own aircraft climb performance is limited and checks for valid radar altitude data. It updates its estimate of ground level, which is the altitude reported by aircraft on the ground, and performs a ground proximity check to determine if own aircraft is at an altitude where aural annunciations should be disabled or resolution advisories to descend should be disabled. It then determines own sensitivity level and altitude layer, sets layer dependent parameters, and decides whether own TCAS may generate surveillance interrogations and traffic resolution

Level Control. Sensitivity ... TRACK-OWN

Update messages, which are subsequently sent to own Mode S transponder.

If own aircraft is permitted to generate surveillance interrogations, MAIN then calls the task TRACK-INTRUDER. This task associates each surveillance report with an Intruder in the Intruder Track File (ITF) entry. For a new Intruder, an ITF entry is created. Range and altitude estimates are updated. For intruders not reporting altitude, only range is updated. The task updates information such as the intruder's CAS sensitivity level, and whether the intruder is climbing, descending or level, and decides whether the intruder is on the ground, in which case no advisory will be generated. Finally, the task deletes ITF entries dropped by the surveillance function.

After TRACK-INTRUDER, MAIN calls Task DETECT.

If TCAS may generate resolution advisories, this task tests each altitude reporting intruder in the ITF against the threat detection criteria. In addition, Task DETECT incorporates various methods by which altitude-crossing resolution advisories can be deferred or avoided entirely.

RESOLVE is the next task called by MAIN after DETECT. This task is called to process each threat declared by DETECT. A sense (climb or descend) is selected for each new threat, and a resolution advisory is selected for new and continuing threats. The task includes provisions for handling multiple threats. RESOLVE has the capability to reverse the sense originally selected and to call for an increase in vertical speed in the selected sense (crossing or noncrossing) if it is projected that adequate altitude separation may not be achieved at the point of closest approach.

Task HOUSEKEEP is the last task in MAIN. Its function is the cleanup of data structures, which may be necessary due to irregular sequences of events.

```
PROCESS Initialize;

     Initialize all constants;
     Set all data structure to null;
     Clear displayed resolution advisory array;
     Set cross reference table to null;
     initialize sensitivity level and layer to minimum;
     initialize display variables referring to previous_cycle RAs;
     initialize coordination intent array;
     initialize ‘‘increase rate RA issued" flag;
     initialize ‘‘DESCEND RAs inhibited flag;
     initialize flag to disable aural annunciations;
     clear coordination lock flag;
     initialize coordination message queues and pointers;
     clear all bus working areas and number of TCAS targets counter;
     start real time clock;
     send initial sensitivity level setting to transponder;
     input own aircraft info; <altitude, Mode S ID, Max, Airspeed,
                 Control panel settings>

END Initialize;
```

References:

The task is called by interrupt in response to certain incoming messages that are received by the Mode S transponder and passed to TCAS for processing. These messages may arrive at any time during the one-second TCAS logic cycle and are of four types: 1) TCAS Resolution Messages (coordination messages) from another TCAS, 2) TCAS Broadcast Interrogation Messages from another TCAS, 3) Sensitivity Level Command Messages from a Mode S ground sensor, and 4) periodic own aircraft data messages.

```
TASK Receive;
   <TASK RECEIVE is invoked upon receipt of an incoming message on the
    transponder-to-TCAS bus.  It determines the message type and then
    places the message on one of two queues---one queue handles TCAS
    resolution (Intent) messages, and a second queue handles other Mode
    S messages (Sensitivity Level Commands, Periodic Aircraft Data, and
    TCAS Broadcast Messages).  TASK RECEIVE then activates (calls) the
    proper routine to process messages on the specific queue. If TCAS is
    not in Coordination Lock State, the message will be processed immediately.
    However, processing for Resolution (Intent) messages shall be deferred if
    the Coordination Lock flag is set.  Also, because Mode S messages are
    lower in priority than Resolution messages, any Mode S message processing
    shall be deferred until completion of the higher priority processing
    associated with coordination.  Any queued Resolution messages that are
    not processed immediately will be processed later when ROUTINE
    COORDINATION_UNLOCK is called by the TASK that originally generated
    the coordination lock.  Queued mode s messages will be processed either
    periodically by ROUTINE MODE_S_MESSAGE_PROCESSING, or at the beginning
    of the next logic cycle, when TASK MAIN_LOOP calls ROUTINE
    MODE_S_MESSAGE_PROCESSING, if processing of queued mode s messages is
    not performed on a periodic basis.>

    IF (TCAS Resolution (Intent) message received;
       THEN put message on Resolution message queue;
          CALL RESOLUTION_MESSAGE_PROCESSING;
    ELSEIF (Mode S message received)
       THEN put message on Mode S message queue;
          CALL MODE_S_MESSAGE_PROCESSING;

END Receive;
```

Each time a message Is placed in the Resolution Message queue, R MSSA GE-PROC G is called. If a coordination lock state is not currently in effect, the message is processed immediately. R GE-PROC G will be called to process any queued messages at that time. If the current lock state is threat-initiated (own aircraft is processing a received TCAS Resolution Message), queued messages will be discovered either when the queue is examined within R GE-PROC G or MONITOR.

Likewise, each time a message is placed in the Mode S Message queue, MO SSA GE-PROC G is called. If a coordination lock state is not currently in effect, the message is processed immediately. If a coordination lock state is in effect, the high priority messages exit; however, the low priority messages are not processed when the current lock state ends. If processing of Mode S messages is not performed on a periodic basis each cycle, NOP MAI must examine the Mode S message queue at the beginning of the next logic cycle and call MO GE-PROC G to process any queued messages at that time.

Note that TCAS enters a coordination lock state when processing messages in the Resolution queue, and not when processing messages in the Mode S Message queue. This means that the processing in R GE-PROC G cannot be interrupted by any incoming messages, but that the processing in MO GE-PROC G can be interrupted by both incoming TCAS Resolution Messages and Mode S messages. The implementation must ensure that MO GE-PROC G is prevented from processing an incoming Mode S message while the processing of a previous Mode S message is still underway. (Messages within a queue must be processed In the order received.) The routines R GE-PROC G and MO GE-PROC G must be re-entrant, since all incoming messages go through a minimum amount of handling regardless of current lock state. (At the very least each incoming message is immediately queued, the appropriate processing routine is called, and the coordination lock state is checked.

```
ROUTINE RESOLUTION_MESSAGE_PROCESSING;

   IF (not in coordination lock state) <NOTE: Uninterruptible test and set
                                              instruction>
       THEN SET G.COLOCK;
           Save Lock time;
           REPEAT WHILE (more entries in Resolution message queue);
               Get Resolution message from queue;
               PERFORM Process_threat_intent;
               Select next entry;
           ENDREPEAT;
           Send Coordination Update message to transponder;
           CALL COORDINATION_UNLOCK;

END RESOLUTION_MESSAGE_PROCESSING;
```

321 Update (t) Coordination- **References:**

```
PROCESS Process_threat_intent;

    <This process analyzes the intent sent by the threat tcas aircraft>

    CLEAR bad data indicator;
    set up Parity;
        <Use (#33 through          CVC, VRC) in coordination message from
        threat TCAS as index into parity table.
        Bit positions may be different in the message received from the transponder>;

    IF (Parity not equal to VSB subfield of coordination message from threat TCAS)
         THEN Bad data received;
         ELSE IF)0(CVC, CHC, VRC, and HRC) are all
               THEN Bad data received;
               ELSE IF (0)CVC or VRC are
                    THEN bad data received;
     IF (Message did not contain bad data)
          THEN PERFORM Find_threat_file_entry;
             Update refresh timer for threat complement'
             PERFORM Process_valid_data;

END Process_threat_intent;
```

344 **References:**

345 (1) Receive-Valid-Resolution-Message

```
PROCESS Find_threat_file_entry;

    <This process locates the threat file entry corresponding to the ID
    of the current threat>

    CLEAR Success_flag;
    REPEAT WHILE (more entries in Threat file AND Success_flag not set);
         IF (entry found with ID matching threat whose msg being processed)
               THEN SET Success_flag;
                    Save pointer to Threat File entry;
               ELSE select next entry;
     ENDREPEAT


     iF (Success_flag not set)
           THEN create new Threat File entry;
                Initialize fields and timers;
                Save pointer to Threat File entry;

END Find_threat_file_entry;
```

44s- (1) Other Aircraft 1

References:

```
PROCESS Process_valid_data;
   <This process peformed upon successful receipt of data from transponder>

     IF (message contains vertical complement to cancel)
         THEN CLEAR vertical complement index in Threat file;
              CALL DELETE_INTENT IN (P.VMAPINTENT(CVC));

     IF (message contains horizontal complement to cancel)
         THEN CLEAR horizontal complement index in threat file;
            Save low order bits of CHC subfield in L2CHC;
              CALL DELETE_INTENT IN (P.VMAPINTENT(L2CHC));

     IF (message contains vertical complement to add)
         THEN If (index in threat file does not already indicate desired value)
                 THEN enter complement value in index in threat file;
                      SET indicated complement in intent array;

     IF (message contains horizontal complement to add)
        THEN Save low order bits of HRC subfield in L2HRC;
             IF (index in threat file does not already indicate desired value)
                 THEN enter compelment value in index in threat file;
                    SET indicated complement in intent array;

END process_valid_data;
```

(1) Other-HRC References: s-802 290

```
ROUTINE MODE_S_MESSAGE_PROCESSING;

    REPEAT WHILE (more entries in Mode S message queue AND
                  not in coordination lock state AND processing of previous
                  Mode S message complete);
       Get message from queue;
       IF (message is Sensitivity Level Command)
           THEN PERFORM SL_command_processing;
       ELSEIF (message is Periodic Data)
           THEN PERFORM Periodic_data_processing;
       ELSEIF (message is TCAS Broadcast data)
            THEN PERFORM Broadcast_processing;
        Select next entry;
    ENDREPEAT;

END MODE_S_PROCESSING;
```

```
PROCESS SL_command_processing;

    IF (message indicates cancellation of SLC)
       THEN cancel SL for this site;
    ELSEIF (valid SL code indicated)
       THEN save sensitivity level for this Mode_S site ID;
            Update sensitivity level timer for this site ID;
    OTHERWISE;  <no action if no valid code>

END SL_command_processing;
```

References: Ground-Commanded-S

```
PROCESS Periodic_data_processing;

    Determine if the data received is own altitude, own Mode S ID,
       own aircraft maximum airspeed, or control panel information'
    IF (altitude message received)
        THEN IF (fine resolution)
             [illegible] altitude from message;
                 O.[illegible];
             [illegible] altitude from messasge;
                 O.[illegible];
    IF (own update message received)
         THEN G.ID[illegible] Mode S ID;
             O[illegible]selected sensitivity level from control panel;
             Save maximum airspeed for use by surveillance;
             Save other display control data, control panel data, hardware
                 status info for use by display subsystems, surveillance,
                 and performance monitoring;

END periodic_data_processing;
```

References: [illegible] Mode-Selector [illegible]
[illegible] Own-Mode-S-Address 254

```
PROCESS Broadcast_processing)

    Store the Mode S ID of the reporting TCAS aircraft and the associated time;

    <Once per second, at the beginning of the surveillance cycle, compute NTA
     to be the number of distinct TCAS addresses monitored within the
     previous period.>

END broadcast_processing)
```

```
ROUTINE DELETE_INTENT IN (pointer to intent))

    IF (intent to delete is not null)
        THEN REPEAT WHILE (more Threat File entries AND match not found);
             Search for another Threat File entry with same intent;
             ENDREPEAT;

             IF (matching entry found)
                THEN;  <cannot delete intent that applies to another threat>
                ELSE CLEAR this entry in intent array;

END DELETE_INTENT
```

Tracking

The TCAS tracking logic is in tasks TRACK-O
... and the ... The input to these two tasks is the Own Surveillance
(consisting of structures 0 and S, respectively) to determine the positions and rates of own and intruder aircraft, and to determine the values of thresholds dependent upon altitude. Once these two tasks are completed, TCAS has the necessary information to determine which intruders qualify as threats.

```
TASK TRACK_OWN;

   <This task tracks own altitude, tests radar altitude, selects
    sensitivity level, turns interrogation and resolution function
    on or off, and estimates ground level>

    Save time of new processing cycle;
    Save operational status of TCAS;
    PERFORM Own_altitude_tracking;
    PERFORM Climb-evaluation; <determine if own aircraft can climb>
    IF (radar altimeter data flagged as bad)
        THEN increment count of cycles of bad data;
        ELSE PERFORM Radar_Credibility_test;
            Clear or increment count of cycles of bad data;
    PERFORM Ground_level_estimation;
    PERFORM Ground_proximity_check;
    PERFORM Set_index;
    PERFORM Set_layer_dependent_parameters;
    PERFORM Update_interrogation_mode;
    PERFORM Update_advisory_mode;
    Set up fields for subsequent SLC Update messages to transponder;
END TRACK-OWN;
```

```
PROCESS Own_altitude_tracking;

    IF (system initialized)
        THEN clear initialized flag;
        IF (fine quantized altitude available)
            THEN initialize own altitude and rate (if avail.) to input data;
            ELSE CALL VERTICAL_TRACKING <for own altitude data>
                IN (initialization flag, valid-report flag,
                    altitude-reporting flag, own report, report time)
                INOUT (Nonlinear Track File Entry, Pointer to structure N);

        ELSE IF (fine quantized altitude data available)
            THEN estimate altitude and rate using smoothing equations;
            ELSE CALL VERTICAL_TRACKING <for own altitude data>
                IN (initialization flag, valid-report flag,
                    altitude-reporting flag, own report, report time)
                INOUT (Nonlinear Track File Entry, pointer to structure N);

END Own_altitude_tracking;
```

```
PROCESS Climb_evaluation;

    <To be supplied according to individual aircraft type and various
    airframe and Flight Management System inputs>

    IF (an RA has not yet been displayed)
        THEN CLEAR the climb inhibit indicators;
            <Once an RA has been displayed to the pilot, the climb inhibit
             and increase climb inhibit indicators must not be changed from
             cycle to cycle to prevent RA strength fluctuations.>
           IF (own altitude is above aircraft altitude limit specified by
               ARINC 735 program pins OR performance limit discrete indicates
              1500 fpm) own aircraft can't climb at
                THEN SET both climb inhibit indicators;
                ELSE IF (performance limit discrete indicates that own
                     aircraft can't climb at 2500 fpm)
                   THEN indicate that own aircraft can't increase climb;

END Climb_evaluation;
```

References: Config-Climb-Inhibit $_{s42}$ (↑), Increase-Climb-Inhibit $_{s43}$ (↑), Discrete $_{020}$, Climb-Inhibit $_{592}$ (↑), Increase-Climb-Inhibit $_{s58}$ (↑)

```
PROCESS Radar_credibility_test;

    Determine credibility of radar altimeter data;
    Set flag to indicate usability of data;

END Radar_credibility_test;
```

3.7.7 **References:** (1) Validity Check

```
PROCESS Ground_level_estimation;

    <Estimate ground level using radar altimeter data>

    IF (On-the-ground indication)
        THEN ground-alt altitude;
    ELSEIF (radar altimeter has been bad for a long time)
        THEN disable on-the-ground logic;
    ELSEIF (radar altimeter has only been bad for a short time)
        THEN; <keep using Previous estimate for ground level>
    ELSEIF (own was far above ground last cycle)
        THEN IF (radar altitude shows value low enough now)
            THEN ground-alt.alt. - radar altitude;
            ELSE;      <ground level unchanged>
    OTHERWISE IF (High enough now)
            THEN indicate own far above ground;
            ELSE ground-alt.alt. - radar altitude;

END Ground_level_estimation;
```

References: Ground- Level -394

```
PROCESS Ground_Proximity_check;
    IF (own aircraft is well-above ground level)
        THEN CLEAR DESCEND RAs prohibited flag;
             CLEAR flag that disables aural annunciations;
        ELSE determine altitude above ground;
            IF (DESCEND RAs prohibited)
                THEN IF (altitude above ground is above P.NODESHI)
                    THEN CLEAR DESCEND RAs prohibited flag;
                    ELSE; <still below no-descend altitude threshold>
                ELSE IF (altitude above ground is below P.NODESLO)
                    THEN SET DESCEND RAs prohibited flag;
                    ELSE; <still above no-descend altitude threshold>
            IF (aural annunciations are inhibited)
              XOLHAURALNI()altitude above ground is above P.
                    THEN CLEAR flag that disables aural annunciations;
                    ELSE; <still below no-aurals altitude threshold>
              XOEIASRALEO()altitude above ground is below P.
                    THEN SET flag that disables aural annunciations;
                    ELSE; <still above no-aurals altitude threshold>

END Ground_proximity_check;
```

s- , 452 (1) Other Aircraft Status 297

```
PROCESS Set_index;

    PERFORM Auto_SL; <Determine SL based on altitude of TCAS aircraft>
    Save Pilot-selected sensitivity level;
    IF (Pilot selected AUTOMATIC mode of operation)
        THEN set Pilot-selected SL to altitude-based SL;
    Initialize level to large integer;
    REPEAT UNTIL (all ground site registers processed);
        IF (ground site does not default to TCAS level selection and its level
            is the lowest level yet seen)
            THEN save that new level;
    ENDREPEAT; <this loop selects lowest Mode S site Sens. Level>
    IF (a ground site has sent a Sensitivity Level Command)
        THEN use the lesser of the ground site SL and pilot-selected SL;
    IF (externally-determined SL is not TA-only (SL2)
        THEN use the lesser of externally-determined SL and
            altitude-based SL;

END Set_index;
```

References:

```
PROCESS Auto_SL;

    IF (own aircraft is on the ground)
        THEN IF (traf. display allowed on ground and pilot has not selected
                Standby mode)
            THEN SL = 2;
            ELSE Select Standby mode;
    ELSEIF (own aircraft can't descend AND own aircraft can't climb)
        THEN SL = 2;
    ELSEIF (Old SL was 3 or less AND radar alt. data avail.)      <TCAS chooses
            level>
        THEN IF (above altitude threshold for SL 4)
                THEN SL = 4;
                ELSE SL = 2;
    ELSEIF (Old SL was 4)
        THEN IF (radar shows below altitude threshold for SL 2)
                THEN SL = 2;
            ELSEIF (radar shows above altitude threshold for SL 5)
                THEN SL = 5;
            ELSEIF (radar altimeter has been bad for long time AND
                   barometric altitude above threshold for SL 5)
                THEN SL = 5;
    ELSEIF (Old SL was 5 AND radar alt. avail.)
        THEN IF (below altitude threshold for SL 4)
                THEN SL = 4;
            ELSEIF (above altitude threshold for SL 6)
                THEN SL = 6;
    ELSEIF (Old SL was 6)
        THEN IF (below altitude threshold for SL 5)
                THEN SL = 5;
        ELSEIF (above altitude threshold for SL 7)
                THEN SL = 7;
    ELSEIF (Old SL was 7)
        THEN IF (below altitude threshold for SL 6)
                THEN SL = 6;
    ELSEIF (radar altimeter has been bad for a long time)
        THEN select default SL value;
END Auto_SL;
```

References:

```
PROCESS Set_layer_dependent_parameters;

  >NOTE: This process assigns an integer value to G.LA

    IF (G.LA EQ 1)
         THEN IF (Own tracked altitude GE Ist Upper hysteresis bound)
                   THEN 2 G.LA;

    IF (G.LA EQ 2)
         THEN IF (Own tracked altitude LE 2nd Lower hysteresis bound)
                   THEN 1 ;G.LA
              ELSE IF (Own tracked altitude GE 2nd Upper hysteresis bound)
                   THEN 3 G.LA;

    IF (G.LA EQ 3)
         THEN IF (Own tracked altitude LE 3rd Lower hysteresis bound)
                   THEN 2 G.LA;
              ELSE IF (Own tracked altitude GE 3rd Upper hysteresis bound)
                   THEN 4 G.LA;

    IF (G.LA EQ 4)
         THEN IF (Own tracked altitude LE 4th Lower hysteresis bound)
                   THEN 3 G.LA;

    Set Positive advisory altitude threshold to appropriate AL array element;
    Set Detection altitude threshold to appropriate TZ array element;
    Set Required separation assuming no vertical tracking error to
         appropriate SF array element;

  END Set_layer_dependent_parameters;
```

References: s-201(T) by altitude:

```
PROCESS Update_interrogation_mode;

    IF )(1TCAS operating AND sensitivity level GT
         THEN SET Interrogation_enabled flag;

         ELSE IF (Interrogation-enabled was set AND Resolution-enabled
                  was not set)
              THEN CLEAR Interrogation-enabled flag;
                   REPEAT WHILE (in coordination lock state);
                      <Loop while waiting for coordination lock state
                       to end.  Performance Monitor should recognize
                       when TCAS has been locked for more than P.TUNLOCK
                       seconds and take appropriate action.>
                   ENDREPEAT;
                   SET G.COLOCK using uninterruptible test
                             and set instruction;
                   Save lock time;
                   CLEAR pointers from threat file to track file;
                   Null the intruder track file;
                   CALL COORDINATION_UNLOCK;

    <If Resolution_enabled mode was set, Update_advisory_mode must
    perform cleanup this cycle.  Track file will be cleared next cycle.>

END Update_interrogation_mode;
```

```
PROCESS Update_advisory_mode;

    <Determine whether TCAS may generate RAs and TAs>

    CLEAR Traffic advisory flag;
    IF (TCAS operational)
        THEN IF (sensitivity level GE 2)
            THEN SET Traffic advisory flag;
                IF (sensitivity level EQ 2 AND (low-altitude TA option
                    not in effect)
                     THEN CLEAR Traffic advisory flag;
               IF (sensitivity level GT 2) AND (external SL GT 2)
                     THEN SET Resolution-enabled flag;
    IF NOT (TCAS operational) OR sensitivity level LE 2 OR external SL LE 2
        THEN IF (Resolution-enabled was set)
            THEN CLEAR Resolution-enabled flag;
                REPEAT WHILE (in coordination lock state);
                    <Loop while waiting for coordination lock state
                     to end.  Performance Monitor should recognize when
                     TCAS has been locked for more than P.TUNLOCK
                     seconds and take appropriate action.>
                ENDREPEAT;
                SET G.COLOCK using uninterruptible test
                             and set instruction;
                Save lock time;
                Add all threats to Working List with status 'Terminate';
                REPEAT (for each ITF entry);
                    Clear waiting-for-intent counter, hit counter,
                        firmness delay counter;
                    Clear display code;
                    Allow subsequent tau to be computed naturally;
                    Select next ITF entry;
                ENDREPEAT;
                CALL COORDINATION_UNLOCK;

END Update_advisory_mode;
```

References:

(1) RA-Inhibit

366

s' NTRUDERS is to generate an intruder The main purpose of Task TRACK-I tracked range, range rate, altitude, and alititude and store them in the ITF entry for that intruder. Range and range rate are generated via an alpha-beta tracker using the sequence of Surveillance range reports. Altitude and altitude rate are generated using a nonlinear vertical tracker (V ERTICAL_TRACKI G).

The first portion of TRACK_I NTRUDERS is a loop on each intruder receiving surveillance data (i.e. a loop on the entries in the Intruder Surveillance Buffer). The remaining portion is a loop on the ITF, where any intruder not receiving a surveillance report is deleted from the ITF. The cycle time is stored in each intruder's record when it is updated. Any intruder whose update time is older is deleted.

```
TASK TRACK_INTRUDERS;

  REPEAT WHILE (more intruders in surveillance buffer);
     IF (no cross-reference for this intruder)
          THEN PERFORM ITF_entry_creation;
               CALL VERTICAL_TRACKING
                    IN (initialization flag, valid-report flag,
                        altitude-reporting flag, own report, report time)
                    INOUT (Nonlinear Track File Entry, pointer to structure N);
               REPEAT WHILE (more position reports for this intruder);
                    PERFORM Position_tracking; <update range and altitude>
               ENDREPEAT;
          ELSE PERFORM Position_tracking; <update range and altitude>
     IF (aircraft reporting altitude)
          THEN copy data from nonlinear track file;
              <keep track file local to task>
               Clear sense-chosen-despite-bad-firmness indicator;
        ELSE IF (altitude-reporting intruder became non-altitude
                 reporting)
                THEN clear N entry contents and delete N entry;
                    IF (an RA is currently displayed for this intruder)
                         THEN SET "threat became non-altitude reporting" flag;
                    Set traffic code to no-display;
                    <Determination of display status will be made in
                     Traffic Advisory logic)>
              Null the nonlinear track file pointer;
                  <int. is non-altitude reporting>
```

```
    Update flag indicating altitude-reporting equipage;
    PERFORM Set_arrow;   <update vertical rate display arrow>
    PERFORM On_ground_test;
    Update intruder's TCAS equipage and Sensitivity Level;
    Select next intruder;
  ENDREPEAT;
  PERFORM Drop_tracks; <drop tracks not updated by surveillance>

END TRACK INTRUDERS;
```

[illegible] Reporting 1

```
PROCESS ITF_entry_creation;

    Create a track file entry;
    Initialize the fields therein;
    Set hit counter to zero;
    Set firmness delay count to zero;
    Clear display vertical rate arrow;
    Clear delete-intruder flag;
    Allow tau to be computed naturally;
    Clear flags associated with reversal
   /increase rate logic;
    Clear rising tau counter for nuisance alarm filter;
    Set traffic code to no-display;
    Initialize valid surveillance report indicator;
    Initialize indicators for annunciation of ‘clear of conflict’,
       ‘threat has become non-altitude reporting during RA’, and ‘threat’s
        track has been dropped by surveillance during RA’;
    Set ITF.LEV to the lowest value that can be used as an index by
        the Traffic Advisory logic;
         <This ensures that ITF.LEV is always defined>

END ITF_entry_creation;
```

```
PROCESS Position_tracking;

    CALL VERTICAL_TRACKING
       IN (initialization flag, valid-report flag, altitude-reporting
           flag, own report, report time)
       INOUT (Nonlinear Track File Entry, Pointer to structure N);
    Update time and time difference since last report;
    IF (valid range report received this cycle)
       THEN update range and range-rate estimates using alpha-beta smoothing;
       ELSE coast range using previous rate estimate;
          <range rate unchanged>
    Correct range and range rate when tracked range is less than zero;
    Save range coast flag;
    PERFORM Valid_report_test;

END Position_tracking;
```

References: (↑) Other-Tracked-Range f-434 (↑) Other-Tracked-Range-Rate f-435

```
PROCESS Valid_report_test;

    [illegible]determine if at least
   [illegible]s have been received within the last            If so, and if the
    current report is valid, then declaration of a TCAS-equipped intruder as
    a threat can occur.  Otherwise, threat declaration will be
    deferred pending better surveillance reports.

    IF (past validity sequence is [illegible]r 'III')
         [illegible]HEN remove the leading '    <S[illegible]tract '
             <No need to remove leading '0[illegible]f past validity sequence is
            [illegible]r '
    Left shift sequence one bit; <[illegible]ltiply by
        [illegible]sequence is '
    IF (valid range data exists this cycle)
         [illegible] sequence;
              <Sequence becomes [illegible]r '
  [illegible]If no valid range data exists this cycle, sequence remains '
    [illegible]r '

END Valid_report_test;
```

[illegible]

```
PROCESS Set_arrow;

    <Determine value to be displayed for vertical rate arrow if intruder
    later qualifies for traffic or resolution advisory>

    IF (aircraft not reporting altitude)
        THEN display no arrow;
    ELSEIF (no established vertical trend)
        <either track newly started, reports are oscillating, or level>
        THEN display no arrow;
    ELSEIF (there is an established trend)
        THEN display an arrow in the direction of the vertical rate;
    OTHERWISE; <retain previous value of arrow when tracker unsure of
        change>

END Set_arrow;
```

s- **References:** (s) Display-Arrow 226

```
PROCESS On_ground_test;

    <Determine if intruder's altitude indicates he is on the ground>

    IF (Intruder not reporting altitude)
        THEN intruder not on ground;
        ELSE IF (intruder ugged to be on ground) previously
            THEN IF (intruder now high enough above ground estimate)
                    THEN declare intruder not on ground;
                    ELSE; <still on ground>
            ELSE IF (intruder now near enough to ground estimate)
                    THEN declare intruder on ground;
    ELSE; <still not on ground>

END On_ground_test;
```

```
PROCESS Drop_tracks;

    REPEAT WHILE (in coordination lock state);
       <Loop while waiting for coordination lock state to end.
        Performance Monitor should recognize when TCAS has been
        locked for more than P.TUNLOCK seconds and take
        appropriate action.>
    ENDREPEAT;
    SET G.COLOCK using uninterruptible test
                 and set instruction;
    Save lock time;
    REPEAT WHILE (more entries in track file);
       IF (no surveillance report this cycle for this track)
         THEN find matching threat file entry;
             IF (no matching entry OR own has no Res. Adv. for this threat)
                THEN clear all variables and flags in the Nonlinear
                       track file and intruder track file entries and
                       delete the entries;
                ELSE flag ITF entry for drop status;
                       <after threat file cleared>
                    IF (RA currently being displayed for this intruder)
                        THEN SET "track dropped" flag;
         Select next ITF entry;
    ENDREPEAT;
    CALL COORDINATION_UNLOCK;

END Drop_tracks;
```

3.4.17 Dropped-Threat-Track- (1) **References:** Source:

The nonlinear vertical tracker VERTICAL_TRACKING is called for each altitude-reporting intruder and for own if own is tracking altitude quantized to 100 ft. This routine determines tracked altitude and tracked altitude rate, given the current Mode C altitude report (if it exists).

The altitude reports input to the vertical tracker are quantized in 100-ft increments. They are also received at discrete time intervals (nominally one second).

The vertical tracker not only estimates tracked altitude and tracked altitude rate, but it also evaluates the quality of its estimate (minimum, fair, good, or maximum). It also determines upper and lower bounds on tracked altitude rate, in effect quantifying the uncertainty in tracked rate. These bounds may be used in the detection and advisory selection logic to avoid declaring an intruder a threat and selecting sense when the vertical track is not reliable.

In the following, a *bin* refers to any possible value of an altitude report (quantized to the nearest 100 feet). A *bin crossing* is a change in the successive reports. The *bin-crossing time* is the time between successive bin crossings (in the same direction), e.g., reports of 800, 900, 900, 900, 1000 yiels a three-second bin crossing time for the 900-foot bin.

```
ROUTINE VERTICAL_TRACKING
    IN (initialization flag, valid-report flag, altitude-reporting flag,
     altitude report, report tine)
    INOUT (Nonlinear Track File Entry, pointer to structure N);

       <Using current and previous altitude reports, determine
        best estimate of intruder's altitude, alt. rate, bounds
        for rate, and our confidence in rate.>

    IF (Intruder is reporting altitude)
         THEN IF (this is first altitude report)
              THEN PERFORM Initialize_vertical_tracker;
              ELSE form altitude prediction;
                   Check credibility of report; <implied acceleration>
                   Clear spurious oscillation flag;
                   IF (report missing or non-credible)
                        THEN treat report as missing;
                             IF (valid report was received previous cycle)
                                  THEN save Pre-coast firmness;
                             Decrement firmness;
                             Use projected altitude;
```

```
                ELSE calculate bin difference from last report;
                     IF (no transition this time)
                          THEN PERFORM No_transition_update;
                               PERFORM No transition_firmrmss;
                          ELSE PERFORM Transition_update;

           Save track update time;
         ; 10Bin count at current trend cannot exceed
           IF ((valid report received) AND (no spurious oscillation))
                THEN save report time;

END VERTICAL_TRACKING;
```

```
PROCESS Initialize_vertical_tracker;

    Obtain an unused Nonlinear track file entry;

    Initialize vertical tracking variables;
    Treapkoerdt;altitude
    Tr=a0ked rate
    Eaxprgeect epdo bsiitni vcer ocsosn sttiamnet ;
    Dlieevcetl;ion indicators
    Rlaatreg ec opnofsi.d,emnoceg .l icmoinsstants;
    Fieirnmiiensum;
    T=ransition status   'start';
    Do not delay rate changes due to transition-time-guessed allowance;
    Save start time;

END Initialize_vertical_tracker;
```

```
PROCESS No_transition_update;

    Determine if there has been a long time since a bin cross transition;
    IF (too long since a transition)
            <false unless 'trend' since TBIN large otherwise>
         THEN reinitialize to level;
            Tracked altitude ← report;
            Tracked rate ← zero;
            Expected bin cross time ← positive constant;
            Transition status ← longago;
            Update upper and lower rate limits;
    ELSEIF (too long for current rate estimate)
         THEN slacken rate estimate;
            Set tracked altitude to bin boundary between report and
                expected bin;
            Reduce count of bins at current rate;
            Update upper and lower rate limits;
    OTHERWISE
         IF (transition status = 'guess')
            THEN decay rate and bin cross time estimates;
         IF (climbing)
            THEN ceiling tracked altitude at report + 50 ft;
         ELSEIF (descending)
            THEN floor tracked altitude at report - 50 ft;
         IF (transition status = longago' or 'guess')
            THEN update steep and slack rate limits;
         ELSEIF (trend AND transition status = transition overdue)
            THEN loosen slack rate limit;
         ELSEIF (transition status = 'start')
            THEN tighten steep and slack rate limits;

END No_transition_update;
```

```
PROCESS No_transition_fironess;

      <Update firmness after non-transition report>

    IF (Transition status = 'oscillation')
        THEN IF (one or more coasts occurred before this report)
            THEN set floor of firmness according to length of oscillating state;
          >2Otherwise, if no previous coasts, CASFIRM remains
             IF (oscillation state is old enough)
                 THEN firmness = maximum;
    ELSEIF (Transition status = 'guess' or 'start')
        THEN IF (time in this bin large enough to rule out steep rate)
             THEN firmness = fair;
        ELSEIF (time in this bin large enough to ensure low rate)
             THEN firmness = good;
        ELSEIF (time in this bin large enough for probable leveloff)
             THEN firmness = maximum;
        IF (Report follows coasts)
             THEN restore firmness to pre-coast value if higher;
    ELSEIF (Transition status = 'longago')
        THEN firmness = maximum;          <rate level>
    OTHERWISE <transition status = 'ended'>
             IF (transition very late)
                THEN increment firmness; <more confident of low rate>
                    IF (report follows coasts)
                       THEN restore firmness to precoast value if higher;
             ELSEIF (transition somewhat late)
                 THEN decrement firmness; <less confident of original rate>
             OTHERWISE IF (many reports so far this bin)
                      <transition not overdue>
                         THEN firmness = larger of 'good' or present level;
                      IF (Report follows coasts)
                         THEN restore firmness to precoast value if higher;

END No_transition_firmness;
```

```
PROCESS Transition_Update;

    Save direction of bin change and bin cross time;
    Calculate time in previous bin;

    IF (transition is in the expected direction)
        THEN compute actual - expected time in bin; <negative if early
                     transition>
             Compute predicted number of bin crossings based on last
                estimate of bin crossing time;
        IF (no good estimate exists OR transition significantly early)
            THEN calculate bound on vertical speed estimate;
                 eaxipmeectiend pbrienvicoruossbitni;me
                 Initialize bin-count-at-current-trend;
                 Tfriamcek0e0d rcartoess previous bin;
                j Ssaivdeu/a2lbound ary
                           tracked rate;
            ELSE update residual;
                IF (residual less than threshold)
                    }THbEiNn-uppdoantte-aBl-ETrMh,isBrEaTte;
                }ELSE alternate update BETA
                        gBivnesn cratedcuirfrfeont rate;
                Smooth expected bin cross time;
                Tfr/ea1k0e0xdpercatted bin cross time;
                sFeeiwdi/2bdoaundtaiyrnueate rate;
        PERFORM Transition_firmness;
        Transition status   'trend';
    ELSE PERFORM Unexpected_transition; <oscillation or level to rate>
 PERFORM Transition_time_and_bin; <determine time of last transition>
              <= now unless coasts>

END Transition_update;
```

```
PROCESS Transition_firmness;

      <Update firmness after transition report>

    IF (Transition status was 'oscillation' OR guess')
        THEN; good firmness
            Update steep and slack rate limits;
        ELSE PERFORM Transition_set_casfirm;
            <compare observed vs. actual bin cross time>
            PERFORM Transition_set_rate_limits;
            <compute steep, slack rate limits>

END Transition_firmness;
```

```
PROCESS Transition_set_casfirm;

      <consider ratio of observed to expected time in previous bin>
    IF (ratio very close to 1)
        THEN maximum firmness;
    ELSEIF (close to 1)
        THEN good firmness;
    ELSEIF (not far from 1)
        THEN fair firmness;
    OTHERWISE minimum firmness;

      <consider observed - expected time in previous bin>
    IF (within quantization limits for perfect tracking) <=5sec>
        THEN maximum firmness;
    ELSEIF (within 5sec. quantization)
        THEN larger of firmness 'good' and present value;
    OTHERWISE; <no jump upward for time quantization>

    <consider time in last bin>

    IF (large enough to rule out all but low rates)
        THEN larger of firmness 'good' and present value;
    ELSEIF (large enough to rule out steep rate)
        THEN larger of firmness 'fair' and present value;
    OTHERWISE;

END Transition_set_casfirm;
```

```
PROCESS Transition_set_rate_limits;

      <Estimate maximum and minimum bounds for vertical rate>

    IF (Firmness is maximum OR difference in expected, observed bincross
        time is small)
         THEN tighten steep, slack rate limits;
    ELSEIF (Ratio of observed to expected bin cross time is less than unity)
         THEN use alternate calculation for rate limits; <early transition>

    OTHERWISE <late transition>
         IF (coast this cycle)

              THEN estimate interval when being in this bin is
                        consistent with expectations;
                   IF (current time falls within that interval)
                        THEN firmness = good;
                        ELSE firmness = minimum;
                             Slack rate limit = 0;
                             Loosen steep rate limit;
              ELSE slack rate limit = 0;
                   Loosen steep rate limit;

END Transition_set_rate_limits;
```

```
PROCESS Unexpected_transition;

    <Transition observed with no trend or in different direction than trend>

    IF (no trend established OR more than I bin crossed)
         THEN firmness =Minimum;
              Transition status ='guess';
              Tracked rate =modest rate in now direction;
            ε Stamp/2 boundary rate;
             /Expected bin cross time;
              Initialize bin-count-at-this-rate, residual;
              Slack rate limit =Level;
              Steep rate limit =large value in now direction;
         ELSE <transition is oscillation, direction opposite prov. trans.>
              IF (Transition status AND oscil. so soon after
                  prior trend crossing may be due only to
                  really be continuing)
                  THEN firmness =Minimum;        <ignore report, treat as coast>
                       Loosen slack rate limit;
                     SAVE flag TTRAN prevent updating
                            in Transition-time-and-bin;
                  ELSE Tracked rate to previous bin;
                       Tracked rate =0;     <treat as true oscillation>
                       Expected bin cross time;
                       IF (previous transition also an oscillation)
                            THEN firmness =Excellent;
                                <well established oscillation>
                            ELSE firmness =Good;
                               Transition status ='oscillation';
                               Initialize start time of oscillation state;
                       Steep and slack rate limits =Zero;

END Unexpected_transition;
```

```
PROCESS Transition_time_and_bin;

        [illegible]sition.

    [illegible]on)
            <jitter near bin boundary>
        THEN save altitude report;
             Do not delay rate change next cycle due to
                transition-time-guessed allowance;
        IF (no coast last cycle)
            THEN update transition time;
            ELSE increase thresholds to delay rate change next cycle
                   due to transition-time-guessed allowance;
                IF (direction agrees with established rate)
                    THEN divide coasted interval equally among bins
                           crossed to estimate last transition time;
                         Time is no earlier than last valid report;
                    ELSE divide coasted interval equally among bins
                           crossed allowing for direction change;
        Update direction indicators;

END Transition_time_and_bin;
```

```
FUNCTION PM;

   <Determine the sign of the argument>

    If (argument is positive or zero)
      THEN PM := +1;
      ELSE PM := -1;

END PM;
```

Detection Logic

DETECT-CONFLICTS determines which intruders are predicted to be sufficiently close in range and altitude to require a resolution advisory. The logic loops on each intruder in the Intruder Track File. Using the tracked geometrical data as input, HIT-OR-MISS-TEST is called to perform tests on range and on altitude. If oonditions for a threat are passed, the detection logic is said to have declared a hit; the hit flag is set. If either test is not satisfied, an RA is normally not generated against the intruder, and the hit flag is cleared. If both tests are satisfied, TRACK-FIRM-TASK is called. If an altitude-crossing resolution advisory can be avoided or deferred, or the criteria for an latitude-crossing advisory warrant its immediate issuance, or of uncertainties in the intruders' rate (due to noise or confidence limits) are within tolerance thresholds, the hit flag is set. Otherwise, the hit flag is cleared, as the logic does not declare a hit when unsure of the critical conditions. Then SET-UP-WORKING-LIST is called, which adds all intruders that have been declared hits to the WORKING-LIST (a list used as input to the next portion of the TCAS logic, Task RESOLVE-CONFLICTS). At the conclusion of the Intruder Track File loop, TEST-FOR-MULTIAIRCRAFT-CONFLICT is called to determine whether a multiaircraft conflict exists.

```
TASK DETECT_CONFLICTS;

     REPEAT WHILE (more entries in Intruder Track File);
          IF (threat track marked for drop OR threat became non-altitude
                    reporting)
               THEN add track to Working List with status 'Terminate';
               ELSE IF (intruder is reporting altitude)
                    THEN PERFORM Set_datection_parameters;
                         PERFORM Hit_or_miss_test;
                              <determine which intruders need RAIS>
                         PERFORM Set_up_working_list;
                              <add then to Working List>
          Select next ITF entry;
     ENDREPEAT;

     PERFORM Test_for_multiaircraft_conflict;

END DETECT_CONFLICTS;
```

```
PROCESS Set detection_parameters;

     Determine sensitivity of own [illegible] levels;
          <for unequipped intruder, use own level>
     Set detection parameters for selected sensitivity level;
     Set tau parameters based on intruder equipage and selected sens. level;

END Set_detection_Parameters;
```

```
PROCESS Hit_or_miss_test;

      <Test for conflict criteria ('hit') >

     IF (Intruder is on the ground)
          THEN detection test fails;
               Enable firmness delay alarm;
               Clear the waiting-for-intent counter;
              IF (RA issued and hit occurred
              1 cycle ago, clear flag)
                    THEN SET ITF clear of conflict flag;
          ELSE calculate relative altitude and altitude rate;
               Calculate absolute value of rel. alt. and rate of divergence;
               PERFORM Range_test;
          IF (range divergence)
               THEN vertical miss distance = current relative altitude;
               ELSE CALL VERTICAL_MISS_DISTANCE_CALCULATION;
          IF (range test failed)
               THEN detection test fails;
                    IF (RA issued and hit occurred
                   12 cycles ago, or
                        cycle ago)
                         THEN SET ITF clear of conflict flag;
               ELSE IF (RA not previously issued) <Range test passed>
                         THEN PERFORM Altitude_test;
                             IF (Altitude test fails)
                                 THEN detection test fails;
                                 ELSE PERFORM Track_firmness_test;
           <both range and alt. tests passed but detection may still fail if
           bad firmness>
                         ELSE detection test passes;

   <Note: If an RA was previously issued for the current threat, and
    that threat is still converging, the RA status will be maintained
    even though own aircraft and the threat aircraft are separated by
    more than ITF.filter. The strength of the RA being maintained will
    be determined in Resolution and Coordination.  Later, when the
    threat begins to diverge in range, the RA will be dropped.>

END Hit_or_miss_test;
```

(↑) Threat-Specific Clear-Of-Conflict

Condition

```
PROCESS Range_test;

     IF (Range diverging more than minimal amount)
          THEN set tau's to minimum values;
               Allow tau to be computed with no ceiling next cycle;
               IF (range-rate product sufficiently large OR range
                    outside incremental protection volume)
                    THEN range test fails;
                         IF (intruder is not a threat)
                              THEN enable firmness delay alarm;
                                   Clear the waiting_for_intent counter;
                    ELSE range test passes;
          ELSE IF (Range rate is of small magnitude)
                    THEN adjust range rate to show slow convergence;
               PERFORM Tau_calculation;
               IF (modified tau within alarm limit)
                    THEN range test passes;
                    ELSE IF (range outside incremental protected volume OR
                         range rate product outside protected area)
                         THEN range test fails;
                         ELSE range test passes;
     IF (Range test passes AND RA has not yet been issued for this threat
               AND tau has been rising for past 3 seconds)
          THEN range test fails;

END Range_test;
```

(1) Threat Range-Test

References:

```
PROCESS Tau_calculation;

     Calculate two taus, one using a range offset;
     IF (tau ceiling in effect) <set after first pass through this process>
          THEN IF (no-offset tau declining)
                    THEN Set rising tau counter to zero;
                    ELSE IF (intruder range exceeds P.NAFRANGE miles)
                         THEN increment rising tau counter;
               Set two taus to lower of current or previous calculated values;
          ELSE set two taus to current values;
               Set rising tau counter to zero;
     Set two taus to zero if calculated value was negative;
SET tau ceiling;

END Tau_calculation;
```

[illegible] -420(↑) True-Tau-

Uncapped [illegible] Other-Pro -430

```
PROCESS Altitude_test;

     IF (current altitude within alarm volume)
          THEN IF (vertical miss distance within alarm volume)
               THEN altitude test passes;
               ELSE altitude test fails;

          ELSE <large current relative altitude>
               IF (altitude not converging)
                    THEN altitude test fails;
                   ELSE calculate vertical tau and pro
                   getRange at coaltitude
                         IF (vertical tau within alarm volume AND
                             (vertical miss distance within alarm volume OR
                             (range at coaltitude within prot. offset AND
                             time to coaltitude is less than range TAU value)))
                            THEN altitude test passes;
                            ELSE altitude test fails;

END Altitude_test;
```

References:

(1) The act: Alt-Test

```
PROCESS Track_firmness_test;

    <Range and alt. tests Passed.  Check firmness and delay 'hit' if time
    permits>

    IF (intruder is TCAS-equippod)
         THEN clear local vertical intent variable;
              IF (a threat file exists for this intruder)
                   THEN REPEAT WHILE (in coordination lock state);
                          <Loop while waiting for coordination lock state to
                          end.  Performance Monitor should recognize when
                          TCAS has been locked for more than P.TUNLOCK seconds
                          and take appropriate action.>
                        ENDREPEAT;
                        SET G.COLOCK using uninterruptible test and set
                                     instruction;
                        Save lock time;
                        Copy threat's intent value into a local variable;
                        CALL COORDINATION_UNLOCK;
              IF (a vertical intent exists against own aircraft)
                   THEN detection test passes;
                   ELSE IF (range reports are valid)
                        THEN IF (tau < threshold) for current SL
                             THEN detection test passes;
                                  PERFORM Avoid_TCAS_TCAS_crossings;
                                  IF (the detection test still passes)
                                       THEN PERFORM Alt_separation_test;
                             ELSE PERFORM Model_worst_rate_errors,
                       <Project climb/descent and intruder to CPA>
                                  PERFORM Evaluate_low_firmness_separation;
                       <determine if separation is adequate to choose sense>
              IF (detection test passes)
                  THEN clear waiting-for-intent counter;
         ELSE IF &(tau is less than threshold for current sens. level
                  firmness)
                  THEN detection test passes;
                       PERFORM Alt_soparation_test;
                  ELSE PERFORM Model_worst_rate_errors;
                       PERFORM Evaluate_low_firmness_separation;
```

```
END Track_firmness_test;
```

-Test

on-Test

Test

```
PROCESS Avoid_TCAS_TCAS_crossings;

    IF (no Threat File entry exists) <threat did not yet send intent>
        THEN IF (threat is non-level AND own aircraft is level)
            THEN IF ((own aircraft is more than P.MINSEP above AND
                      crossing is pro jected) OR (own aircraft is more
                     than P.MINSEP below AND crossing is pro jected))
                     THEN IF (own TCAS has not deferred its advisory
                              too long)
                             THEN increment waiting counter;
                                  detection test fails;

END Avoid_TCAS_TCAS_crossings;
```

(↑) TCAS-TCAS-Crossing-Test

```
PROCESS Alt_separation_test;

    IF (own aircraft is level AND is separated vertically from the
       intruder by more than P.MAXALTDIFF ft)
         THEN CLEAR selected_sense flags;
            CALL MODEL_MANEUVERS
                OUT (predicted separation for climb, descend)
    IF (climb separation greater than descend)
         THEN select 'climb' sense;
            ELSEIF (own cannot climb AND climb maneuver not much
                    worse than descend)
                THEN select 'climb' sense;
            OTHERWISE select 'descend' sense;
            IF (own aircraft is at least P.MINSEP above threat)
                THEN IF (descend sense has been chosen AND
                        climb sense provides at least ALIN ft)
                    THEN select climb sense;
            IF (own aircraft is at least P.MINSEP below threat)
                THEN IF (climb sense has been chosen AND
                        descend sense provides at least ALIN ft)
                    THEN select descend sense;
          IF (a crossing RA would still be selected against an
              intruder that is separated vertically by more than
              P.MAXALTDIFF ft)
                THEN the detection test fails;

END Alt_separation_test;
```

[illegible] Separation-Test -39

```
PROCESS Model_worst_rate_errors;

   <Compare with limits of threat's possible
    altitude over entire range of alt. rate uncertainty.  Decide
    if either RA for own suffices vs. worst case threat rate.>

    Set subroutine parameter thresholds;

    Determine altitude for own climb, if capable;
    <model level if own can't climb or already has 'Descend'>
    Determine altitude for own descent;
    <model level if own already has 'Climb'>
    Determine altitude for threat at his largest positive vertical rate;
    Determine altitude for threat at his most negative vertical rate;
    Determine worst-case separation for own climb and descend;

END Model_worst_rate_errors;
```

```
PROCESS Evaluate_low_firmness_separation;

   IF (neither sense gives adequate separation)
       THEN detection test fails;
            Increment firmness-delay cycle count;
       ELSE detection test passes;
            IF (climb sense gives better separation)
                THEN indicate climb sense OK;
                     Save second-choice separation; <for Don't Care>
                ELSE IF (cannot climb but don't descend nearly as
                           good as descend)
                         THEN indicate climb sense OK;
                             Save second-choice separation; <for Don't Care>
        <this forces don't descend. Bias against altitude crossing>
                         ELSE indicate descend sense OK;
                             Save second-choice separation; <for Don't Care>
            IF ((climb sense selected and intruder is more than
                      P.LOWFIRM above own aircraft)
                 OR (descend sense selected and intruder is more than
                      P.LOWFIRM below own aircraft))
                 THEN detection test fails; <crossings not permitted on low
                                              firmness>
                      Increment firmness-delay cycle counter;
                      CLEAR Bad firmness OK indicator;
                      CLEAR Second-choice separation;

END Evaluate low_fironess_separation;
```

359 **References:** Separation-Test

```
PROCESS Set_up_working_list;

    IF ('hit' declared)
        THEN indicate highest display priority;
            IF (new threat)
                THEN add to Working List with status 'new';
                ELSE add to Working List with status 'continuing';
            Set hit counter to indicate established threat;
        ELSE indicate no display; <may be overridden by Traffic Adv. logic>
            IF (2 consecutive miss)
                THEN IF (2 cycles ago) hit, miss
                    THEN IF (advisory has been displayed at least minimum time)
                        THEN add to Working List with status 'terminated';
                            Set hit counter to zero;
                        ELSE add to Working List with status 'continuing';
                            Indicate highest display priority;
                            CLEAR ITF clear of conflict flag;
                              <Don't announce "clear of traffic" yet>
                    ELSE add to Working List with status 'continuing';<keeps RA>
                        Set hit counter to show first 'miss';
                        Indicate highest display priority;

END Set_up_working_list;
```

36th References: (1) Source: Threat-Clear-Of-Conflict

```
PROCESS Test_for_multiaircraft_conflict;

    CLEAR multiaircraft flag and threat counter;
    REPEAT WHILE (more entries on Working List);
        IF (status is 'now' or 'continuing')
            THEN increment count;
        Select next Working List entry;
    ENDREPEAT;

    IF (more than one threat counted)
        THEN SET multiaircraft flag;

END Test_for_multiaircraft_conflict;
```

```
ROUTINE VERTICAL_MISS_DISTANCE_CALCULATION

  IN (Rel. alt., rel. alt. rate, start time, end time, clip time)
  OUT (vertical miss distance);

     Calculate VMD at end of critical interval using lesser of
         clip time or start time;
     Calculate VMD at start of critical interval using lesser of
         clip time or end time;
     IF (signs of two VMD's differ) <altitude cross during critical interval>
         THEN return VMD TBD;
     ELSEIF (own aircraft above throughout interval)
         THEN return lesser of two VMD'S;
     OTHERWISE return greater of two VMD'S;

END VERTICAL_MISS_DISTANCE_CALCULATION;
```

f Distance (f) Vertical Miss- References: -426

Resolution Advisory Logic

THE RESOLUTION ADVISORY determines what resolution advisories should be posted against the intruders designated as threats in the previous task, DETECT. For TCAS-equipped intruders, coordination is required to assure compatible maneuvers.

The [illegible] entry on the [illegible] (Working List structure) consists of threats, as well as intruders whose status as threats is to be terminated this cycle. New threats (intruders requiring an RA for the first time) are identified as are previously established (continuing) threats. In order to process multiple threats, the Working List is sorted so that terminal threats precede new threats, which precede continuing threats.

[illegible] on the CORRIDOR each entry in turn. The logic performs as follows throughout the life of a single threat. On the threat's first cycle (status of new), a Threat File (TF) entry will be created, and a sense (CLIMB or DESCEND) chosen. If the threat is TCAS-equipped and has chosen a sense versus own, own must choose the complementary sense. If a TCAS threat has not chosen a sense, or if the threat is unequipped, own selects the sense that provides best vertical separation at closest approach. Then a particular resolution advisory in the chosen sense is selected—generally the weakest one that gives adequate separation. The Threat File variables are updated. if the threat is TCAS II or III equipped, own sends a TCAS Resolution Message to the threat. The message contains a Resolution Advisory Complement (DON'T CLIMB or DON'T DESCEND) according to the sense own has selected. This completes the task for the threat's first cycle with an RA.

On each subsequent cycle, when the threat is a continuing threat, own reevaluates the particular sense and strength of the resolution advisory chosen. The sense may be reversed or the advisory may be strengthened or weakened. Again the Threat File variables are updated. If the threat is equipped, own informs it that an RA is still in effect.

Finally, when the threat is a terminal threat, own deletes the Threat File entry for the intruder (no longer a threat), unless the intruder is equipped and is still sending RA Complements. In that case, the Threat File entry remains until intruder no longer considers own a threat and sends a message removing the RA Complement.

An equipped intruder is normally behaving towards own very much like own is behaving towards it. Typically, each passes the other's range and altitude tests over roughly the same interval of time. The Threat File entry updated in [illegible]es also to store RA Complement received from the Intruder. The Threat File has timers for own last RA refresh and Intruder's RA Complement refresh.

```
TASK RESOLUTION_AND_COORDINATION
   IN (WL entry);

      Set up pointers to ITF and TF using Working List pointer entry;
      REPEAT WHILE (in coordination lock state);
        <Loop while waiting for coordination lock state to and.  Performance
         Monitor should recognize when TCAS has been locked for more than
         P.TUNLOCK seconds and take appropriate action.>
      ENDREPEAT;
      SET G.COLOCK using uninterruptible test and set instruction;
      Save lock time;

      IF (threat 'terminate' status)

          THEN PERFORM Update_threat_file_Own; <delete TF entry if no
                                threat intent>
              CALL DELETE_RESOLUTION_ADVISOR   <remove RA unless
                                                 needed for other threat>
              Clear flags for increase/reversal;
      ELSEIF (status = 'new')
          THEN Indicate vertical Resolution Advisory about to be chosen;
               PERFORM New_threat_file_entry;
               PERFORM Select_sense;
      OTHERWISE save previous cycle's advisory; <status = 'continuing'>
      IF (status is 'now' or 'continuing')
          THEN PERFORM Process_new_or_continuing_threat;
   IF (threat status 'terminate' OR display of RA has not been deferred)
          THEN Send Coordination Update message to transponder;
   IF (own TCAS is operational AND threat is TCAS-equipped)
          THEN PERFORM Send_initial_intent;
   CALL COORDINATION_ UNLOCK;
   IF (own TCAS is operational AND threat is TCAS-equipped)
          THEN PERFORM Complete-send-intent;

END RESOLUTION_AND_COORDINATION;
```

[3][2][1] Update (R) Coordination- References:

```
PROCESS Update_threat_file_own;

   IF (threat's) terminate status
        THEN save previous advisory selection;
             Save pointer to own resolution advisory;
             IF (threat still responsible for an advisory complement)
                  THEN CLEAR pointers to own RA list;
                       Update refresh time for own advisory;
                  ELSE Null the ITF Threat File pointer;
                       Clear all variables and flags in the threat file
                         entry and delete the entry;
        ELSE IF (advisory selection changed from last cycle)
                  THEN save latest advisory selection and time of
                          advisory change;
             Save pointers for old and now RA; <args. in call to
                   RESOLUTION_UPDATE>
             Update refresh time for own advisory;

END Update_threat_file_own;
```

```
PROCESS New_threat_file_entry;

    <this Process initializes existing or creates new threat file entry>

    IF (intruder is Mode S-equipped)
        THEN Search for threat file entry with same discrete address;
    IF (matching entry not found)
        THEN create new threat file entry;
             Save threat's Mode S ID, if any, in TF entry;
             Set threat intent refresh timer to initial negative value;
                <indicates no intent received>
             Clear all flags and variables;
             Clear advisory bit strings;
                <Indicates no advisory present>
    Save TF back pointer to ITF;
    Set own RA change and refresh timers to current time;
    Indicate new threat for display;
    Initialize flags and variables for reversal
    Save TF pointer in ITF;

END New_threat_file_entry;
```

```
PROCESS Select_sense;

   IF (Threat is TCAS equipped AND has sent a vertical complement)
        THEN PERFORM Form_complement; <Select compatible sense>
        ELSEIF (detection test did not select low-firmness sense) <normal case>
                  THEN CALL MODEL_MANEUVERS
                         OUT (predicted separation for climb, descand);
                       IF (climb separation greater than descend)
                            THEN select 'climb' sense;
                       ELSEIF (own cannot climb AND climb maneuver not
                               much worse than descend)
                            THEN select 'climb' sense;
                       OTHERWISE select 'descend' sense;
                       IF (own aircraft is at least P.MINSEP above threat)
                            THEN IF (descend sense has been chosen AND
                                     climb sense provides at least ALIM ft)
                                      THEN select climb sense;
                                           Store second choice for
                                              Don't Care test;
                       IF (own aircraft is at least P.MINSEP below threat)
                            THEN IF (climb sense has been chosen AND
                                     descend sense provides at least ALIM ft)
                                      THEN select descend sense;
                                           Store second choice for
                                              Don't Care test;
        ELSEIF (detection selected 'climb' despite low firmness)
             THEN select 'climb' sense;
                  Store second choice for Don't Care test;
        OTHERWISE select 'descend' sense; <low-firmness descend OK>
                  Store second choice for Don't Care test;
        IF (threat is TCAS equipped)
             THEN PERFORM TCAS_threat_processing;
             ELSE PERFORM Don't_care_test; <see if either sense acceptable>

END Select_sense;
```

[illegible] References:

[illegible]

```
PROCESS Form_complement;

     Derive own sense from threat's complement;
     IF (threat told own TCAS not to climb)
         THEN select descend sense;
         ELSE select climb sense;

END Form_complement;
```

```
PROCESS TCAS_threat_processing;

     Indicate "do care" situation; <use selected sense>
     IF (own Mode S ID is higher value than intruder's)
          THEN SET flag to defer display of RA;
               <Potential for incompatible senses exists>
               Change traffic display status from RA to TA;
               Save OWNTENT in temporary Threat File storage;
               CLEAR own RA variables;

END TCAS_threat_Processing;
```

DE(1) Delayed Advisory (1) References:

```
PROCESS Don't_care_test;

   <WL=threat whose WL entry is input to task>
   <TF=threat examined in loop below>
   <DC=``Don't Care''>

   IF (either sense provides adequate separation)
      THEN SET Don't-care flag for WL threat;
      ELSE CLEAR Don't_care flag for WL threat;
          IF (own Resolution advisories show a Positive in
              second-choice sense>
               THEN calculate own altitude following a leveloff;

                    REPEAT WHILE (more entries in threat file AND
                                  DC flag for WL threat not set);
                         IF (resolution against TF threat shows Positive
                             in same sense as second choice for WL threat)
                              THEN calc. altitude relative to TF threat
                                      and time for leveloff;
                                <result of 'do care' for WL threat>
                                   CALL VERTICAL_MISS_DISTANCE_CALCULATION
                                      IN (rel alt, rel vert rate, start
                                          time(WL threat), end time
                                          (WL threat), clip time(WL threat));
                                   IF (sep with leveloff vs. TF threat
                                        less than that for second choice
                                        maneuver vs. WL threat)
                                         THEN SET Don't-care flag for the
                                                  WL threat;
                                             <allow second choice sense>
                         Select next threat file entry;
                     ENDREPEAT;

END Don't_care_test;
```

References: 357 Don't-care Test

```
PROCESS Process_now_or_continuing_threat;

   IF (status is 'continuing' AND threat is established)
       THEN PERFORM Reversal_check;
   IF (RA display is not deferred)
       THEN PERFORM Soloct_advisory;
           IF (multiple threats this cycle)
               THEN PERFORM multiaircraft_processing;
               ELSE PERFORM Update_threat_file_own;
                    CALL RESOLUTION_UPDATE
                       IN (OLDPOI, OPTR);
           IF (status is 'continuing' AND threat is established)
               THEN PERFORM Increase_ check;
       ELSE update refresh timor for own advisory;

END Process_now_Or_continuing_threat;
```

```
PROCESS Reversal_check;

   CLEAR flag to consider an increase rate RA;
   IF (intruder is TCAS-equipped)
        THEN IF (own Mods S ID is higher)
                  THEN IF (RA display has been deferred)
                            THEN set OWNTENT to temporarily-saved RA;
                                 Increment the deferral cycle counter;
             IF (threat has selected same sense as own)
                  THEN PERFORM Form_Complement; <Reverse own sense>
                       Indicate that previous intent must be cancelled;
                       <Using reversal indication flag>
             IF ((RA display was deferred AND (either own TCAS has waited
                  more than P.WTTHR seconds OR threat has sent its intent))
                  OR (sense reversal occurred above))
                  THEN PERFORM Set_up_for_advisory;
                       CLEAR flag to defer RA display;
             IF (RA display is still deferred)
                  THEN change traffic display status from RA to TA;

        ELSE IF (P.MIN_RI_TIME sec. or more remain AND range TAU did not
                  start rising when the threat was more than P.NAFRANGE
                  miles away)
                  THEN IF (no inc. rate RA has been issued
                            AND current RA is crossing)
                           THEN Calculate CPA time using threat's pro
                                   INT;
                                PERFORM Reversal_pro
                                check;
                                IF (reversal not selected AND time to CPA
                                    is not sufficient for reversal against
                                    threat which may be close in altitude)
                                     THEN SET flag to consider increase
                                            rate RA;
                            ELSE PERFORM Cross_through_check;

END Reversal_check;
```

742 (4) Display Advisory References:

```
PROCESS Set_up_for_advisory;

     CLEAR OWNTENT(
     Save new advisory in TF.PERMTENT;
     Save time of new advisory in ITF.TCMD;
     IF (threat is not TCAS-equipped)
          THEN PERFORM Dont_care_test;

END Set_up_for_advisory;
```

```
PROCESS Reversal_pro

   IF (own and intruder have not yet crossed altitudes AND
         (either they are separated by at least P.AVEVALT ft OR
         at least P.HINRVSTIME seconds remain to CPA))
        THEN IF (intruder causing altitude crossing RA)
                  THEN IF (int's proi.alt. at CPA passed own's alt.
                          in RA direction)
                       THEN SET ITF reversal flag;
             ELSEIF (own aircraft is causing altitude crossing RA)
                 THEN IF (intruder now pro
                          own's altitude)
                       THEN CLEAR own causing crossing flag;
                            SET intruder causing crossing flag;
                       ELSE IF ((not multiaircraft situation) OR
                                (intruder is not level AND rates
                                are opposite in sign))
                       /* Prevent unnecessary reversals in multi-A
                            THEN CALL MODEL_MANEUVERS
                                   IN (ITF entry)
                                   OUT (prod. sep. for current situation);
                                 IF (pred. sep for cim is better than des)
                                      THEN IF (prov. sense was des. AND
                                               clm better than des)
                                            THEN SET ITF reversal flag;
                                 ELSEIF (clm. inhibited and des.
                                         better than biased climb)
                                      THEN IF (prov. sense was clm. AND
                                            des better than biased clm)
                                            THEN SET ITF reversal flag;
                                 OTHERWISE IF (prev. sense was clm. AND
                                               des better than clm)
                                            THEN SET ITF reversal flag;
          OTHERWISE;
          IF (a reversal needs to be considered)
                THEN PERFORM Reversal_modeling;

END Reversal_pro
```

```
PROCESS Reversal_modeling;

   CALL MODEL_MANEUVERS
     IN (ITF entry)
     OUT (predicted separation for sense reversal);
   IF (climb previously selected and predicted separation for descend
         is positive)
       THEN second choice is climb;
            Select descend sense;
            PERFORM Set_up_for_advisory;
   ELSEIF (descend previously selected and predicted separation for climb
           is positive)
       THEN second choice is descend;
            Select climb sense;
            PERFORM Set_up_for_advisory;
   OTHERWISE CLEAR reversal flag in ITF;

END Reversal_modeling;
```

```
PROCESS Cross_through_check;

    IF (encounter is non-altitude crossing)
         THEN IF ((RA is climb sense AND intruder is more than P.CROSSTHR
                    ft above) OR (RA is descend sense AND intruder is
                    more than P.CROSSTHR ft below))
              THEN CLEAR ITF.INCREASE;
                   CLEAR ITF.INC_ENC;
                   Initialize time of increase;
                   Complement current RA sense;
                   SET ITF.REVERSE;
                   PERFORM Set_up_for_advisory;

END Cross_through_check;
```

References: 00-Ft-Crossing

```
PROCESS Select_advisory;

   IF ((hit counter shows one miss) OR (no time to change maneuver))
        THEN select previous advisory; <which is working well>
        ELSE IF (threat has vertical rate close to level)
                  THEN ECHO; CALL CHECK_PRO
                       IF (current relative altitude outside threshold)
                            THEN PERFORM Try_vsl; <select weakest safe
                               advisory or positive if nothing else safe>
                            ELSE ECHO; CALL CHECK_PRO
                                 IF (VMD outside positive advisory threshold
                                     OR own has a substantial vertical rate)
                                      THEN PERFORM Try_vsl;
                                      ELSE select positive advisory;
                  ELSE ECHO; CALL CHECK_PRO
                       IF (VMD outside positive advisory threshold OR
                           own has a substantial vertical rate)
                            THEN PERFORM Try_vsl;
                            ELSE select positive advisory;
           PERFORM No_weaken_test;
           PERFORM Extreme_altitude_check;

END Select_advisory;
```

[illegible]

```
PROCESS Try_vsl;

   IF (range diverging)
        THEN IF (status is new')
                  THEN select negative advisory;
                       <Select_advisory has checked that pos. not necessary>
                  ELSE save previous advisory; <which is working>
        ELSE CALL VSL_OVER_INTERVAL (2000 fpm);
             IF (VSL suffices)
                  THEN select VSL 2000;
                  ELSE CALL VSL_OVER_INTERVAL (1000 fpm);
                       IF (VSL suffices)
                            THEN select VSL 1000;
                            ELSE CALL VSL_OVER_INTERVAL (500 fpm);
                                 IF (VSL suffices)
                                      THEN select VSL 500;
                                      ELSE CALL VSL_OVER_INTERVAL (0 fpm);
                                           IF (VSL suffices) <negative
                                               in this case>
                                              THEN select negative advisory;
                                              ELSE select positive advisory;

END Try_vsl;
```

s-72 (T) RA Strength

References:

```
PROCESS No_weaken_test;

     <Special tests which don't allow an advisory to weaken or strengthen
     from the one previously selected>

   IF (status is not new)
        THEN IF (reversal situation)
                  THEN use shorter no-weaken time;
                  ELSE use standard no-weaken time;
             IF (selected advisory is same or stronger)
                  THEN IF (tau shows sufficient time to wait for
                           improved firmness)
                            THEN keep previous advisory;
                       <stronger RA may be overreaction>
             ELSEIF (Previous advisory was positive) <and is weakening>
                  [illegible] CALL CHECK_PRO
                       IF (relative altitude is still inside positive
                           alarm thresh)
                            THEN keep previous advisory;
                  <keep stronger RA displayed until flown clear of conflict>
                            ELSE IF (last advisory not on long enough
                                      OR low track firmness)
                                 THEN keep previous advisory;
             OTHERWISE IF (last advisory not on long enough OR low firmness)
                            THEN keep previous advisory;

END No_weaken_test;
```

[illegible]

[illegible] Weaken-Positive

```
PROCESS Extreme_altitude_check;

   IF (positive CLIMB selected)
        THEN IF (own aircraft can't climb AND reversal not selected)
             THEN convert advisory to DON'T DESCEND;

   ELSEIF (Positive DESCEND selected)
        THEN IF (altitude above ground is below threshold where
                 descent allowed)
             THEN convert advisory to DON'T CLIMB;
                  CLEAR ITF reversal flag;
                  <Reversal Only announced with Positive DESCEND>

END Extreme_altitude_Check;
```

322 (L) Ext-Alt-Check
References:

```
PERFORM Multiaircraft_resolution;

   PERFORM Multiaircraft_threat_file_update; <create list of old, new advs.>
   REPEAT WHILE (more advisories to revise);
       CALL RESOLUTION_UPDATE;   <delete, add new RA's>
   ENDREPEAT;

END Multiaircraft_processing;
```

[illegible] **References:** (1) RA Strength

```
PROCESS Multiaircraft_resolution;

CLEAR flags indicating if any aircraft above' or 'below';

   <Above and below refer to sense of resolution, not current altitude.
    Following logic loops through the entire threat file. "TF
    threat' referred to by the TF entry being examined in the
    loop. "Current threat' was the input
    parameter to this task.>

   REPEAT WHILE (more threat file entries);
        IF (resolution already selected for TF threat)
             THEN save sense versus TF threat;
        ELSEIF (TF threat is same threat as current threat)
             THEN save sense versus current threat;
        OTHERWISE no further processing for this threat;
             <haven't selected resolution yet or TF entry due to intent only>
        IF (sense has been saved)
             THEN IF (don't_care flag set for TF threat)
                       THEN no further processing for TF threat;
                       <threat is new with no sense commitment>
                       ELSE IF (sense is 'descend')
                            THEN indicate threat 'above';
             <own committed to descend sense for at least one threat>
                            ELSE indicate threat 'below';
             <own committed to climb sense for at least one threat>
        Select next threat file entry;
   ENDREPEAT;

END Multiaircraft_resolution;
```

References:

```
PROCESS Multiaircraft_threat_file_update;

   <NoDCon't care".            See preceding page for other terminology>

   Save resolution for current threat;
  ;=0Number of RA changes to record in TF
   IF (own committed to at least one climb sense and descend sense)
        THEN SET positive-to-negative flag;
        ELSE CLEAR positive-to-negative flag; <change negatives to positive>
             IF (no commitments in either sense) <all DC cases or
                   intent-only TF entry>
                  THEN use sense of current threat;
                       <arbitrary in this case>
                  ELSE use sense appropriate to threat(s) found;
                       <commitment in only one sense>
   PERFORM Multiaircraft_loop_on_threat_file;
   Restore RA for current threat <OWNTENT> to value it had upon entry;

END Multiaircraft_threat_file_update;
```

```
PROCESS Multiaircraft_loop_on_threat_file;

   REPEAT WHILE (more threats in threat file); <local loop on entire TF>
     IF (advisory is vertical type)  <Provision for future horizontal type>
          THEN IF (RA is positive and positive-to-negative flag set OR
                    VSL and negative-to-positive indicated)
                   THEN change advisory as indicated by flag;
                     <change negative to positive and clear vsl's if flag
                     cleared, change positive to negative if flag set>
               IF (all threats are to have same sense AND saved sense for
                   current threat differs from that for TF threat)
                    THEN reverse sense; <can't happen to estab. sense,
                                         only to new DC tyoe>
               IF ((converted from negative to positive RA) AND
                  ((climb sense selected but CLIMB RAs are inhibited)
                  OR (descend sense selected but DESCEND RAs are inhibited)))
                    THEN convert the positive RA back to negative;
               IF (RA for TF threat now differs from its RA on previous cycle)
                    THEN increment number of RA changes to record in
                           Threat File;
                         Set pointers to old and new RA;
                            <compare to Proc. Update_threat_file_Own>
                         IF (TF threat is in Working List)
                              THEN update its time of advisory change;
               IF (TF throat is in Working List)
                    THEN update its time of refresh;
     Select next threat file entry;
   ENDREPEAT;

END Multiaircraft_loop_threat_file;
```

```
PROCESS Increase_check;

   IF (current RA is positive CLIMB or DESCEND AND has been displayed for
       at least 2 cycles)
        THEN IF (range TAU did not start rising when the threat was more
                 than P.NAFRANGE miles away AND (int. is TCAS-equipped OR
                 inc. rate RA should be considered in cross situation OR RA
                 is not altitude crossing))
                  THEN CLEAR 'within inc. rate RA threshold' flag;
                       IF (increase should be considered in crossing enc.)
                            THEN CLEAR flag to consider increase rate RA;
                                 j Calculate diff. of own alt. and int pro alt.
                                  IF (des RA AND alt.diff <= P.AVEVALT ft)
                                       THEN SET "in threshold"  flag;
                                  ELSEIF (-P.AVEVALT ft) =< des RA AND alt.diff. >
                                       THEN SET 'in threshold'  flag;
                            jELSE check; PERFORM Increase_pro
                       IF (min. sep. over critical interval < threshold)
                            THEN IF ((inc. RA issued) OR
                                  (time to CPA is > P.MIN-RI-TIME))
                                       THEN IF ((RA sense is descend AND
                                            own aircraft alt. < PZO-INCDES ft)
                                            OR (RA sense is climb AND inc.
                                            climb RAs inhibited))
                                                 THEN; <No inc. rate RA>
                                                 ELSE Int. time = current time;
                                                      IF (first time for
                                                          increase)
                                                         THEN save time of RA;
                                                      SET ITF.INCREASE,
                                                          ITF.INC_ENC;
             IF (increase rate RA was issued AND has been displayed for
                 at least P.TNOWEAK seconds)
                   THEN CLEAR ITF.INCREASE;
                        Initialize time of increase;
   ELSE CLEAR ITF.INCREASE; <RA not positive>
        Initialize time of increase;

END Increase_check;
```

312D(↑) 321V(↑)(↑) Coherence Increase -440

```
PROCESS Increase_pro

   IF (tracker firmness is at least at a specified level
        AND Tau LE threshold for this logic for current SL)
      THEN IF (RA is climb sense)
              THEN use larger of own alt.rate or nominal climb rate;
              ELSE (use lesser of own alt.rate or nominal descent rate;
           Calculate relative altitude and altitude rate;
           CALL VERTICAL_MISS_DISTANCE_CALCULATION;
           CALL CHECK_PRO [illegible]

[illegible] Increase_pro
```

[illegible] -440

```
PROCESS Send_initial_intent;

    <This Process begins transmission of intent to the threat TCAS.>
    <Form TCAS Resolution Message>

   Form complement of own resolution intent;
   IF (status is 'terminate,)
       THEN insert complement in deletion field of message;
       ELSE insert complement in addition field of message;
           IF (sense has been reversed)
           THEN cancel previous RA's complement;
   Set up parity field in Resolution message;
     <Message bits for VSB subfield are set equal to the Kth entry
     in the parity table to confirm the validity of the RA message.
    (K is the value of message bits 46 through      CVC, VRC)>
   Send initial TCAS Resolution (Intent) message to threat;

END Send_initial_intent;
```

References: (↑) CVC -421 (↑) Resolution-Message -42 (↑) VRC -388 -42

```
PROCESS Complete_send_intent;

   <This process completes the transmission of
    own's intent to the threat TCAS.>

   Wait until surveillance delivers reply or decides no reply
          for initial intent message;
   IF (coordination reply message not received)
        THEN CLEAR SUCCESS flag;
             REPEAT UNTIL (Successful msg transfer or too many no-replies)
                  Send TCAS Resolution (Intent) massage to threat;
                  Wait until surveillance delivers reply or decides no reply;
                  IF (Coordination reply message is received)
                       THEN SET SUCCESS flag;
                       ELSE Increment no-reply counter;
   ENDREPEAT;

END Complete_send_intent;
```

421 **References:**

(1) Resolution-Message

```
ROUTINE CHECK_PRO

  IN (position, threshold, own advisory)
  OUT (within-threshold flag);
       <set flag if position on unsafe side of threshold>

    CLEAR within-threshold flag;
    IF (advisory sense is climb')
        THEN IF (position is less than threshold)
                THEN SET within-threshold flag;
        ELSE IF (position is greater than negative of threshold)
                THEN SET within_threshold flag;

END CHECK_PRO
```

```
ROUTINE COORDINATION_UNLOCK

    End coordination lock state;
    IF (any messages in Resolution message queue)
    THEN CALL RESOLUTION_MESSAGE_PROCESSING;
        END COORDINATION_UNLOCK;
```

```
YROUTINE DELETE_RESOLUTION_ADVISOR
    IN (pointer to Resolution Advisory); <for this threat>

     IF (Resolution Advisory to delete is not null)
          THEN REPEAT WHILE (more entries in Threat File AND match not found);
               Search for another Threat File entry causing this
                  Resolution Advisory;
               ENDREPEAT;

               IF (Matching entry found)
                   THEN; <cannot delete Res. Adv. that applies to another
                          threat>
                   ELSE CLEAR Res. Advisory;

YEND DELETE_RESOLUTION_ADVISOR
```

```
ROUTINE MODEL_MANEUVERS
   IN (ITF entry)
   OUT (predicted separation for climb, descend);

     Model with true tau unless it exceeds a parameter value;
     Model with tailchase modified tau as long as it is within limits;
     PERFORM Set_up_maneuvers;
     IF (time permits own to maneuver)
          THEN PERFORM Modeling_calculations;

END MODEL_MANEUVERS;
```

```
PROCESS Set_up_maneuvers;

   Use nominal delay ties for non-reversal situations;
   IF (reversal situation)
        THEN use shorter delay time;
   IF (time to go less than delay)
        THEN CALL_PROC
        EXEC_VERTICAL_GIVEN_BITS
               OUT (own predicted alt.);
             CALL_PROC
             EXEC_VERTICAL_GIVEN_BITS
               OUT (intruder predicted alt.);
             Calculate difference in own
             & threat predicted alt.;
             Indicate modeling calculations not needed;
        ELSE select positive maneuvers for climb and descend modeling;
             IF (the modeling logic has not been invoked by the reversal logic)
                  THEN IF (own already has a descend sense advisory)
                            THEN change climb sense advisory to don't descend;
                                   <can't give climb>
                                 Modal quick reaction time; <for subsequent RA>
                       IF (own air already has a climb sense advisory)
                            THEN change descend sense advisory to don't climb;
                                   <can't give descend>
                                 Model quick reaction time;

END Set_up_maneuvers;
```

```
PROCESS Modeling_calculations;

   IF (positive RA chosen for modeling own climb AND
       own aircraft cannot climb AND reversal not selected)
        THEN change own climb to don't descend for modeling;

   CALL_OVER_INTERVAL
    & IN (own, position, rate, modeled climb rate, delay, true
          reversal flag);
   CALL_OVER_INTERVAL
    & IN (own, position, rate, modeled descend rate, delay, true
          reversal flag);
   IF (reversal not selected)
        THEN_OVER_INTERVAL
              & IN. (int. pos., trackod rate, no adv., no delay, true
                    tau, reversal flag);
        ELSE IF (intruder causing crossing OR int. level and own crossing
                  from above OR int. rate opposite
                  in sign)
                   THEN_OVER_INTERVAL
                      IN (int.pos., outer rate, no adv., no delay,
                          true A mod.tau, reversal flag);
                   ELSE_OVER_INTERVAL
                     IN (int.pos., inner rate, no adv., no delay,
                         true mod.tau, reversal flag);
                <Intruder rate and own modeled rate are same sign or
                  intruder is level and own is crossing from below>

   CALL SEPARATION_OVER_INTERVAL
      IN (intruder'         s positions at both tau's for climb and descend)
      OUT (separation for climb, descend sense);

END Modeling_calculations;
```

```
ROUTINE HOT_UPPER_INTERVAL

   IN (position, rate, advisory, delay, true tau, modified tau,
       reversal flag)
   OUT (separation at two times);

        <calculate vertical separation at two tau values>

  CALL HOT_VERTICAL_GIVEN_BITS
     IN (true tau, alt., rate, maneuver, delay, reversal flag)
     OUT (own altitude at end of critical interval);

  CALL HOT_VERTICAL_GIVEN_BITS
     IN (modified tau, alt., rate, maneuver, delay, reversal flag)
     OUT (own altitude at beginning of critical
          interval);

END HOT_UPPER_INTERVAL;
```

```
ROUTINE HOUT_VERTICAL_GIVEN_BITS

    IN (time, altitude, alt. rate, advisory, delay, reversal flag)
    OUT (predicted altitude);

   Use nominal acceleration if not in reversal situation;
   IF (reversal situation)
        THEN use P.RACCEL as acceleration;
   IF (advisory is not null)
        THEN IF (climb sense advisory)
                  THEN set goal to minimum required rate;
              <else goal stays zero indicating required rate already obeyed>
             ELSEIF (descend sense advisory)
                  THEN set goal to maximum required rate;
              <else goal stays zero indicating required rate already obeyed>
             OTHERWISE;
 <else> if advisory null, initial rate applies to entire projection

   IF (acceleration required to obey RA)
        THEN PERFORM Calculate_acceleration;
   Calculate displacement after time at initial rate;
   IF (any time accelerating)
        THEN calculate additional displacement after acceleration;
             IF (any time at goal rate)
                  THEN calculate additional displacement using goal rate;

END HOUT_VERTICAL_GIVEN_BITS;
```

```
PROCESS Calculate_acceleration;

   Calculate Whether goal rate will be achieved by the pro
jected time;
   Calculate time at initial rate, time to accelerate,
     rate achieved, and time at goal rate;

END Calculate_acceleration;
```

```
ZGOAL_VERTICAL_GIVEN_

   IN (proc time, altitude, alt. rate, advisory(       goal) rate,
       delay, direction(sense))
   OUT (predicted altitude);

    Calculate time to accelerate from current rate to goal;
    Escape time - delay time;
    IF (goal rate will not be achieved)
         THEN time of acceleration interval
              Calculate average rate over acceleration period;
              Time at goal rate
         ELSE time of acceleration interval
              Calculate average rate over acceleration period;
              Time at goal rate time;
    Compute over delay time;
    IF (any time left to accelerate)
         THEN add additional displacement during acceleration;
              IF (any time at goal rate)
                   THEN add additional displacement at goal rate;

ZGOAL_VERTICAL_GIVEN_
```

```
ROUTINE RESOLUTION_UPDATE

  IN (pointer to RA to delete, pointer to RA to add);

    Select Pointer to RA to be deleted;
   YCALL DELETE_RESOLUTION_ADVISOR
      IN (pointer to advisory);
    Select Pointer to advisory to add;
    IF (pointer non-null)
        THEN SET Specified Resolution Advisory bit;

END RESOLUTION_UPDATE;
```

[illegible]te-RA [illegible]

```
ROUTINE SEPARATION_OVER_INTERVAL

   IN (own climb projected at beginning and end of interval,
      intruder alt projected at beginning and end of interval
      (if own climb), intruder projected
      at end of interval (if own descent), own projected descent
      at beginning and end of interval)
   OUT (projected separation with climb sense and descend sense);

   Adjust threat alts so they do not go below ground level;
   Adjust own projected descent so they do not go below floor
                                    for descent advisories;
  Climb separation = own climb - intruder
                        projected alt. at the two tau values;
  Descend separation = intruder projected alt. - own
                        projected descent at the two tau values;

END SEPARATION_OVER_INTERVAL;
```

```
ROUTINE VSL_OVER_INTERVAL

 IN (rate, ITF entry)
 OUT (VSL_suffices flag);

    Zero hysteresis term for required separation (ALIM);
    Set local copy of intruder's vertical rate;

    IF (modeling VSL for new threat)
         THEN require extra separation;     <nominally
    ELSFIF (modeling negative for new threat)
         THEN require standard separation;
    OTHERWISE <continuing threat>
         Determine previous cycle's RA in feet par minute;
         IF (previous RA is stronger than RA being modeled)
              THEN require extra separation;   <as above>
    IF (a new threat with sense chosen despite low firmness)
         THEN IF (climb sense)
              THEN use upper bound of intruder's rate;
              ELSE use lower bound of intruder's rate; <desc sense>

    CALL VSL_TEST
       IN (rate, tau at end of critical interval, modification to
           std req'd sep, rate)
       OUT (VSL_sufficas flag)
       INOUT (ITF entry);

    IF (VSL_sufficas is set)
         THEN CALL VSL_TEST
            IN (rate, tau at beginning of critical interval,
                modification to standard req'd separation, rate)
            OUT (VSL_sufficas flag)
         INOUT (ITF entry);

END VSL_OVER_INTERVAL; <Return VSL_suffices only if both tests passed>
```

s-7 (1) RA Strength

References:

```
ROUTINE VSL_TEST

 IN (rate, time, horizontal-vertical equivalence factor, rate)
 OUT (VSL_suffices flag)
 INOUT (ITF entry);

    Calculate limit for time;
    IF (climb sense selected)
         THEN IF (own rate exceeds sub g(own VSL)       <i.e. VSL preventive only>
                   THEN model immediate reduction of own rate to the VSL;
                        <most dangerous compliant maneuver>
         ELSE IF (status is 'continuing')
                   THEN calculate delay time accounting for time
                           since advisory;
                   ELSE use nominal delay time; <status new>
              CALL ZGOAL_VERTICAL_GIVEN_HORIZ;
                    <model delay since VSL corrective>
              Calculate VMD resulting from projection;

   ELSE IF (own rate less than sub g(VSL)         <i.e. VSL preventive only>
              THEN model immediate increase of own rate to the VSL;
              ELSE IF (status is 'continuing')
                        THEN calculate delay time accounting for
                                 time since advisory;
                        ELSE use nominal delay time;
                   CALL ZGOAL_VERTICAL_GIVEN_HORIZ;
                   Calculate VMD resulting from projection;
    CALL CHECK_PROJECTION;
    IF (proj VMD is within threshold diminished if appropriate by
         horizontal-vertical equivalence factor)
         THEN CLEAR VSL_suffices flag;
         ELSE SET VSL suffices flag;

END VSL_TEST;
```

s-72 (?) RA Strength References:

```
FUNCTION EVAL

  IN (Resolution Advisory)
  OUT (value);
      <assign a numeric value to an advisory according to strength>

     IF (no vertical advisory)
          THEN EVAL := 0;
          ELSE assign EVAL value shown below;
               <positive as> = 5
               <negative> = 4
               <VSL500> = 3
               <VSL1000> = 2
               <VSL2000> = 1

END EVAL;
```

References:

f (↑) Climb-Strength f -384 (↓) Descend-Strength -39

```
FUNCTION MMINDE

  IN (advisory bit string)
  OUT (index);

    MMINDE = position of first          'I' in input bit string;
       will be from

END MMINDE
```

```
FUNCTION RANAP

  IN (Resolution Advisory string)
  OUT (mapped index);


    Return mapped index value associated with input bit string,
       as defined in Table 36

END RAMAP;
```

```
TASK HOUSEKEEPING;

    REPEAT WHILE (in coordination lock state);
        <Loop while waiting for coordination lock state to end.
         Performance Monitor should recognize when TCAS has been
         locked for more than P-TUNLOCK seconds and take
         appropriate action.>
    ENDREPEAT;

    SET G.COLOCK using uninterruptible test and set;
    Save lock time;
    Null the Delete_RA List and Delete_Intent List;
    PERFORM Threat_file_housekeeping;
    PERFORM Resolution_advisory_housekeeping;
    PERFORM Sonsitivity_Ievel_housekeeping;
    Send Coordination Update message to transponder;
    CALL COORDINATION_UNLOCK;

END HOUSEKEEPING;
```

Update Coordination-

(7) References:

```
PROCESS Threat_file_housekeeping;

   REPEAT WHILE (more entries in threat file);
        IF (Threat File entry now contains no RA or complement)
             THEN IF (Intruder Track File entry exists for this threat)
                       THEN null the ITF pointer to this threat file;
                  Clear TF entry variables and delete entry;
        Select next entry;
   ENDREPEAT;
   REPEAT WHILE (more entries in threat file);
        IF (refresh timer is not negative) <RA issued>
             THEN IF (too long since own advisory was refreshed)
                       THEN put pointer to own Res. Adv. on Delete_RA List;
                            IF (threat intent not received or was
                                deleted by housekeeping)
                                  <lower loop, this PROCESS>
                                 THEN null the ITF Threat File pointer;
                                      Clear TF entry variables and delete entry;
                                 ELSE null own Resolution Advisory;
        Select next entry;
   ENDREPEAT;
   REPEAT WHILE (more entries in threat file);
        IF (refresh timer is not negative) <intent received>
             THEN IF (too long since threat's RA Complement was refreshed)
                       THEN put pointers to threat's RA Complements on
                            Delete_Intent List;
                            IF (own never had RA versus threat or RA
                                 removed by housekeeping)
                                   <upper loop, this PROCESS>
                                 THEN null the ITF Threat File pointer;
                                      Clear TF entry variables and delete entry;
                                 ELSE null threat RA Complements;
        Select next entry;
   ENDREPEAT;
   PERFORM Threat_file_ITF_Iinkup;

END Threat_file_housekeeping;
```

44 (t) Other Aircraft 1

s- **References:**

```
PROCESS Threat_file_ITF_linkup;

    REPEAT WHILE (more entries in Intruder Track File);
        IF (intruder is TCAS-equipped AND TF entry is not linked to ITF)
            THEN search for TF entry with same discrete address as in ITF;
                IF (matching entry found)
                    THEN save back Pointer to ITF in the TF entry;
                         Save Pointer to TF in the ITF entry;
        Select next ITF entry;
    ENDREPEAT;

END Threat_file_ITF_linkup;
```

```
PROCESS Resolution_advisory_housekeeping;

     <Process Resolution Advisory and Intent Deletion Lists
      received from previous PROCESS>

   <Delete all RA's not used against current threats>
   REPEAT WHILE (more entries on Delete_RA List);
       YCALL DELETE_RESOLUTION_ADVISOR
              <delete if not needed for other threat>
          IN (index to Res. Adv.);
        Select next list entry;
   ENDREPEAT;

   <Delete all RA Complemmts not used by other threats>
   REPEAT WHILE (more entries on Delete_intent List);
        CALL DELETE_INTENT
          IN (index to RA complement);
        Select next list entry;
   ENDREPEAT;

END Resolution_advisory_housekeeping;
```

```
PROCESS Sensitivity_level_housekeeping;

    REPEAT UNTIL (processed all site timers);
         IF (site timer not updated recently enough)
              THEN set site's sensitivity level and timer to null;
    ENDREPEAT;

END Sensitivity_level_housekeeping;
```

[illegible] Logic

[illegible] is called following task R

[illegible] Traffic-advisory-detection for each intruder in the Intruder Threat File. It determines which intruders should receive traffic advisories. It resembles hit-or-miss-test for resolution advisories, in that it performs a test for each intruder on range and, for altitude-reporting, on altitude. The thresholds, however, are larger (↑Table 2 on Page 68)

```
TASK TRAFFIC_ADVISOR

    <Determine which intruders qualify for Traffic Advisories.
     Determine type.  Rank by priority.  Pass to display.>

    Set Traffic Display Vector list to null;

    IF (Traffic advisory mode is enabled)
        THEN REPEAT WHILE (more entries in Intruder Track File);
                PERFORM Traffic_advisory_detection;
                Select next entry;
            ENDREPEAT;

            PERFORM Traffic_display;

END TRAFFIC_ADVISOR
```

[illegible] (↑[illegible]) [illegible]-Threat-Condition

```
PROCESS Traffic_advisory_detection;

    IF (traffic code shows Res.  Adv. status)
            <Set when WL entry created in DETECT_CONFLICTS>
        THEN (Set min TA display time); <No change, will be added to TDV>
    ELSEIF (intruder was determined to be on ground)
        THEN set traffic code to no-display;
             Clear the TA display timer;
    OTHERWISE PERFORM Traffic_parameters;
              PERFORM Traffic_range_test;
              IF (traffic range test passed)
                 THEN PERFORM Range_hit_processing;
                 ELSE IF (intruder has not been displayed long enough)
                     THEN decrement traffic display timer;
                         IF (track is non-mode C type)
                             THEN set traf. code to non-Mode C TA status;
                             ELSE set traffic code to Mode C TA status;
                     ELSE proximity test required;
              IF (proximity test required)
                 THEN PERFORM Proximity_test;
                     IF (proximity test passed)
                         THEN set traffic code to Proximity Advisory status;
                         ELSE set traffic code to no-display;

END Traffic_advisory_detection;
```

340 (t) Potential-Threat-Condition

References:

```
PROCESS Traffic_parameters;

    Determine larger of own and intruder's sensitivity levels;
       <For unequipped intruder, use own sensitivity level>
    Select values of traffic detection parameters based on sensitive level;
    IF (intruder not reporting altitude)
        THEN (use RA threshold value as TA threshold);
               <TAU threshold is reduced for non-Mode C TAs
                to prevent unnecessary advisories>
    ELSEIF (resolution mode is enabled)
        THEN use relative altitude values in track file;
    OTHERWISE calculate relative altitude, alt. rate;
             Calculate absolute values of rel. alt., rate;
             <Logic to calculate these has not been executed this cycle>

END Traffic_parameters;
```

```
PROCESS Traffic_range_test;

    IF (range is smaller than immediate range threshold)
        THEN declare range test passed;
    ELSEIF (range rate diverging more than small amount)
        THEN IF (range rate product exceeds threshold
                   OR range exceeds incremental volume)
                  THEN declare range test failed;
                  ELSE declare range test passed;
    OTHERWISE IF (range rate has very small magnitude)
                  THEN set range rate to small negative value;
                     <to avoid zero divide>
              Calculate modified tau for traffic advisory detection;
              IF (tau is less than threshold)
                  THEN declare range test passed;
                  ELSE declare range test failed;

END Traffic_range_test;
```

References:

(1) Potential-Threat-Range-Test 1

```
PROCESS Range_hit_processing;

    IF (track is non-mode C type)
        THEN IF ((no TA previously issued for this track) AND
                (either bearing is unknown or range is coasted this cycle))
            THEN set traffic code to no-display;
            ELSE IF (own aircraft is above non-mode C cutoff altitude)
                THEN IF (intruder has been displayed long enough)
                         THEN set traffic code to no-display;
                         ELSE set traffic code to non-mode C traffic;
                              decrement traffic display timer;
                ELSE set traffic code to non-mode C;
                     enforce min. disp. time;
        ELSE PERFORM Traffic_altitude_test;
            IF (traffic altitude test passed)
                THEN set traffic code to Mode C Traffic Advisory;
                     Enforce min. disp. time;
                ELSE IF (intruder has not been displayed long enough)
                        THEN set traffic code to Mode C Traffic Advisory
                              status, and decrement traffic display timer;
                        ELSE proximity test required;

END Range_hit_processing;
```

References:

3.10 (t) Potential-Threat-Condition

```
PROCESS Traffic_altitude_test;

     IF (absolute value of relative altitude is small)
          THEN declare altitude test Passed;
     ELSEIF (relative altitude rate diverging by sufficient amount)
          THEN declare altitude test failed;
     OTHERWISE calculate altitude tau;
               IF (altitude tau small enough)
                    THEN declare altitude test passed;
                    ELSE declare altitude test failed;

END Traffic_altitude_test;
```

References:

(1) Potential-Threat-Alt-Test

```
PROCESS Proximity_test;

    IF (range exceeds proximity threshold)
        THEN declare proximity test failed;
    ELSEIF (track is non-mode C type)
        THEN IF (own altitude is above non-mode C cutoff)
        <assumes non-mode-C intruders stay below ceiling allowed by ATC>
                THEN declare proximity test failed;
                ELSE IF ((neither a Traffic nor Proximity Advisory
                          has been issued for this track) AND (either
                          bearing or range coasted this cycle))
                           THEN declare proximity test failed;
                           ELSE declare proximity test passed;
    OTHERWISE IF (relative altitude is within threshold)
                THEN declare proximity test passed;
                ELSE declare proximity test failed;

END Proximity_test;
```

3-13 **References:** (t) Proximate-Traffic-Condition

```
PROCESS Traffic_display;

    CLEAR Display_prox flag;
    IF (Pilot selected proximate traffic display)
        THEN Initialize display countdown timer;
    IF (Display timer has not counted down)
        THEN SET Display_prox flag;
            decrement Proximate traffic display countdown timer;
    REPEAT WHILE (more Intruder Track File entries AND Display_prox flag
                off);
        IF (traffic code high enough) <currently Res. or Traffic Adv. status>
            THEN SET Display_prox flag;
            ELSE select next entry;
    ENDREPEAT;

    REPEAT WHILE (more Intruder Track File entries); <loop through ITF>
        IF (traffic code indicates Proximity test passed AND Display-Prox
            flag is set)
            THEN change traffic code to Display Proximity status;
        PERFORM Traffic_score;
        Select next ITF entry;
ENDREPEAT;

    Sort pointers according to priority indicated by Traffic Score field;
    Move sorted pointers with nonzero Traffic Score to Traffic Display
        Vector;

END Traffic_display;
```

References:

(1) Intent Info

```
PROCESS Traffic score;

    Calculate score multiplier according to altitude reporting capability;

    IF (Traffic code is less than desired display threshold) <normally
              display Proximity, Traffic, and Resolution>
       ;THEN traffic Score field
    ELSEIF (Traffic code indicates Resolution Advisory)
        THEN assign high score value;
    ELSEIF (Traffic code indicates Traffic Advisory)<includes non-alt.>
        THEN IF (range is less than incremental volume)
                  THEN assign medium high score, sorted by range;
             ELSEIF (intruder range is converging)
                  THEN assign medium score, sorted by range tau;
             OTHERWISE assign medium low score, sorted by range;
          <altitude reporting intruders get higher score>
    ELSEIF (Traffic code indicates Proximity advisory)
        THEN assign low score, sorted by range;
    OTHERWISE assign zero score;

END Traffic_score;
```

```
TASK DISPLAY_ADVISORIES;

  <This task decides which advisory and rate to display by analyzing the
  threat file.>

    REPEAT WHILE (in coordination lock state);
          <Loop while waiting for coordination lock state to end.
           Performance Monitor should recognize when TCAS has been
           locked for more than P.TUNLOCK seconds and take appropriate
           action.>
    ENDREPEAT;
    SET G.COLOCK using uninterruptible test and set instruction;
    Save lock time;

    Save previous cycle's strongest climb and descend sense advisories;
    CLEAR global new-threat flag, preventive-to-corrective change flag, and
         increase-rate-RA-issued flag;

    REPEAT WHILE (more entries in Threat File);
         IF (own logic has selected a Resolution Advisory for intruder)
              THEN IF (advisory has climb sense)
                        THEN save this advisory if strongest climb
                                sense so far;
                        ELSE save this advisory if strongest descend
                                sense so far;
                   IF (this threat is new)
                        THEN indicate at least one new threat this cycle;
                             CLEAR new threat flag;
         Select next Threat File entry;
    ENDREPEAT;

    PERFORM Sot_up_goal_rate; <determine rate needed to comply with RA>
    PERFORM Corrective_preventive_ test; <set CORRECTIVE if RA not obeyed>
    PERFORM Set_up_global_flags; <set up display-related global flags>

    CALL COORDINATION_UNLOCK;

END DISPLAY_ADVISORIES;
```

```
PROCESS Set_up_goal_rate;

   <This process selected by Resol.
   descent rates demanded by maximum climb                                   Adv.>

   Initialize goal rates so they are always met;

   REPEAT WHILE (more Threat File entries);
       IF (an RA has been issued against this threat)
           THEN PERFORM Deternine_goal_ rate;
       Select next Threat File entry;
   ENDREPEAT;

END Set_up_goal_rate;
```

```
PROCESS Determine_goal_rate;

   IF (advisory is climb sense)
       THEN set climb goal to larger of min required by RA, or goal
                calculated for previous threat;
            IF (RA positive this cycle)
                THEN IF (not positive last cycle or any new threat
                         this cycle)
                          THEN set displayed-model-goal to larger of
                               nominal climb rate, current climb rate,
                              nor displayed-model-goal for
                               threat if any;
                     IF (an increase rate RA is called for AND either own
                           tracked rate does not now exceed the prescribed
                           increase climb rate, or own previously qualified
                           for the increase RA)
                         THEN SET increase-rate-RA-issued flag;
                              IF (first cycle for increase rate)
                                  THEN SET flags indicating increase
                                                rate encounter, corrective RA;
                                       Set displayed-model-goal to
                                                prescribed increase climb rate;
                         ELSE IF (increase rate RA reversed this cycle)
                                  THEN set displayed-model-goal to max of
                                         nominal climb rate or current rate;
                     Increase climb goal if necessary;
       ELSE perform corresponding calculations for descend sense;

END Determine_goal_rate;
```

References: Corrective-Climb, Altitude-Rate, Increase-Climb, Descend, Corrective, Goal -390

```
PROCESS Corrective_preventive_test;

   IF (both goal rates are zero)  <due to Don't Climb A Don't Descend>
        THEN IF (previous climb sense advisory was corrective AND own
                  rate has slackened to small value AND (RA has weakened
                in strength OR increase rate RA
                just ended))
                   THEN climb sense advisory is no longer corrective;
                   ELSE IF (previous descend sense advisory was corrective
                             AND own rate has slackened to small value AND
                             (RA has weakened in strength OR increase rate
                             RA
                             just ended))
                              THEN descend sense advisory is no longer
                                   corrective;
             IF (climb corrective flag not set AND own is climbing
                    at a rate greater than a small value)
                   THEN climb sense advisory is corrective and is
                          preventive-to-corrective transition;
                   ELSE IF (descend corrective flag not set AND own is
                        descending at a rate greater than a small value)
                             THEN descend sense advisory is corrective and is
                                  preventive-to-corrective transition;
        ELSE IF previous climb sense advisory was corrective AND own climb rate
                   meets/exceeds goal rate AND      (RA has weakened OR
                   increase rate RA
                   just ended))
                   THEN climb sense advisory is no longer corrective;
                   ELSE IF (previous descend sense advisory was corrective
                              AND own descend rate meets
                              /exceeds goal rate
                              AND (RA has weakened OR increase rate RA
                              just ended))
                             THEN descend sense advisory is no longer corrective;
             IF (climb corrective flag not set AND own climb rate does
                    not meet goal rate)
                   THEN climb sense advisory is corrective and is
                          Preventive-to-corrective transition;
                   ELSE IF (descend corrective flag not set AND
                              own descend rate does not meet goal rate)
                        THEN descend sense advisory is corrective and is
                              preventive-to-corrective transition;
```

```
END Corrective_preventive_test;
```

Corrective-Climb

RA-

Climb-RA-Strength-Changed

Strength-Changed

Strengthened

```
PROCESS Set_up_global_flags;

   CLEAR Global flags to be set up;
   IF (a positive advisory is being given which is not corrective)
        THEN SET 'maintain' flag; <GZMODEL contains rate to maintain>
        ELSE IF (no positive RA displayed)
                 ;THEN display-model-goal
   REPEAT WHILE (more entries in Intruder Track File);
        CLEAR Don't-care flag; <as of now, TCAS 'cares'>
        IF (Firmness delay began this cycle)
             THEN indicate at least one firmness delay condition to display;
        IF (no RA is to be displayed)
             THEN CLEAR increase rate encounter flag;
                  IF (Mode C threat became non-altitude reporting during RA)
                       THEN indicate that announcement is needed;
                  ELSEIF (surveillance dropped track on threat during RA)
                       THEN indicate that announcement is needed;
                  ELSEIF (clear of conflict)
                       THEN indicate that "clear of conflict" is to
                               be announced;
             ELSE PERFORM Crossing_flag_check; <RA is to be displayed>
                  IF (RA sense has been reversed and RA is positive Climb
                      or Descend and increase rate RA not issued after
                      reversal)
                       THEN indicate that announcement is needed;
        Select next Intruder Track File entry;
   ENDREPEAT;
   PERFORM Set_up_display_outputs;
   IF (a corrective Resolution advisory is present)
        THEN IF (the advisory has changed from last cycle)
                THEN ind. at least one corrective advisory changed this cycle;
   IF (any new threat OR any change from preventive to corrective
          OR any corrective advisory has changed)
        THEN SET aural alarm flag;

END Set_up_global_flags;
```

, (↑) Alarm-Inhibit

Control (↑) Vertical-Control (↑) Corrective-Strength-Has-Changed

```
PROCESS Crossing_flag_check;

   IF (there is an RA against this intruder AND a threat file entry exists)
       THEN IF ((RA is Climb sense AND own aircraft is at least
                   F.CROSSTHR ft below intruder) OR (RA is Descend
                   sense AND own aircraft is at least P.CROSSTHR ft
                   above intruder))
               THEN IF (RA is positive CLIMB or DESCEND)
                       THEN SET global altitude-crossing flag for display;
                    IF (both ITF crossing flags are not set)
                       THEN Calculate intruder's projected altitude at CPA;
                           IF (own aircraft is modeled as level)
                                 <negative RA issued>
                               THEN SET intruder crossing flag in ITF;
                           ELSEIF (own not modeled as level and intruder
                                 is projected to cross through own
                                 aircraft's current altitude)
                               THEN SET intruder crossing flag in ITF;
                           OTHERWISE SET own crossing flag in ITF;

       <Note: Setting of the INT-CROSS or OWN-CROSS flags determines
        which algorithm is used in the reversal logic to detect an
        intruder level-off maineuver.>

               ELSE IF (encounter was previously altitude crossing)
                       THEN IF (RA is climb sense and own is at least
                                P.CROSSTHR ft above intruder OR RA is
                                descend sense and own is at least
                                P.CROSSTHR ft below int)
                             THEN CLEAR both ITF crossing flags;
                             ELSE; <Crossing flags remain set>
       ELSE CLEAR both ITF crossing flags;

END Crossing_flag_check;
```

802 (1) Crossing **References:**

```
PROCESS Set_up_display_outputs;

   <Determine advisory annunciation precedence>
   IF (increase rate RA issued)
        THEN CLEAR reversal, maintain rate, and altitude crossing flags;
             IF (increase rate RA was not present last cycle)
                  THEN indicate that RA changed to increase rate this cycle;
        ELSE CLEAR indication that increase rate RA was present last cycle;
             IF (maintain rate RA or reversed sense RA issued)
                  THEN CLEAR altitude crossing flag;
                       IF (reversed sense RA requires maintenance of rate)
                            THEN CLEAR sense reversal indication;
                                   <announce maintain>
                                 IF (this is first cycle for maintain rate)
                                      THEN indicate that adv. changed;
                                         <sound alarm>
   IF (no RA is to be displayed)
        THEN IF (announcement needed for Mode C threat that became non-altitude
                    reporting during preceding RA)
                  THEN CLEAR track drop and clear of conflict flags;
                  ELSE IF (announcement needed for threat whose track was
                             dropped during preceding RA)
                            THEN CLEAR clear of conflict flag;
                            ELSE IF (no clear of conflict announcement)
                                    THEN indicate "No Advisory" in
                                          DITS Word 270;
        ELSE IF (a corrective RA has not been issued)
                  THEN indicate "Preventive RA" in DITS Word 270;
                  ELSE indicate either "Corrective Climb Sense RA" or
                     "Corrective Descend Sense RA" in DITS Word 270;
  [illegible] as rate to display;
     <Note: The actual rate that is shown on the RA display is dependent upon
     the quantization and segmentation of the instrument's "eyebrow" lights,
     and so could be different than the rate specified in DITS Word Label
     270>
   Formulate DITS VERTICAL RA DATA OUTPUT WORD FOR TCAS (Label 270)
    to be sent to the RA display, TA display and aural annunciation subsystem;
     <Note that the logic specified above precludes the setting of multiple
     flags for encoding in WORD 270.>
```

```
END Set_up_display_outputs;
```

s- s-7-2 (r) Terminal Control 23

Hardware Design Specifications

This section would includes the design specification for those parts of the system that will be implemented in hardware. We do not have an example for TCAS and thus have not included one. We assume that standard diagrams and notations would be used.

Definition of Standard Conditions

Unless otherwise specified, the signal levels specified in this document are defined at an RF reference point at the antenna end of the cable that connects the TCAS interrogator/receiver equipment to its antenna. Specification values in this document are based upon an antenna transmission line loss equal to the maximum for which the TCAS equipment is designed.

> **Note:** TCAS may be installed with less than the designed maximum transmission line loss. Nevertheless, the standard conditions of this document are based on the maximum design value. Insertion loss internal to the antenna should be included as part of the net antenna gain.

These performance standards, where applicable are specified for an avionics configuration that includes both a Mode S transponder and TCAS equipment. Design specifications that may exist at a possible interface between the Mode S transponder and the TCAS equipment are not covered in detail.

Interference Compatibility with Own Aircraft Mode S Transponder

All of the TCAS requirements stated in this specification shall be met when the TCAS equipment is operating in conjunction with a Mode S transponder with the possible exception of those times that the Mode S transponder is active. The active state of the Mode S transmitter is defined as either the time interval between the leading edge of the first transmitted pulse of a reply minus $10\mu s$ or the time interval during which a mutual suppression occurs, whichever is greater.

Verification Requirements

A standard test specification would go here.

Level 5

Physical Implementation

Software

This section of an intent specification would contain the actual code and hardware implementation. Because this TCAS system specification is written by the FAA and each manufacturer will generate their own code and physical design, it would not contain these things. However, each manufacturer would include their particular system here. For illustration, we have included some code that was written by MITRE from the pseudocode shown in Level 4. We assume this code was written to be used in parts of the simulation of TCAS during evaluation and testing of the logic.

```
<*** INTRUDER TRACK FILE ENTRY ***>
STRUCTURE ITF
  GROUP identity
    PTR CREFNO <Cross-reference to surveillance buffer>
    INT IDINT <Mode S discrete address, if any>
    INT IROW <Track file row number>
    PTR NPTR <Nonlinear track file entry>
    PTR TPTR <Threat File entry>
    PTR WPTR <Working list entry>
  GROUP capability
    INT EQUIP <CAS equipage>
    BIT IOGROUN <Intruder on ground>
    INT LEV <Index to SL-dependent parameters>
    BIT MODC <Mode-C type track>
```

↑ *Supervisory-Interface* ▷ *Pilot-Displays* ▷ *Traffic-Advisories[i]* ▷ *Altitude-Reporting*

```
    INT PLINT <Intruder sensitivity level>
  GROUP position
    FLT A <Abs. value of relative altitude>
    FLT ADOT <Signed value of relative altitude rate>
    INT ARROW <Vertical rate arrow for display>
    FLT BEARING <Bearing relative to own airframe>
    BIT BEAROK <Valid data contained in BEARING>
```

↑ *Supervisory-Interface* ▹ *Pilot-Displays* ▹ *Traffic-Advisories[i]* ▹ *Bearing-Valid*

```
FLT R <Tracked range>
FLT RD <Tracked range rate>
BIT RFLG <Valid data in S.RR (range report);        >
FZLT R  <Relative altitude>
FZLT R   <Relative altitude rate>
INT VALREP <Valid surveillance report indicator>
FDLXINT     <Tracked altitude rate>
FLXT      <Tracked altitude>
```

```
<*** INTRUDER TRACK FILE ENTRY CONTINUED ***>
  GROUP detection
    BIT ALTITUDE_LOST <Mode C threat has become non-altitude reporting>
    BIT CLEAR_CONFLICT <Clear of conflict with this threat>
    BIT DITF <Track needs to be dropped>
    INT KHIT <Detection hit counter>
    BIT TRACK_DROP <Surveillance dropped track against threat>
  GROUP pro
    BIT DCFLG <Don't care about sense>
    FLT SECH <Separation for second-choice sense>
    BIT TAUCAP <Tau not allowed to rise>
    FLT TAUR <modified tau>
    FLT TAURISE <Counter - number of scans TRTRU has been rising>
    FLT TAUV <Time to co-altitude>
    FLT TRTRU <True tau>
    FLT VMD <Vertical miss distance at closest approach>
  GROUP evaluation
    BIT INC_ENC <Inc.  rate RA issued for this threat this encounter>
    BIT INCREASE <Increase rate RA current for this threat>
    FLT INCTIME <Duration of increase rate RA for this threat>
    BIT INT_CROSS <Intruder causing alt-crossing RA to be selected>
    BIT OWN_CROSS <Own causing alt-crossing RA to be selected>
    BIT REVERSE <Reversal RA issued for this threat>
    BIT TIEBREAKER_REVERSAL <Own must reverse due to lost tiebreaker>
  GROUP delay
    INT BADFOK <Sense chosen despite low firmness>
    INT IFIRM <Firmness of vertical track>
    INT LEVELWAIT <No.  of cycles own TCAS has waited for intent>
    INT PREWARN <Firmness delay cycle counter>
    FLT ZDINR     <Slacker bound of rate uncertainty>
    FLT ZDOUTR     <Steeper bound of rate uncertainty>
```

```
<*** INTRUDER TRACK FILE ENTRY CONTINUED ***>
  GROUP traffic
    INT TACODE <Status for display>
    FLT TASCORE <Priority relative to entries with same TACODE>
    FLT TATIME <Time set to enforce min TA display time>
  GROUP timer
    FLT DT <Time difference of 2 latest reports.>
    FLT RRTI <Time of latest range/alt. report>
    FLT TCMD <Time RA issued/changed for this intruder>
    FLT TDATAI <Cycle time (TCUR) of last update>

ENDSTRUCTURE;
```

```
<*** THREAT FILE ENTRY ***>
STRUCTURE TF

  GROUP identity
    INT ID <Mode S discrete address>
    PTR IPTR <Track file row number>

  GROUP advisory
    BIT DEFER_DISPLAY  <RA display deferred pending compatibility
                        check with intent from threat TCAS>
    INT DEFER_COUNT <No.  of cycles RA display has been deferred>
    BITS PERMTENT(32)   <Advisory saved for this threat>
    INT POOWRAR <Index to own Res.  Adv.  array>
    INT POTHRAR(2)   <Index to Res.  Adv.  Complements received from threat>
```

↑ *Supervisory-Interface* ▹ *Pilot-Controls* ▹ *Resolution-Advisory* ▹ *Vertical-RAC*
↑ *Supervisory-Interface* ▹ *Pilot-Controls* ▹ *Resolution-Advisory* ▹ *Horizontal-RAC*

```
    BITS TEMPRA(12)    <Temporary storage for deferred RA>

  GROUP timer
    BIT NEW <New entry due to own Res.  Adv.>
    FLT TLRCMD <Time (TCUR) own advisory refreshed>
    FLT TTHLRCM <Time threat adv.  refreshed>

ENDSTRUCTURE;
```

```
<***WORKING LIST ENTR ***>

[illegible] entries are ordered by status with                    >

STRUCTURE WL

  GROUP entry
    INT STATUS <Entry type>
    PTR IPTR <Pointer to ITF entry>

ENDSTRUCTURE;
```

```
<*** NONLINEAR TRACK FILE ENTR ***>

<There is an NLTF entry for own and for each ITF entry of altitude-reporting

STRUCTURE N

  GROUP output
    FLT Z1   <Tracked altitude, old   >
    FLT ZD2    <Tracked altitude rate, old   >

  GROUP geometry
    INT DIREC <Direction of previously established trend>
    INT DIRECN  <Direction of last transition>
    INT LASTRAN <Mode of track>
    FLT RESID <Residual, old    >
    FLT TBIN <Estimated time to cross alt.  bin, old   >
    FLT TTRANGUES <Delay for changing rate following coasts>
    FLT ZDINNER     <Slacker bound of rate uncertainty>
    FLT ZDOUTER     <Steeper bound of rate uncertainty>
    FLT ZSAVE     <Previous altitude report, old   >

  GROUP firmness
    FLT BINSTHIS   <Bins crossed at current rate, old   >
    INT CASFIRM <Firmness indicator>
    INT FIPRECO <Firmness prior to coast sequence>

  GROUP timer
    FLT TDAT <Time of last altitude report, old   >
    FLT TOSCIL <Time oscillation state entered>
    FLT TSTART <Startup time>
    FLT TTRAN <Time of last transition, old   >
    FLT TUPDT <Time of last track update, old   >

ENDSTRUCTURE;
```

```
<***GLOBAL VARIABLES***>
STRUCTURE G

  GROUP status
    INTMEALLPRO      <Countdown timer for prox traffic display>
    BIT INITFLG <System initializing>
    BIT INTMODE <Interrogation enabled>
    BIT MACFLG <Multiple threats>
    PTR NPTR <Own nonlinear track file, if mode-C tracking used>
    BIT OPFLG <System operational>
    INT RADAROUT <Number of cycles without radar alt.  data>
    BIT RAMODE <Resolution advisories enabled>
    BIT TAMODE <Traffic advisories enabled>
    FLT TCUR <Time of current processing cycle>

  GROUP sensitivity
    INT INDE  <Own sensitivity level>
    INT LA    <Altitude-related sensitivity level>
    15T LEVELSIT(    <Sens.  levels sent from ground sites>
    15T LEVELTIM(    <Time each level refreshed>

  GROUP position
    BIT CLIMBINHIB <fpm500aircraft cannot climb at         >
    BIT INC_CLMINHIB <fpm2500aircraft cannot climb at         >
    BIT NODESCENT <Own near ground; descend RAs inhibited>
    FLT TDATAO <Time of own last altitude update>
    FGROUND     <Ground elevation estimate>
    FOWN      <Own tracked altitude>
    FDOWN      <Own tracked altitude rate>

  GROUP settable
    FLT ALIM <Positive advisory alt.  threshold>
    FLT SENSFIRM <Required separation assuming no vert.  tracking error
    FLZR      <Detection alt.  threshold>
```

```
<***GLOBAL VARIABLES CONTINUED***>
  GROUP resolution
    BIT COLOCK <Coordination Lock>
    INT IDOWN <Own aircraft's Mode S discrete address>
```

↑ *Own-Aircraft-Model* ▹ *Mode-S-Address*

```
    4BITS INTENT(   <Threat Resolution Advisory Complement array>
    B14TS RA(    <Resolution Advisory Array>
```

↑ *Supervisory-Interface* ▹ *Pilot-Controls* ▹ *Resolution-Advisory* ▹ *Climb-RA*
↑ *Supervisory-Interface* ▹ *Pilot-Controls* ▹ *Resolution-Advisory* ▹ *Descend-RA*

```
    FLT TLOCK <Time Lock State initiated>
  GROUP display
    BIT ALARM <Sound aural alarm>
```

↑ *Supervisory-Interface* ▹ *Pilot-Controls* ▹ *Aural-Alarm*

```
    BIT ALLCLEAR <Announce "clear of conflict" message>
    BMTTLAOST       <Announce "threat became non-altitude reporting durin
    BYCORCAHNANG       <Changed RA is corrective>
    BYRIOSASN       <Encounter is an altitude cross>
    BYIIRMADEL       <Intruder's RA delayed by firmness for first time>
    BYNCRAENASE       <Increase rate RA issued>
    BMEWTAHR       <RA due to new threat>
    BMRIECAOR       <RA changed from preventive to corrective>
    BMEIVEARNSE       <RA reversal issued>
    BMRTACAKDROP       <Announce "surveillance dropped track during RA">
    INT CLSTROLD <Previous value of CLSTRONG>
    INT CLSTRONG <Strongest climb sense RA>
```

↑ *Supervisory-Interface* ▹ *Pilot-Controls* ▹ *Resolution-Advisory* ▹ *Climb-RA*
↑ *Supervisory-Interface* ▹ *Pilot-Controls* ▹ *Resolution-Advisory* ▹ *Descend-RA*

```
    BIT CORINC <Increase rate RA is corrective>
    BIT CORRECTIVE_CLM <Climb sense RA is corrective>
    BIT CORRECTIVE_DES <Descend sense RA is corrective>
    INT DESTROLD <Previous value of DESTRONG>
    INT DESTRONG <Strongest descend sense RA>
    BIT INC_ENCOUNTER <Increase rate RA previously issued>
    BIT MAINTAIN <Positive RA is preventive>
    BIT PREVINCREASE <Increase rate RA issued previous scan>
    BIT TURN_OFF_AURALS <If set, aural annunciations are inhibited>
```

↑ *Supervisory-Interface* ▹ *Pilot-Controls* ▹ *Aural-Alarm*
↑ *Supervisory-Interface* ▹ *Pilot-Controls* ▹ *Aural-Alarm-Inhibit*

```
    FDMZODEL      <Escape rate to maintain for safe separation>
```

↑ *Supervisory-Interface* ▹ *Pilot-Controls* ▹ *Own-Goal-Altitude-Rate*

```
  GROUP cross_reference
    PTR CREFPTR(     <Array of pointers to ITF entries
    indexed on surveillance buffer numbers>
ENDSTRUCTURE;
```

<***OWN AIRCRAFT DATA INPUT***>
STRUCTURE O

GROUP status
BIT ALLPRO <Pilot selected prox traffic display>

↑ *Own-Aircraft-Model* ▷ *Prox-Traffic-Display*

BIT GROUND_MODE <'1'Traffic display permitted on ground >

↑ *Own-Aircraft-Model* ▷ *Traffic-Display-Permitted*

INT MANUAL <Manual sensitivity level selection>

↑ *Supervisory-Interface* ▷ *Pilot-Controls* ▷ *Mode-Selector*

BIT OOGROUN <Own aircraft on ground>

↑ *Own-Aircraft-Model* ▷ *Air-Status*

BIT RADAROK <Valid radar altitude data this cycle>

↑ *Own-Aircraft-Model* ▷ *Radio-Altimeter-Status*

BIT TCASOP <TCAS operational status>

↑ *CAS-Operating-Modes* ▷ *Status*

GROUP geometry
FLT ZADC <Own fine altitude>

↑ *Own-Aircraft-Model* ▷ *Barometric-Altitude*

FLT TADC <Time of fine altitude report>
FLT ZDADC <Own altitude rate (optional)>

↑ *Own-Aircraft-Model* ▷ *Altitude-Rate*

INT ZOWN <Own Mode C altitude report>

↑ *Own-Aircraft-Model* ▷ *Barometric-Altitude*

FLT TROWN <Time of Mode C alt. report>
FLT ZRADAR <Radar altimeter report>

↑ *Own-Aircraft-Model* ▷ *Radio-Altitude*

ENDSTRUCTURE;

STRUCTURE REALTIME

GROUP data
FLT TCLOCK <Onboard realtime clock>

```
ENDSTRUCTURE;
```

```
<***INTRUDER SURVEILLANCE BUFFER DATA***>
STRUCTURE S

  GROUP identity
    INT SURVNO <Surveillance buffer number>
    INT IDINTR <Mode-S discrete address, if any>
    INT E   <Equipage>
    INT PLINT <Sensitivity level (=0 non-TCAS)                  >
    BIT MODC <Intruder is reporting altitude>

  GROUP geometry
    FLT RR <Range report>
    FLT RRTIME <Time stamp for range          >
    BIT RFLG <Valid data in RR.       >
    INT ZINT     <Mode-C altitude report>
    BIT ZFLG     <Valid data in ZINT      >
    FLT BEAR <Intruder bearing relative to own heading>
```
↑ *Supervisory-Interface* ▹ *Pilot-Displays* ▹ *Traffic-Advisories[i]* ▹ *Bearing*
```
    BIT BEAROK <Valid data in BEAR>

ENDSTRUCTURE;
```

```
<*** SYSTEM PARAMETERS ***>
STRUCTURE P

  GROUP track
    FLT ALFAO <Own fine altitude smoothing>
    FLT BETAO <Own fine alt.  rate smoothing>
    FLT ALFAR <Intruder range smoothing>
    FLT BETAR <Intruder range rate smoothing>

  GROUP sensitivity
    INT AUTOMATIC <TCAS is in automatic mode; SL selected based on alt.
    F4LT BOT(   <Lower hysteresis bounds for alt.-sensitive parameter
    F4LT TOP(   <Upper hysteresis bounds for alt.-sensitive parameter
    INT INDSTART <Sensitivity level for system startup>
    FLT KNOWGROH <Upper alt.  for updating ground alt.>
    FLT KNOWGROL <Lower alt.  for updating ground alt.>
    BIT LOWTA <Option to generate TAs in sens.  le2vel  >
↑ Own-Aircraft-Operating-Modes ▷ TA-In-Sens-Level-2
    FLT NEARGROH <Upper hysteresis bound for on-ground test>
    FLT NEARGROL <Altitude AGL where threat declared on ground>
    INT RADARLOST <No.  cycles without radar alt.  to invoke default>
    INT SITMA  <Number of ground sites that can set sensitivity>
    INT SLNORAD <Default SL, when radar alt.  absent long time>
    FLT STIMOUT <Timeout for ground SL command>
    INT TA_ONL  <TCAS is in TA-only mode; no RAs permitted>
    FLARGE     <default when ground level can't be estimated>
    F          <Upper boundary SL 2/4           >
    F          <Lower boundary SL 2/4           >
    F          <Upper boundary SL 4/5           >
    F          <Lower boundary SL 4/5           >
    F          <Upper boundary SL 5/6           >
    F          <Lower boundary SL 5/6           >
    F          <Upper boundary SL 6/7           >
    F          <Lower boundary SL 6/7           >
```

```
<***SSYTEM PARAMETERS CONTINUED ***>
  GROUP detect
    7FLT DMODTBL(   <Incremental range protection>
    FTBL(H        <Divergence range*range rate hyperbola>
    2FLT LOWFIRMR  <Alt.  cross threshold on low firmness>
    FALTDIMIFF        <Max.  altitude difference to issue an RA>
    FLT NAFRANGE <Minimum range to filter nuisance alarms>
    FLT RDTHR <Small range rate for tau calc.>
    XFLT RMA  <Maximum range for threat detection>
    7FLT TRTETBL(   <Range tau limit, equipped intruder>
    7FLT TRTUTBL(   <Range tau limit, unequipped intruder>
    7FLT TVPCTBL(   <Max.  pgrotion for VMD calculation   >
    7FLT TVPETBL(   <Max.  pgrotion modeling escape   >
    7FLT TVTETBL(   <Altitude tau limit, equipped intruder>
    7FLT TVTUTBL(   <Altitude tau limit, unequipped intruder>
    FDZTHR     <Small vertical rate to inhibit zero-divide for tau c
    F2LZ       <Immediate altitude threshold>

  GROUP delay
    FLT MINTAU <Lower limit for tau and value when diverging>
    FLT MINTAUM <Floor for beginning of critical interval>
    2FLT SF(   <Req'd sep.  for bad firmness hit using vert rate limi
    INT WTTHR <Waiting threshold for RA deferral>
```

<*** SYSTEM PARAMETERS CONTINUED ***>

```
  GROUP model
    FLT BACKDELA  <Floor on (current time - begin-maneuver time)>
    FLT CLMRT <Nominal rate of response to positive climb>
    FLT DESRT <Nominal rate of response to positive descent>
    FLT MINRVSTIME <Min.time for reversal for threat close in alt.>
    FLT MINSEP <Min.  separation needed for non-cross logic>
    FLT MIN_RI_TIME <Min.time to inhibit reversals and increases>
    FLT CROSSINC       <Incr.  sep req'd to cross alt when can't climb>
    FLT OWNDEL <Pilot delay time, positive to negative>
    FLT QUIKREAC      <Pilot delay time between RA's>
    FLT RACCEL <Acceleration response to Sense Reversal RA>
    FLT TMIN <Time minimum to display RA            >
    FLT TV  <Response delay to first RA>
    FLT VACCEL <Nominal acceleration responding to RA>
    FLT ZDESBOT     <Own final altitude after leveloff at NODESLO>
    FLT MAXA(           <Greatest rate to model>
    FLT MINTB(          <Smallest rate to model>
  GROUP seladv
    FLT AL(   <Positive advisory threshold>
    FLT FRTHR(     <Tau threshold to delay stronger RA if bad firmness>
    FLT ILEV <Vertical rate inducing immediate positive RA>
    FLT NEWVSL <Factor by which VSL assumed to be violated>
    FLT NODESHI <Ceiling on own alt.AGL to issue DESCEND RAs>
    FLT NODESLO <Floor on own alt.  AGL to issue DESCEND RAs>
    FLT OLEV <A "substantial" vertical rate>
    FLT STROFIR <Tau ceiling to delay stronger RA if bad firmness>
    FLT TNOWEAK <Time advisory not allowed to weaken>
    FLT TRVSNOWEAK <Time adv.  not allowed to weaken in reversals>
    FLT 1000V     <1000 ft/min limit       >
    FLT 2000V     <2000 ft/min limit       >
    FLT 500V    <500 ft/min limit      >
```

```
<***SSYTEM PARAMETERS CONTINUED ***>
  GROUP eval
    FLT AVEVALT <Required separation to pass Increase Rate test>
    7FLT AVEVTAU(   <Tau ceiling to perform Increase Rate test>
    INT MINFIRM <Intruder firmness level required for Increase Rate t
    FNLZT    _INCDES <Alt.  limit to issue Increase Descent RAs>

  GROUP coordination
    4INT HMAPINTENT(   <Map codes in coordination msg into horizontal INTENT
    4INT VMAPINTENT(   <Map codes in coordination msg into vertical INTENT a
    FLT TCATRES <Time to clean up stray TF entry>
    IMNAT TR    <Interrogation limit when no response>
    FLT TUNLOCK <Limit for coordination lock state>
    16INT PTABLE(    <Parity table to confirm validity of RA msg>

  GROUP correct_prev
    3FLT CLIMBGOAL(   <Expected rate for climb sense RA>
    3FLT DESGOAL(   <Expected rate for desc.  sense RA>
    FLT HUGE <Large constant>
    FSLTEHCOR       <Hysteresis test corrective                >
    FLT SMALLRATE <Rate limit for compliance with DCL,DDES>

  GROUP general
    BIT AIRDATA <Fine altitude data available>
```

↑ *Own-Aircraft-Model* ▷ *Barometric-Altimeter-Status*

```
    FLT CROSSTHR <Altitude-crossing threshold>
    INT LARGEINT <General-purpose large integer>
    FLT TINIT <Initial value for timer variables>

  GROUP display
    FLT INC_CLMRATE <Climb rate for increase rate RA>
    FLT INC_DESRATE <Descend rate for increase rate RA>
```

```
<***SYSTEM PARAMETERS CONTINUED ***>
  GROUP traffic
    FLT ABOVNMC <Max.  alt.  to display traffic adv.  on int.  not report
    DUR ALLPRO     <Max duration for prox traffic display>
    INT DISPTHR <minimum TACODE level displayed>
    FLT DMODTA_TBL(   <Incremental range volume>
    FLT HISCORE <TASCORE for         >
    FLT H   _TBL(   <Range*Range rate divergence>
    FLT LOSCORE <nominal TASCORE for proximity type>
    FLT MEDHISCORE <nominal TASCORE for nearest intruders>
    FLT MEDLOSCORE <nominal TASCORE for diverging intruder>
    FLT MEDSCORE <nominal TASCORE for converging intruder>
    INT NEEDPRO  <TACODE level for which proximity traffic displayed>
    FLT MINTATIME <Minimum display time for a TA>
    FLT PRO   <Altitude limit for proximity alert>
    FLT PRO   <Range limit for proximity alert>
    FLT RDTHRTA <Range rate limit for tau calc.>
    FLT RTHRTA_TBL(   <Range for immediate Threat alert>
    FLT TRTHRTA_TBL(   <Range tau for Threat alert>
    FLT TVTHRTA_TBL(   <Altitude tau for Threat alert>
    FLT  THRTA     <Alt.  rate limit for tau calc.>
    FLT     _AURALHI <Upper "aurals inhibited" boundary>
    FLT     _AURALLO <Lower "aurals inhibited" boundary>
    FLT  RTA     <Altitude limit for threat alert>

ENDSTRUCTURE;
```

```
<***NLTF PARAMETERS***>
STRUCTURE PN

  GROUP general
    FLT DT <Nominal update time>
    FLT LARGEPOS <Bincross time when rate is level>
    FLQT   <Size of altitude bin quantization>
    FLT SMALLPOS <Term to avoid zero-divide>
    FDLZARGE     <DSteepest confidence limit on   >
    FDLZIKEL      <Most likely vertical rate after a transition>

  GROUP no_transition
    FLT LATELEVEL <DBincross discrepancy to declare     >
    FLT LATESLACK <DBincross discrepancy to slacken   >
    FLT TGOLEV <DInterval of unchanged reports to declare     >
    FDLZECA      <DESign factor LASTRAN for                   >

  GROUP no_trans_firmness
    F1LT GUESDU  <Bin occupancy time to rule out steep rate>
    F2LT GUESDU  <Bin occupancy time to rule out moderate rate>
    F3LT GUESDU  <Bin occupancy time to imply leveloff>
    F1LT OVERDU  <Bincross discrepancy to lower confidence in estab.  r
    F2LT OVERDU  <Bincross discrepancy to raise confidence rate is zer
```

```
<***NLTF PARAMETERS CONTINUED***>
  GROUP transition
    HMATEEARL      <Bincross discrepancy to ignore prior history, reinit
    FLT GUESRATE <Vertical Rate estimate when bincross follows level>
    HLT RESIDECA  <RESID adj when substantial bincross discrepan
    FLT RESIDIMIN <RESID adj when slight bincross discrepancy        >
    HLT SECS  <Constant to compute ceiling on relative weighting>
    FLT SHORT <Constant to compute ceiling on relative weighting>
    FLT SLITEOFF <Bincross RESID discrepancy to ad          >
    FLT TBINBETA <Rel weighting, observed to expected, to update bincr
    FLT TBINMIN <Floor on bincross estimate when reinitializing rate>
    FLT TRANGU <RATE, SLITEOFF, LATESLACK following
    FLT       <Time near bin bndry when track vulnerable to spuriou

  GROUP transition_firmness
    F2LT DUE  <Error allowance in computing allowable time in curre
    F1LT HI  <Expected bincross, fair firmness                  >
    F2LT HI  <Expected bincross, good firmness                  >
    F3LT HI  <Expected bincross, maximum firmness
    F1LT LO  <Deficiency in actual, fair firmnes
    F2LT LO  <Deficiency in actual, good firmnes
    F3LT LO  <Deficiency in actual, maximum firm
    F2LT OUT  <Bincross discrepancy floor, good firmness>
    F3LT OUT  <Bincross discrepancy floor, maximum firmness>
    H1LT SLACKEN  <Bin occupancy time needed for fair firmness>
    E2LT SLACKEN  <Bin occupancy time needed for good firmness>
    FLTZRACC     <Uncertainty factor acceleratio

ENDSTRUCTURE;


TASK MAIN_LOOP;

    PERFORM Initialize;
    REPEAT WHILE ($TRUE);

        Set Working List to null;

        <If processing of Mode S messages is not performed on a periodic
```

```
     basis per cycle, then ROUTINE MODE_S_MESSAGE_PROCESSING must be
     invoked here in TASK MAIN_LOOP to handle any Mode S messages
     that may have arrived while own TCAS was in Coordination Lock
     State.>

    CALL TRACK_OWN;
    IF (G.IMODE EQ $TRUE)
        THEN CALL TRACK_INTRUDERS;
            IF (CRSMODE EQ $TRUE)
                THEN CALL DETECT_CONFLICTS;
    REPEAT WHILE (more WL entries);
        CALL RESOLUTION_AND_COORDINATION
          IN (WL entry);
        Select next WL entry;
    ENDREPEAT;

    CALL TRAFFIC_ADVISORY;
    CALL DISPLAY_ADVISORIES;
    CALL HOUSEKEEPING;

ENDREPEAT;

END MAIN_LOOP;
```

```
PROCESS Initialize;

    Initialize all constants;
    Set all data structures to null;
    CLEAR all bits in G.RA array;
  NUSET; all pointers in G.CREFPTR array to
    SET G.INITFLG;
    ZGROUND
    G.RADARLOST;
    G.IXDE      INDSTART;
    GZR.A1;
    CLEAR all elements in G.LEVELSIT array;
    CLEAR G.INTMODE;
    CLEAR G.RAMODE;
    CLEAR G.CORRECTIVE_CLM, G.CORRECTIVE_DES;
    G.ZMODEL
   ;G.COLSTRONG, G.DESTRONG
   ;G.COLSTROLD, G.DESTROLD
    G.TIMELPRO
    CLEAR all bits in G.INTENT array;
    CLEAR G.PREVINCREASE;
    CLEAR G.INC_ENCOUNTER;
    CLEAR G.CORINC;
    CLEAR G.NODESCENT;
    SET G.TURN_OFF_AURALS;
    CLEAR G.COLOCK;
    Initialize coordination message queues and pointers;
    Clear all bus working areas and NTA;
    Start real-time clock;
    Send initial sensitivity level setting to transponder;
    Input own aircraft info; <Altitude, Mode S ID, Max.  Airspeed,
    Control Panel settings>

END Initialize;
```

```
<*** RECEIVE TASK LOCAL VARIABLES ***>
STRUCTURE RCV_VAR
  GROUP general
    INT INTIND <Numeric value of complement to cancel>
    BIT SUCCESS <Matching entry found>

  GROUP message
    INT CHC <3-bit subfield to cancel horizontal complement>
    INT CVC <2-bit subfield to cancel vertical complement>
    INT HRC <3-bit subfield to add horizontal complement>
    INT HSB <5-bit subfield for horiz.sense parity protection>
    INT IIS <Mode S interrogator identification subfield>
    INT ICHC   <Local variable to hold low order bits of CHC>
    INT IHRC   <Local variable to hold low order bits of HRC>
    INT MID <Mode S ID of interrogating TCAS aircraft>
    BIT MTB <Multi-threat bit>
    INT SLC <Uplinked Sensitivity Level Command from Mode S sensor>
    INT VRC <2-bit subfield to add vertical complement>
    INT VSB <4-bit subfield for vert.  sense parity protection>
    BIT BAD_DATA <Parity error or illegal data in message>
    INT PARIT  <Value from P.PTABLE based on bits in message>

ENDSTRUCTURE;
```

```
TASK RECEIVE;
    IF (TCAS Resolution (Intent) message received)
        THEN put message on Resolution message queue;
            CALL RESOLUTION_MESSAGE_PROCESSING;
    ELSEIF (Mode S message received)
        THEN put message on Mode S message queue;
            CALL MODE_S_MESSAGE_PROCESSING;

END RECEIVE;
```

```
ROUTINE RESOLUTION_MESSAGE_PROCESSING;

    <The queued message must be decoded and the appropriate field
     values assigned to the proper variables used by this logic.
     Fields in the message are CVC, CHC, MID, MTB, HRC, VRC, VSB, HSB>

    IF (QACSSEOCK E          <NOUTEinterruptible test and
        THEN SET G.COLOCK; set of G.COLOCK>
            G.HCLLOCK;
            REPEAT WHILE (more entries in Resolution message queue);
                Get Resolution message from queue;
                PERFORM Process_threat_intent;
                Select next entry;
            ENDREPEAT;
          : Set up fields in Coordination Update message
                RAC.     IGTEMAT, ARA
            Send Coordination Update message to transponder;
            CALL COORDINATION_UNLOCK;

END RESOLUTION_MESSAGE_PROCESSING;
```

```
PROCESS Process_threat_intent;

    CLEAR BAD_DATA;
    PWR=PT TABLE (        BI 46             CVC, VRC) in coordination message
        from threat TCAS);
    <NOTE: Bit positions may be different for message received from transpond
    MNE(PASSBIT subfield in coordination message from threat TCAS)
        THEN SET BAD_DATA;
        ELSE IF ((CVC AND (          CHOICE AND (VRC E                  (VRC E
                THEN SET BAD_DATA;
                ELSE IF ((CVC 3 E (VRC E
                        THEN SET BAD_DATA;
    IF (BAD_DATA FALSE)
        THEN PERFORM Find_threat_file_entry;
            TF=CLOCK; CM
            PERFORM Process_valid_data;

END Process_threat_intent;
```

```
PROCESS Find_threat_file_entry;

    CLEAR SUCCESS;
    REPEAT WHILEQ(ASSE)e;TF entries AND SUCCESS E
        IF (entry found with TF.IQMED)
            THEN SET SUCCESS;
                Save pointer to TF entry;
            ELSE select next entry;
    ENDREPEAT;

    IF(ASSE)UCCESS E
        THEN create new TF entry;
            TF.INPEIRS;
            TF.ID=ID;
            TF=.TIINOMD;
            CLEAR TF.NEW;
           2=TF.THOPRIAR,ATF.POTHRAR(
            CLEAR All bits in TF.PERMTENT;
            CLEAR TF.DEFER_DI;SPLA
            TF.DEFER_C;O=UNT
            CLEAR All bits in TF.TEMPRA;
            Save pointer to TF entry;

END Find_threat_file_entry;
```

```
PROCESS Process_valid_data;
    IF (CVC0) NE    <vertical complement to cancel>
        THEN1= TF.POTHRAR(
            Save current TF pointer;
            CALL DELETE_INTENT
                IN (P.VMAPINTENT(CVC));
            Restore current TF pointer;
    IF (CHC0) NE    <horizontal complement to cancel>
        THEN2= TF.POTHRAR(
            L2CHC= low order bits of CHC;        <Bits 6-8 coordination
                message from threat TCAS III aircraft>
            Save current TF pointer;
            CALL DELETE_INTENT
                IN (P.HMAPINTENT(L2CHC));
            Restore current TF pointer;

    IF0)(VRC NE    <vertical complement to add>
        THEN IF (1F NEOTPHRVMA(PINTENT(VRC))
                THEN VMAPTF.POTHRAR(VRC);
                    SET G.INTENT(P.VMAPINTENT(VRC));
    IF0)(HRC NE    <horizontal complement to add>
        THEN L2HRC= low order bits of HRC;           <Bits 5-6 coordination
                message from threat TCAS III aircraft>
            IF (2F NEOTPHHMA(PINTENT(            L2HRC))
                THEN HMAPTF.POTHRAR(              L2HRC);
                    SET G.INTENT(P.HMAPINTENT(L2HRC));

END Process_valid_data;
```

```
ROUTINE MODE_S_MESSAGE_PROCESSING;

    <Note that ROUTINE MODE_S_MESSAGE_PROCESSING can be interrupted by
     additional incoming Mode S messages.  The implementation must ensure
     that MODE_S_MESSAGE_PROCESSING is prevented from beginning processing
     of an incoming Mode S message while the processing of a previous
     Mode S message is still underway.>

    <The queued message must be decoded and the appropriate field
     values assigned to the proper variables used by this logic.
     Fields in the messages are IIS, SLC, TCAS Broadcast Bit,
     own Mode S ID, own altitude, own aircraft maximum airspeed,
     and control panel data.>

    REPEAT WHILE (more entries in Mode S message queue AND G.COLOCK E
            $FALSE AND processing of previous Mode S message complete);
        Get message from queue;
        IF (message is Sensitivity Level Command)
            THEN PERFORM SL_command_processing;
        ELSEIF (message is Periodic Data)
            THEN PERFORM Periodic_data_processing;
        ELSEIF (message is TCAS Broadcast data)
            THEN PERFORM Broadcast_processing;
        Select next entry;
    ENDREPEAT;

END MODE_S_MESSAGE_PROCESSING;
```

```
PROCESS SL_command_processing;

    <Fields in message are IIS, and SLC. IIS identifies interrogator.>

    IF (SLC = 1) On (message)
        THEN G.LEVELSIT(IIS) = 0;
    ELSEIF (SLC GT 1) AND (SLC LE 15)          <Valid SLCs are 2 through 15>
        THEN G.LEVELSIT(IIS) = SLC;
             G.LEVELTIM(IIS) = CLOCK;
    OTHERWISE;

END SL_command_processing;
```

```
PROCESS Periodic_data_processing;
```

↑ *Supervisory-Interface* ▷ *Pilot-Controls* ▷ *Mode-Selector*
↑ *Own-Aircraft-Model* ▷ *Barometric-Altitude*
↑ *Own-Aircraft-Model* ▷ *Mode-S-Address*
↑ *Own-Aircraft-Model* ▷ *Altitude-Rate*

```
    Determine if the data received is own altitude, own Mode S ID,
        own aircraft maximum airspeed, or control panel information;
    IF (altitude message received)
        THEN IF (fine resolution)
                THEN ZOWN ← fine altitude from message;
                    O.HALTCK;
                ELSE ZOWN ← coarse altitude from message;
                    O.HCRLOCK;
    IF (own update message received)
        THEN G.ID ← own Mode S ID;
            O.MANUAL ← pilot-selected Sensitivity Level from Control Panel;
            Save maximum airspeed for use by surveillance;
            Save other display control data, control panel data, hardware
                status info for use by display subsystems, surveillance, and
                performance monitoring;

END Periodic_data_processing;
```

```
PROCESS Broadcast_processing;

    Store the Mode S ID of the reporting TCAS aircraft and the
        associated time;

    <Once per second, at the beginning of the surveillance cycle,
     compute NTA to be the number of distinct TCAS addresses monitored
     within the previous -20 second period.                     >

END Broadcast_processing;
```

```
ROUTINE DELETE_INTENT
  IN (INTIND);

    IF0)(INTIND NE
        THEN CLEAR SUCCESS;
            REPEAT WHILEQ(FALSE);more entries in TF AND SUCCESS E
              2)THOR(INTIND)RE(
                    THEN SET SUCCESS;
                    ELSE select next TF entry;
            ENDREPEAT;

            QRUE)SUCCESS E
                THEN; <cannot delete intent which applies to another threat>
                ELSE CLEAR G.INTENT(INTIND);

END DELETE_INTENT;
```

```
STRUCTURE TRACKVAR

  GROUP range
    INT LOWLEVEL <Ground site sensitivity level choice>
    BIT RADARCRED
    FLT RP <Pgroted range at old rate   >
    INT SIT

  GROUP vertical
    FLT ALT_ABOVE_GND <Own aircraft's altitude above ground level>
    F1LT BETA
    FLT BINFIRS <Earliest credible time in this bin>
    FLT BINLAST <Latest credible time in this bin>
    FLT BLIM
    FLT CREDREP <Variable used to determine if report credible>
    FLT DBINS <Number of bins crossed since last report>
    FLT DELT <Actual minus expected time in previous bin>
    FLT DTOWN
    FMLT D   <Difference between current              >
    BIT INIT <Initial report flag>
    INT ISGN <Direction of change   +1up, -1down          >
    BIT MODEC_FLAG
    FLT NEW_TBINMIN <Calculated TBINMIN value for high vert.rates>
    BIT OSTOSS <Spurious oscillation flag>
    FLT PREDBINS <Predicted number of bin crossings>
    FSLQGN     <100 * ISGN>
    FLT T <Time>
    FLT TBINSAVE <Pre-update value of TBIN (bin cross time)>
    FLT TN <-(time until next transition due); pos if overdue>
    FLT TPREV <Time in previous bin>
    FZLT T
    FQLHT VARNO    <Term to inhib estimating bin entered recently, after
    BHZG     <Valid-report flag (TRUE  > valid)>
    FPLZ    <Pgroted altitude at old rate   >
    IRNZPT     <Altitude report>
```

```
<***TRACKING LOCAL VARIABLES CONTINUED***>

  GROUP sensitivity
    INTERNALSL        <SL input from pilot or Mode S sensor>
ENDSTRUCTURE;
```

```
TASK TRACK_OWN;

    G.HCLOURCK;
    G.EOPTFLAGSOP;
    PERFORM Own_altitude_tracking;
    PERFORM Climb_evaluation; <determine if own aircraft can climb>
    IDF $(E) RADAROK E
        THEN, ADGAROUTAROUT
        ELSE PERFORM Radar_credibility_test;
            IDF $(E) RADARCRED E
                THEN G.RADAROUT
                ELSE, ADGAROUTAROUT
    PERFORM Ground_level_estimation; <sGROUND.        >
    PERFORM Ground_proximity_check;
    PERFORM Set_index;
    PERFORM Set_layer_dependent_parameters;
    PERFORM Update_interrogation_mode;
    PERFORM Update_advisory_mode;
 :  Set up fields for subsequent SLC Update messages to transponder
        SLC.   INDE   <Indicate current sensitivity level and TCAS capability>
        IF (QRSMODE E
            THEN 3 RI      <Onboard TCAS with vertical-only RA capability>
            ELSE IF (QRSMODE E
                    THEN 2 RI      <Onboard TCAS with RA capability inhibited>
                    ELSE 0 RI      <No onboard TCAS>

END TRACK_OWN;
```

```
PROCESS Own_altitude_tracking;

    IF (G.[illegible]LG E
        THEN CLEAR G.INITFLG;
            [illegible]P.AIRDATA E
                T[illegible]TDATAO
                    [illegible]N;
                    [illegible];                 <i[illegible] available, else use  >
                ELSE CALL VERTICAL_TRACKING
                    IN [illegible]N, O.TROWN)
                    INOUT (N entry, G.NPTR);
                    [illegible]N
                    [illegible]WN
        [illegible] IF (P.AIRDATA E
                T[illegible] G.TDATAO;
                    [illegible]N                  * DTOWN;
                    [illegible]LFAO               * [illegible]C-
                    [illegible]TAO                      * [illegible]N;
                    G [illegible];
                ELSE CALL VERTICAL_TRACKING
                    IN [illegible]N, O.TROWN)
                    INOUT (N entry, G.NPTR);
                    [illegible]N
                    [illegible]WN

END Own_altitude_tracking;
```

PROCESS Climb_evaluation;

↑ *Own-Aircraft-Operating-Modes* ▷ *Climb-Inhibit*,
↑ *Own-Aircraft-Operating-Modes* ▷ *Config-Climb-Inhibit*,
↑ *Own-Aircraft-Operating-Modes* ▷ *Increase-Climb-Inhibit*

```
    Determine if own aircraft is in regime where it can perform climb escape ma

    IF ([illegible])
        THEN CLEAR G.CLIMBINHIB;
            CLEAR G.INC_CLMINHIB;
            IF (own altitude is above aircraft altitude limit specified
                 by 735RING program pins OR performance limit discrete
                1500 fpm) indicates that own aircraft can't climb at
                THEN SET G.CLIMBINHIB;
                    SET G.INC_CLMINHIB;
                ELSE IF (performance limit discrete indicates that
                        2500 fpm) aircraft can't climb at
                        THEN SET G.INC_CLMINHIB;

END Climb_evaluation;
```

```
PROCESS Radar_credibility_test;

   Determine credibility of O.
   RADAR data;
    IF (RADAR data is credible)
    READ.
        THEN SET RADARCRED;
        ELSE CLEAR RADARCRED;

END Radar_credibility_test;
```

```
PROCESS Ground_level_estimation;

    TRUE(G.OOGROUN E
        OGROUND.
    ELSEIF (G.RADAROUT GT P.RADARLOST)
        NGROUND.
    ELSEIF (G.ORADAROUT GT
        THEN; <GROUND update          >
    ELSEIF (GROUND E
        THEN RADAR LT(P.KNOWGROL)
                GROUND.
                ELSE; <ground level unchanged>
    OTHERWISE IF (RADAR GT P.KNOWGROH)
                GROUND.
                GROUND.

END Ground_level_estimation;
```

```
PROCESS Ground_proximity_check;
```

↑ *Own-Aircraft-Operating-Modes* ▹ *Descend-Inhibit*

```
    IF (G.ON_GROUND EQ TRUE)
        THEN CLEAR G.NODESCENT;
            CLEAR G.TURN_OFF_AURALS;
        ELSE ALT_ABOVE_GND := G.ALT_ABOVE_GROUND;
            IF (G.NODESCENT EQ TRUE)
                THEN IF (ALT_ABOVE_GND GT P.NODESHI)
                        THEN CLEAR G.NODESCENT;
                        ELSE;
                ELSE IF (ALT_ABOVE_GND LT P.NODESLO)
                        THEN SET G.NODESCENT;
                        ELSE;
            IF (G.TURN_OFF_AURALS EQ TRUE)
                THEN IF (ALT_ABOVE_GND GT P.   _AURALHI)
                        THEN CLEAR G.TURN_OFF_AURALS;
                        ELSE;
                ELSE IF (ALT_ABOVE_GND LT P.   _AURALLO)
                        THEN SET G.TURN_OFF_AURALS;
                        ELSE;

END Ground_proximity_check;
```

PROCESS Set_index;

↑ *CAS-Operating-Modes* ▷ *Effective-SL*

```
    PERFORM Auto_SL; <Determine G.INDEX >
    EXTERNALSL;                <Pilot input>
    OPERATING(AUTOMATIC)
        THEN ALSL            INDE
    LOWLARGEINT;
    SIT 1;
   X);REPEAT UNTIL (SIT GT P.SITMA
        IF (GALLEVELSITE(SIT) (SIT) LT LOWLEVEL)
            THEN LOWLEVEL (SIT);
        SIT
    ENDREPEAT; <this loop selects lowest Mode S site Sens.  Level>
    IF (LOWLEVEL NE P.LARGEINT) <Site has provided an SL>
       THEN (ALSL, LOWLEVEL);
    IF (NALSL NE P.TA      _ONL
        THEN G.INDEX(G.        INDEXNALSL);
```

END Set_index;

```
PROCESS Auto_SL;
```

↑ *CAS-Operating-Modes* ▷ *Auto-SL*

```
    IF (O.OOGROUND EQ TRUE)
        THEN IF (O.GROUND_MODE EQ TRUE AND O.MANUAL NE
                THEN G.INDEX = 2
                ELSE G.INDEX = 1
    ELSEIF (O.GROUNDDESCENTCEIMBINHIB EQ TRUE
        THEN G.INDEX = 2
    ELSEIF (G.INDEX ... G.RADAROUT E              <TCAS chooses>
      ...RADARLIFT(P..
                THEN G.INDEX = 4
                ELSE G.INDEX = 2
    ELSEIF (G.INDEX = 4
        THEN IF ...G.RADAROUT ... E
                THEN G.INDEX = 2
            ELSEIF ...G.RADAROUT ... E
                THEN G.INDEX = 5
            ELSEIF ...DOWNRADAROUT GT P.RADARLOST AND G.
                THEN G.INDEX = 5
    ELSEIF (G.INDEX = 5
        THEN IF ...G.RADAROUT ... E
                THEN G.INDEX = 4
            ELSEIF (...DOWN GE P.
                THEN G.INDEX = 6
    ELSEIF (G.INDEX = 6
        THEN IF (...DOWN LE P.
                THEN G.INDEX = 5
            ELSEIF (...DOWN GE P.
                THEN G.INDEX = 7
    ELSEIF (G.INDEX = 7
        THEN IF (...DOWN LE P.
                THEN G.INDEX = 6
    ELSEIF (G.RADAROUT GT P.RADARLOST)
        THEN G.INDEX = SLNORAD;
END Auto_SL;
```

```
PROCESS Set_layer_dependent_parameters;

    IF (Q.LAYER EQ 1)
        THEN IF (G.ZDOWN GE P.TOP(1))
                THEN G.LAYER = 2;

    IF (Q.LAYER EQ 2)
        THEN IF (G.ZDOWN LE P.BOT(2))
                THEN G.LAYER = 1;
            ELSE IF (G.ZDOWN GE P.TOP(2))
                    THEN G.LAYER = 3;

    IF (Q.LAYER EQ 3)
        THEN IF (G.ZDOWN LE P.BOT(3))
                THEN G.LAYER = 2;
            ELSE IF (G.ZDOWN GE P.TOP(3))
                    THEN G.LAYER = 4;

    IF (Q.LAYER EQ 4)
        THEN IF (G.ZDOWN LE P.BOT(4))
                THEN G.LAYER = 3;

    G.PALATIM(      G.LAYER);
    G.ZHR           G.LAYER);
    G.PSESTS(FIRM        G.LAYER);

END Set_layer_dependent_parameters;
```

```
PROCESS Update_interrogation_mode;

    IF (QROPFIAGNDE G.            )DXGE
        THEN SET G.INTMODE;

        ELSE IF (GEQARSMEDDANDE G.RAMODE E
                THEN CLEAR G.INTMODE;
                    REPEAT WHILE (TRUE);LOCK E
                        <Loop while waiting for coordination lock state
                         to end.  Performance Monitor should recognize
                         when TCAS has been locked for more than P.TUNLOCK
                         seconds and take appropriate action.>
                    ENDREPEAT;
                    SET G.COLOCK using uninterruptible test and set instruction;
                    G.ECOLOCK;
                    REPEAT WHILE (more TF entries);
                        TF.INPUTRS;
                        Select next TF entry;
                    ENDREPEAT;
                    Null the ITF file and G.CREFPTR table;
                    CALL COORDINATION_UNLOCK;

<If Resolution_enabled mode was set, Update_advisory_mode must
perform cleanup this cycle.  Track file will be cleared next cycle.>

END Update_interrogation_mode;
```

```
PROCESS Update_advisory_mode;
    CLEAR G.TAMODE;
    IF (G.OPFLG EQ $TRUE)
        THEN IF (G.INDEX GE 2)
                THEN SET G.TAMODE;
                    IF ((G.INDEX EQ 2) AND (P.LOWTA EQ $FALSE))
                        THEN CLEAR G.TAMODE;
                    IF ((EXTERNALSL GT 2) AND (G.INDEX GT 2))
                        THEN SET G.RAMODE;
    IF ((G.OPFLG EQ $FALSE) OR (G.INDEX LE 2) OR (EXTERNALSL LE 2))
        THEN IF (G.RAMODE EQ $TRUE)
                THEN CLEAR G.RAMODE;
                    REPEAT WHILE (G.COLOCK EQ $TRUE);
                        <Loop while waiting for coordination lock state
                         to end.  Performance Monitor should recognize
                         when TCAS has been locked for more than P.TUNLOCK
                         seconds and take appropriate action.>
                    ENDREPEAT;
                    SET G.COLOCK using uninterruptible test and set instruction;
                    G.TLOCK = TCLOCK;
                    REPEAT WHILE (more entries in TF);
                        IF (TF.POOWRAR NE 0 OR TF.DEFER_DISPLAY EQ $TRUE)
                            THEN add threat to WL with WL.STATUS = $TERM;
                        Select next TF entry;
                    ENDREPEAT;
                    REPEAT WHILE (more entries in ITF);
                        ITF.LEVELWAIT, ITF.KHIT, ITF.PREWARN = 0;
                        ITF.TACODE = $NOTAPA;
                        CLEAR ITF.TAUCAP;
                        Select next ITF entry;
                    ENDREPEAT;
                    CALL COORDINATION_UNLOCK;
END Update_advisory_mode;
```

```
TASK TRACK_INTRUDERS;
  REPEAT WHILE (more intruders in surveillance buffer);
    IF (G.CREFPTR(S.SURVNO) EQ $NULL)
        THEN PERFORM ITF_entry_creation; <uses first two S entries>
            CALL VERTICAL_TRACKING
              IN (1, S.ZFLG, ITF.MODC, first alt rpt, S.RRTIME for rpt)
              INOUT (N entry, ITF.NPTR);
            REPEAT WHILE (more surveillance reports for this intruder);
                                   <begin with second S entry>
                PERFORM Position_tracking;
            ENDREPEAT;
        ELSE point to ITF entry with ITF.IROW = G.CREFPTR(S.SURVNO);
            PERFORM Position_tracking;
    IF (S.MODC EQ $TRUE)
        THEN ITF.IFIRM = N.CASFIRM;
            ITF.BADFOK = 0;
            ITF.ZINT = N.Z, ITF.ZDINT = N.ZD;
            ITF.ZDINR = N.ZDINNER, ITF.ZDOUTR = N.ZDOUTER;
        ELSE IF (ITF.MODC EQ $TRUE) <Mode C intruder became Non-Mode C>
                THEN clear N entry contents and delete N entry;
                    IF (ITF.KHIT GT 0 AND ITF.TACODE EQ $RA)
                        THEN SET ITF.ALTITUDE_LOST;
                    ITF.TACODE = $NOTAPA;
            ITF.NPTR = $NULL; <N entry is nonexistent in any case>
    Calculate ITF.BEARING from S.BEAR;
    ITF.BEAROK = S.BEAROK;
    ITF.MODC = S.MODC;
    PERFORM Set_arrow;
    PERFORM On_ground_test;
    Obtain ITF.EQP, ITF.PLINT from surveillance buffer;
    ITF.TDATAI = G.TCUR;
    Select next intruder;
  ENDREPEAT;
  PERFORM Drop_tracks;
END TRACK_INTRUDERS;
```

```
PROCESS ITF_entry_creation;

    < This process uses the first two S data structure entries.  >
    < References to "S" below apply to the first entry unless >
    < otherwise specified.  The second entry is reused by process >
    < Position_tracking.  >

    Create an ITF entry with ITF.IROW = unused row;
    G.CREFPTR(S.SURVNO) = ITF.IROW;
    ITF.CREFNO = S.SURVNO;
    ITF.TPTR, ITF.WPTR = $NULL;
    ITF.IDINT = S.IDINTR;
    ITF.MODC = S.MODC;
    ITF.RRTI = S.RRTIME;
    ITF.DT = Second S.RRTIME - First S.RRTIME;
    ITF.R = First S.RR;
    ITF.RD = (Second S.RR - First S.RR)/ITF.DT;
    ITF.KHIT, ITF.PREWARN, ITF.ARROW = 0;
    CLEAR ITF.DITF, ITF.TAUCAP, ITF.IOGROUN;
    ITF.TATIME = 0;
    CLEAR ITF.REVERSE, ITF.INCREASE;
    CLEAR ITF.INT_CROSS, ITF.OWN_CROSS;
    ITF.LEVELWAIT = 0;
    ITF.TAURISE = 0;
    ITF.TACODE = $NOTAPA;
    ITF.VALREP = 0;
    CLEAR ITF.CLEAR_CONFLICT, ITF.ALTITUDE_LOST, ITF.TRACK_DROP;
    ITF.LEV = 2;

END ITF_entry_creation;
```

```
PROCESS Position_tracking;

    CALL VERTICAL_TRACKING
      IN (0, S.ZFLG, ITF.MODC, S.ZRINT <if any>, S.RRTIME for report)
      INOUT (N entry, ITF.NPTR);
    ITF.DT = S.RRTIME - ITF.RRTI;
    ITF.RRTI = S.RRTIME;
    RP = ITF.R + (ITF.RD*ITF.DT);
    IF (S.RFLG EQ $TRUE)
        THEN ITF.R = RP + (P.ALFAR * (S.RR-RP));
            ITF.RD = ITF.RD + (P.BETAR * (S.RR-RP)/ITF.DT);
        ELSE ITF.R = RP;
        <ITF.RD unchanged>
    IF (ITF.R LT 0)
        THEN ITF.R = 0;
            ITF.RD = MAX(0, ITF.RD);
    ITF.RFLG = S.RFLG;
    PERFORM Valid_report_test;

END Position_tracking;
```

```
PROCESS Valid_report_test;
    IF (ITF.VALREP GT 3)
        THEN ITF.VALREP = ITF.VALREP - 4;
    ITF.VALREP = 2 * ITF.VALREP;
sp
    IF (ITF.RFLG EQ $TRUE)
        THEN ITF.VALREP = ITF.VALREP + 1;
END Valid_report_test;
```

```
PROCESS Set_arrow;
```

↑ *Supervisory-Interface* ▹ *Pilot-Displays* ▹ *Traffic-Advisories[i]* ▹ *Display-Arrow*

```
    IF (ITF.MODC EQ $FALSE)
        THEN ITF.ARROW = 0;
    ELSEIF (N.LASTRAN EQ $START, $OSCIL, or $LONGAGO)
        THEN ITF.ARROW = 0;
    ELSEIF (N.LASTRAN EQ $TREND)
        THEN ITF.ARROW = PM(ITF.ZDINT);
    OTHERWISE; <don't change arrow when LASTRAN = $GUESS>

END Set_arrow;
```

```
PROCESS On_ground_test;

    IF (ITF.MODC EQ $FALSE)
        THEN CLEAR ITF.IOGROUN;
        ELSE IF (ITF.IOGROUN EQ $TRUE)
                THEN IF (ITF.ZINT - G.ZGROUND GT P.NEARGROH)
                        THEN CLEAR ITF.IOGROUN;
                        ELSE;
                ELSE IF (ITF.ZINT - G.ZGROUND LT P.NEARGROL)
                        THEN SET ITF.IOGROUN;
                        ELSE;

END On_ground_test;
```

```
PROCESS Drop_tracks;

    REPEAT WHILE (G.COLOCK EQ $TRUE);
        <Loop while waiting for coordination lock state
         to end.  Performance Monitor should recognize
         when TCAS has been locked for more than P.TUNLOCK
         seconds and take appropriate action.>
    ENDREPEAT;
    SET G.COLOCK using uninterruptible test and set instruction;
    G.TLOCK = TCLOCK;
    REPEAT WHILE (more entries in ITF);
        IF (G.TCUR NE ITF.TDATAI)
            THEN IF (ITF.TPTR EQ $NULL OR
                    (ITF.TPTR->TF.POOWRAR EQ 0 AND
                     ITF.TPTR->TF.DEFER_DISPLAY EQ $FALSE))
                    THEN G.CREFPTR(ITF.CREFNO) = $NULL;
                        Clear all variables and flags in the N entry;
                        Delete N entry (if any) addressed by ITF.NPTR;
                        IF (ITF.TPTR NE $NULL)
                            THEN TF.IPTR = $NULL;
                        Clear all variables and flags in the ITF entry;
                        Delete ITF entry from ITF;
                    ELSE SET ITF.DITF;
                        IF (ITF.KHIT GT 0 AND ITF.TACODE EQ $RA)
                            THEN SET ITF.TRACK_DROP;
        Select next ITF entry;
    ENDREPEAT;
    CALL COORDINATION_UNLOCK;

END Drop_tracks;
```

```
ROUTINE VERTICAL_TRACKING
  IN (INIT, ZFLG, MODEC_FLAG, ZREPT, T)
  INOUT (N entry, pointer to N entry);

    IF (MODEC_FLAG EQ $TRUE)
       THEN IF (INIT EQ $TRUE OR no N entry exists for this intruder)
            THEN PERFORM Initialize_vertical_tracker;
            ELSE ZP = N.Z + (T - N.TUPDT)*N.ZD;
                DZM = ZREPT - N.ZSAVE;
                IF (DZM EQ 0)
                    THEN CREDREP = 0;
                    ELSE CREDREP = ABS(DZM - N.ZD*(T - N.TDAT)) - PN.Q *
                        (T - N.TDAT) - PN.GUESRATE * (T - N.TDAT);
                CLEAR OSTOSS;
                IF (ZFLG EQ $FALSE OR CREDREP GT 0)
                    THEN CLEAR ZFLG;
                        IF (N.TDAT EQ N.TUPDT)
                            THEN N.FIPRECO = N.CASFIRM;
                        N.CASFIRM = MAX(0, N.CASFIRM-1);
                        N.Z = ZP;
                    ELSE DBINS = ABS(DZM)/PN.Q;
                        IF (DZM EQ 0)
                            THEN PERFORM No_transition_update;
                                PERFORM No_transition_firmness;
                            ELSE PERFORM Transition_update;

                N.TUPDT = T;
                N.BINSTHISZD = MIN(N.BINSTHISZD, 10);
                IF ((ZFLG EQ $TRUE) AND (OSTOSS EQ $FALSE))
                    THEN N.TDAT = T;

END VERTICAL_TRACKING;
```

```
PROCESS Initialize_vertical_tracker;

    Obtain an unused N entry and set parameter pointer to it;

    N.Z = ZREPT;
    N.ZD = 0;
    N.TDAT = T;
    N.ZSAVE = ZREPT;
    N.TUPDT = N.TDAT;
    N.TBIN = PN.LARGEPOS;
    N.BINSTHISZD = 0;
    N.TTRAN = T;
    N.RESID = 0;
    N.DIREC, N.DIRECNZ = 0;
    N.ZDINNER = -PN.ZDLARGE;
    N.ZDOUTER = PN.ZDLARGE;
    N.CASFIRM = 0;
    N.LASTRAN = $START;
    N.TTRANGUES = 0;
    N.TSTART = T;

END Initialize_vertical_tracker;
```

```
PROCESS No_transition_update;
    TN = (T-N.TTRAN +PN.DT-N.TBIN)/PN.DT;
    IF (TN GT PN.LATELEVEL OR (T-N.TTRAN) GT PN.TGOLEV)
        THEN N.Z = ZREPT;
            N.ZD, N.BINSTHISZD, N.RESID, N.DIREC = 0;
            N.TBIN = PN.LARGEPOS;
            N.LASTRAN = $LONGAGO;
            N.ZDINNER = 0.;
            N.ZDOUTER = N.DIRECNZ * PN.Q/(T - N.TTRAN);
    ELSEIF (TN GE PN.LATESLACK + N.TTRANGUES)
        THEN T7 = N.TBIN + (0.3*N.TBIN + 0.5*PN.DT)*(TN-0.3)**2;
            N.ZD = PM(N.ZD)*ABS(PN.Q/T7);
            N.BINSTHISZD = MAX(2,N.BINSTHISZD-1);
            N.Z = ZREPT + PN.Q*N.DIRECNZ/2;
            N.ZDINNER = 0.;
            N.ZDOUTER = N.DIRECNZ * PN.Q/(T - N.TTRAN);
    OTHERWISE IF (N.LASTRAN EQ $GUESS)
                THEN N.ZD = PN.ZDDECAY*N.ZD;
                    N.TBIN = PN.Q/(ABS(N.ZD) + PN.SMALLPOS);
            IF (N.ZD GT 0)
                THEN N.Z = MIN(N.Z + N.ZD*PN.DT, ZREPT + PN.Q/2);
            ELSEIF (N.ZD LT 0)
                THEN N.Z = MAX(N.Z + N.ZD*PN.DT, ZREPT - PN.Q/2);
            IF (N.LASTRAN EQ $LONGAGO OR N.LASTRAN EQ $GUESS)
                THEN N.ZDINNER = 0.;
                    N.ZDOUTER = N.DIRECNZ * PN.Q/(T-N.TTRAN+1);
            ELSEIF (N.LASTRAN EQ $TREND AND TN GE 0)
                THEN N.ZDINNER = N.DIRECNZ * MIN(PN.Q/(T-N.TTRAN+1), ABS(N.ZDINNER));
            ELSEIF (N.LASTRAN EQ $START)
                THEN N.ZDINNER = -PN.Q/(T-N.TSTART);
                    N.ZDOUTER = -N.ZDINNER;
END No_transition_update;
```

```
PROCESS No_transition_firmness;

    IF (N.LASTRAN EQ $OSCIL)
        THEN IF (N.TDAT NE N.TUPDT)
                THEN N.CASFIRM = MAX(N.CASFIRM, N.FIPRECO);
            IF (T-N.TOSCIL GT PN.GUESDU2)
                THEN N.CASFIRM = 3;
    ELSEIF ((N.LASTRAN EQ $GUESS) OR (N.LASTRAN EQ $START))
        THEN IF (T-N.TTRAN GE PN.GUESDU1 AND T-N.TTRAN LE PN.GUESDU2)
                THEN N.CASFIRM = 1;
            ELSEIF (T-N.TTRAN GT PN.GUESDU2 AND T-N.TTRAN LE PN.GUESDU3)
                THEN N.CASFIRM = 2;
            ELSEIF (T-N.TTRAN GT PN.GUESDU3)
                THEN N.CASFIRM = 3;
            IF (N.TDAT NE N.TUPDT)
                THEN N.CASFIRM = MAX(N.FIPRECO, N.CASFIRM);
    ELSEIF (N.LASTRAN EQ $LONGAGO)
        THEN N.CASFIRM = 3;
    OTHERWISE IF (TN GT PN.OVERDU2)
                THEN N.CASFIRM = MIN(2, N.CASFIRM+1);
                    IF (N.TDAT NE N.TUPDT)
                        THEN N.CASFIRM = MAX(N.FIPRECO, N.CASFIRM);
            ELSEIF (TN GT PN.OVERDU1)
                THEN N.CASFIRM = MAX(0, N.CASFIRM-1);
            OTHERWISE IF (T-N.TTRAN GT PN.GUESDU2)
                        THEN N.CASFIRM = MAX(2, N.CASFIRM);
                    IF (N.TDAT NE N.TUPDT)
                        THEN N.CASFIRM = MAX(N.FIPRECO, N.CASFIRM);

END No_transition_firmness;
```

```
PROCESS Transition_update;

    ISGN = PM(DZM);
    QSIGN = PN.Q * ISGN;
    TBINSAVE = N.TBIN;
    TPREV = (T-N.TTRAN)/DBINS;
    IF (N.DIREC EQ ISGN)
        THEN DELT = TPREV - N.TBIN;
            PREDBINS = (T - N.TDAT)/N.TBIN;
            IF ((N.BINSTHISZD LE 0) OR (ABS(DELT) GT PN.EARLYLATE*PN.DT+N.TTRA
                OR (DBINS LT (PREDBINS-1.1)) OR (DBINS GT (PREDBINS+1.1)))
                THEN NEW_TBINMIN = MIN(PN.TBINMIN, (0.7*TPREV + 0.3*N.TBIN));
                    N.TBIN = MAX(TPREV, NEW_TBINMIN);
                    N.RESID = 0;
                    N.BINSTHISZD = 1;
                    N.ZD = QSIGN/N.TBIN;
                    N.Z = ZREPT - QSIGN/2 + N.ZD*PN.DT/2;
                ELSE N.RESID = PN.RESIDIMIN*N.RESID + DELT;
                    IF (ABS(N.RESID) LE PN.SLITEOFF+N.TTRANGUES)
                        THEN BLIM = (N.TBIN-PN.SHORT)**2 / (N.TBIN**2 + PN.SECSQ);
                            BETA1 = MAX(DBINS/(N.BINSTHISZD+DBINS), BLIM);
                            N.BINSTHISZD = N.BINSTHISZD + DBINS;
                        ELSE BETA1 = PN.TBINBETA;
                            N.BINSTHISZD = 3;
                            N.RESID = N.RESID * PN.RESIDECAY;
                    N.TBIN = N.TBIN + BETA1 * (TPREV-N.TBIN);
                    N.ZD = QSIGN/N.TBIN;
                    N.Z = ZREPT - QSIGN/2 + N.ZD*PN.DT/2;
            PERFORM Transition_firmness;
            N.LASTRAN = $TREND;
        ELSE PERFORM Unexpected_transition;
    PERFORM Transition_time_and_bin;

END Transition_update;
```

```
PROCESS Transition_firmness;

    IF (N.LASTRAN EQ $OSCIL OR N.LASTRAN EQ $GUESS)
        THEN N.CASFIRM = 2;
            N.ZDINNER = N.ZD*(T-N.TTRAN-1)/(T-N.TTRAN);
            N.ZDOUTER = PN.ZDFRACC*N.ZD*(T-N.TTRAN+1)/(T-N.TTRAN);

        ELSE PERFORM Transition_set_casfirm;
            PERFORM Transition_set_rate_limits;

END Transition_firmness;
```

```
PROCESS Transition_set_casfirm;

    IF (PN.LO3 LE (TPREV/TBINSAVE) LE PN.HI3)
        THEN N.CASFIRM = 3;
    ELSEIF (PN.LO2 LE (TPREV/TBINSAVE) LE PN.HI2)
        THEN N.CASFIRM = 2;
    ELSEIF (PN.LO1 LE (TPREV/TBINSAVE) LE PN.HI1)
        THEN N.CASFIRM = 1;
    OTHERWISE N.CASFIRM = 0;

    IF (ABS(DELT) LT PN.OUT3)
        THEN N.CASFIRM = 3;
    ELSEIF (ABS(DELT) LT PN.OUT2)
        THEN N.CASFIRM = MAX(2, N.CASFIRM);
    OTHERWISE ;

    IF (TPREV GT PN.SLACKEN2)
        THEN N.CASFIRM = MAX(2, N.CASFIRM);
    ELSEIF (TPREV GT PN.SLACKEN1)
        THEN N.CASFIRM = MAX(1, N.CASFIRM);
    OTHERWISE ;

END Transition_set_casfirm;
```

```
PROCESS Transition_set_rate_limits;

    IF (N.CASFIRM EQ 3 OR ABS(DELT) LT PN.OUT2)
        THEN N.ZDINNER = N.ZD*(N.BINSTHISZD*(T-N.TTRAN)-1)/(N.BINSTHISZD*(T-N.TTR
            N.ZDOUTER = N.ZD*(N.BINSTHISZD*(T-N.TTRAN)+1)/(N.BINSTHISZD*(T-N.TTRAN));
    ELSEIF (TPREV/TBINSAVE LT 1)
        THEN N.ZDINNER = QSIGN/TBINSAVE;
            N.ZDOUTER = PN.ZDFRACC * QSIGN/MAX(T-N.TTRAN-1, 1);
    OTHERWISE IF (N.TDAT NE N.TUPDT)
                THEN BINFIRS = N.TTRAN + (DBINS*N.TBIN) - PN.DUE2;
                    BINLAST = N.TTRAN + ((DBINS+1)*N.TBIN) + PN.DUE2;
                    IF (BINFIRS LE T AND T LE BINLAST)
                        THEN N.CASFIRM = 2;
                        ELSE N.CASFIRM = 0;
                            N.ZDINNER = 0;
                            N.ZDOUTER = ISGN * MAX(PN.Q / (T-N.TTRAN),
                            ABS(N.ZDOUTER));
                ELSE N.ZDINNER = 0;
                    N.ZDOUTER = ISGN * MAX(PN.Q/(T-N.TTRAN), ABS(N.ZDOUTER));

END Transition_set_rate_limits;
```

```
PROCESS Unexpected_transition;

    IF (N.DIREC EQ 0 OR DBINS GT 1)
        THEN N.CASFIRM = 0;
            N.LASTRAN = $GUESS;
            N.ZD = PN.GUESRATE * ISGN;
            N.Z = ZREPT - QSIGN/2 + N.ZD*PN.DT/2;
            N.TBIN = PN.Q/ABS(N.ZD);
            N.BINSTHISZD, N.RESID = 0;
            N.ZDINNER = 0;
            N.ZDOUTER = ISGN * PN.ZDLIKELY;
        ELSE IF (N.LASTRAN EQ $TREND AND (TPREV/N.TBIN) LE PN.ZRJIT)
                THEN N.CASFIRM, N.ZDINNER = 0;
                    SET OSTOSS;
                ELSE N.Z = ZREPT - (PM(DZM)*PN.Q/3);
                    N.ZD, N.BINSTHISZD, N.RESID = 0;
                    N.TBIN = PN.LARGEPOS;
                    IF (N.LASTRAN EQ $OSCIL)
                        THEN N.CASFIRM = 3;
                        ELSE N.CASFIRM = 2;
                            N.LASTRAN = $OSCIL;
                            N.TOSCIL = N.TTRAN;
                    N.ZDINNER, N.ZDOUTER = 0;

END Unexpected_transition;
```

```
PROCESS Transition_time_and_bin;

    IF (OSTOSS EQ $FALSE)
        THEN N.ZSAVE = ZREPT;
            N.TTRANGUES = 0;
            IF (N.TDAT GE N.TUPDT)
                THEN N.TTRAN = T;
                ELSE N.TTRANGUES = PN.TRANGU;
                    IF (N.DIREC EQ ISGN)
                        THEN VARNOQCH = 0.5*QSIGN*(T - N.TDAT)/DZM;
                            N.TTRAN = MIN(T-VARNOQCH, N.TTRAN+DBINS*TBINSAVE);
                            N.TTRAN = MAX(N.TDAT, N.TTRAN);
                        ELSE N.TTRAN = (T*DBINS+N.TDAT)/(DBINS+1);
            N.DIREC, N.DIRECNZ = PM(DZM);

END Transition_time_and_bin;
```

```
FUNCTION PM
  IN (ARG)
  OUT (PM);

    FLT ARG;
    INT PM;

    IF (ARG GE 0)
        THEN PM = +1;
        ELSE PM = -1;

END PM;
```

```
<***DETECT TASK LOCAL VARIABLES***>
STRUCTURE DETVAR
  GROUP settable
    FLT DMOD <Incremental range protection>
    FLT H1 <Range-range rate hyperbola threshold>
    FLT TRTHR <Range tau threshold>
    FLT TVPCMD <Max tau for VMD computation>
    FLT TVTHR <Time-to-coaltitude threshold>

  GROUP flags
    BIT CLIMB_SENSE <Climb sense selected based on modeling>
    BIT DESCEND_SENSE <Descend sense selected based on modeling>
    BIT HITFLG <Hit/miss flag>
    BIT RHIT <Range hit flag>
    BIT ZHIT <Altitude hit flag>

  GROUP calculated
    INT COUNT <Number of threats>
    FLT HMD <Range at coaltitude (hor miss distance)>
    INT VERTICAL_INTENT <Local copy of TCAS threat's vertical intent>
    FLT RDTEMP
    FLT T1
    FLT T3
```

```
<***DETECT TASK LOCAL VARIABLES CONTINUED***>

  GROUP badfirm_model
    FLT TAULIM
    FLT TRTLIM
    BITS VRAC(10) <Advisory modeled for climb sense>
    BITS VRAD(10) <Advisory modeled for descend sense>
    FLT ZCLM1 <Own altitude at TRTRU given climb>
    FLT ZCLM2 <Own altitude at TAUR given climb>
    FLT ZDES1 <Own altitude at TRTRU given descend>
    FLT ZDES2 <Own altitude at TAUR given descend>
    FLT ZDIHI <Upper vertical rate limit>
    FLT ZDILO <Lower vertical rate limit>
    FLT ZDINTC1 <Intruder altitude at TRTRU given upper-limit rate>
    FLT ZDINTC2 <Intruder altitude at TAUR given upper-limit rate>
    FLT ZDINTD1 <Intruder altitude at TRTRU given lower-limit rate>
    FLT ZDINTD2 <Intruder altitude at TAUR given lower-limit rate>
    FLT ZMPCLM <Separation resulting from climb>
    FLT ZMPDES <Separation resulting from descend>


  GROUP vmd_calc
    FLT CLIP <Ceiling on taus>
    FLT RELZ <Relative altitude>
    FLT RELZD <Relative altitude rate>
    FLT TAU1 <Tau at end of critical interval>
    FLT TAU2 <Tau at beginning of critical interval>
    FLT VMD <Vertical miss distance>
    FLT VMD1 <Vertical miss distance at end of critical interval>
    FLT VMD2 <Vertical miss distance at beginning of critical inte

ENDSTRUCTURE;
```

```
TASK DETECT_CONFLICTS;

    REPEAT WHILE (more entries in ITF);
        IF (ITF.DITF EQ $TRUE OR ITF.ALTITUDE_LOST EQ $TRUE)
            THEN add track to Working List with WL.STATUS = $TERM;
            ELSE IF (ITF.MODC EQ $TRUE)
                    THEN PERFORM Set_detection_parameters;
                        PERFORM Hit_or_miss_test;
                        PERFORM Set_up_working_list;
        Select next ITF entry;
    ENDREPEAT;

    PERFORM Test_for_multiaircraft_conflict;

END DETECT_CONFLICTS;
```

```
PROCESS Set_detection_parameters;

    ITF.LEV = MAX(G.INDEX, ITF.PLINT); <note PLINT=0 for unequipped threat>

    DMOD = P.DMODTBL(ITF.LEV);
    TVPCMD = P.TVPCTBL(ITF.LEV);
    H1 = P.H1TBL(ITF.LEV);
    IF (ITF.EQP EQ $TCAS)
        THEN TRTHR = P.TRTETBL(ITF.LEV);
            TVTHR = P.TVTETBL(ITF.LEV);
        ELSE TRTHR = P.TRTUTBL(ITF.LEV);
            TVTHR = P.TVTUTBL(ITF.LEV);

END Set_detection_parameters;
```

```
PROCESS Hit_or_miss_test;

<Test for conflict criteria ('hit') >

    IF (ITF.IOGROUN EQ $TRUE)
        THEN CLEAR HITFLG;
            ITF.PREWARN = 0;
            ITF.LEVELWAIT = 0;
            IF (ITF.KHIT EQ 1 AND ITF.TACODE EQ $RA)
                THEN SET ITF.CLEAR_CONFLICT;
        ELSE ITF.RZ = G.ZOWN - ITF.ZINT;
            ITF.RZD = G.ZDOWN - ITF.ZDINT;
            RDTEMP = ITF.RD;
            ITF.A = ABS(ITF.RZ);
            ITF.ADOT = ITF.RZD * SIGN(ITF.RZ);
            PERFORM Range_test;
            IF (ITF.RD GT 0)
                THEN ITF.VMD = ITF.RZ;
                ELSE CALL VERTICAL_MISS_DISTANCE_CALCULATION
                      IN (ITF.RZ, ITF.RZD, ITF.TRTRU, ITF.TAUR, TVPCMD)
                      OUT (ITF.VMD);
            IF (RHIT EQ $FALSE)
                THEN CLEAR HITFLG;
                    IF (ITF.KHIT EQ 1 AND ITF.TACODE EQ $RA)
                        THEN SET ITF.CLEAR_CONFLICT;
                ELSE IF (ITF.KHIT EQ 0)
                        THEN PERFORM Altitude_test;
                            IF (ZHIT EQ $FALSE)
                                THEN CLEAR HITFLG;
                                ELSE PERFORM Track_firmness_test;
                        ELSE SET HITFLG;

END Hit_or_miss_test;
```

```
PROCESS Range_test;

    IF (ITF.RD GT P.RDTHR)
        THEN ITF.TAUR, ITF.TRTRU, ITF.TAUV = P.MINTAU;
            CLEAR ITF.TAUCAP;
            IF (ITF.R * ITF.RD GT H1 OR ITF.R GT DMOD)
                THEN CLEAR RHIT;
                    IF (ITF.KHIT = 0)
                        THEN ITF.PREWARN = 0;
                            ITF.LEVELWAIT = 0;
                ELSE SET RHIT;
        ELSE IF (ITF.RD GE -P.RDTHR)
                THEN RDTEMP = -P.RDTHR;
            PERFORM Tau_calculation;
            IF (ITF.TAUR LT TRTHR AND ITF.R LE P.RMAX)
                THEN SET RHIT;
                ELSE IF (ITF.R GT DMOD OR ABS(ITF.R*ITF.RD) GT H1)
                        THEN CLEAR RHIT;
                        ELSE SET RHIT;
    IF (RHIT EQ $TRUE AND ITF.KHIT EQ 0 AND ITF.TAURISE GE 3)
        THEN CLEAR RHIT;


END Range_test;
```

```
PROCESS Tau_calculation;

    IF (ITF.R GT 0)
        THEN T1 = -(ITF.R-((DMOD**2)/ITF.R))/RDTEMP;
        ELSE T1 = P.MINTAU;
    T3 = -ITF.R/RDTEMP;
    IF (ITF.TAUCAP EQ $TRUE)
        THEN IF (T3 LT ITF.TRTRU)
                THEN ITF.TAURISE = 0;
                ELSE IF (ITF.R GT P.NAFRANGE)
                        THEN ITF.TAURISE = ITF.TAURISE + 1;
            ITF.TAUR = MIN(T1, ITF.TAUR);
            ITF.TRTRU = MIN(T3, ITF.TRTRU);
        ELSE ITF.TAUR = T1;
            ITF.TRTRU = T3;
            ITF.TAURISE = 0;
    ITF.TAUR = MAX(P.MINTAU, ITF.TAUR);
    ITF.TRTRU = MAX(P.MINTAU, ITF.TRTRU);
    SET ITF.TAUCAP;

END Tau_calculation;
```

```
PROCESS Altitude_test;

    IF (ITF.A LT G.ZTHR)
        THEN IF (ABS(ITF.VMD) LT G.ZTHR)
                THEN SET ZHIT;
                ELSE CLEAR ZHIT;

        ELSE IF (ITF.ADOT GE P.ZDTHR)
                THEN CLEAR ZHIT;
                ELSE ITF.TAUV = -ITF.A/ITF.ADOT;
                    HMD = ITF.R + ITF.RD*ITF.TAUV;
                    IF (ITF.TAUV LT TVTHR AND
                        ((ABS(ITF.VMD) LT G.ZTHR) OR
                        (ABS(HMD) LT DMOD AND ITF.TAUV LT ITF.TRTRU)))
                        THEN SET ZHIT;
                        ELSE CLEAR ZHIT;

END Altitude_test;
```

```
PROCESS Track_firmness_test;
<Range and alt.  tests passed.  Check firmness and delay 'hit' if time permit
    CLEAR HITFLG;
    IF (ITF.EQP EQ $TCAS)
        THEN VERTICAL_INTENT = 0;
            IF (ITF.TPTR NE $NULL)
                THEN REPEAT WHILE (G.COLOCK EQ $TRUE);
                        <Loop while waiting for coordination lock state
                         to end.  Performance Monitor should recognize
                         when TCAS has been locked for more than P.TUNLOCK
                         seconds and take appropriate action.>
                    ENDREPEAT;
                    SET G.COLOCK using uninterruptible test and set
                        instruction;
                    G.TLOCK = TCLOCK;
                    VERTICAL_INTENT = ITF.TPTR->TF.POTHRAR(1);
                    CALL COORDINATION_UNLOCK;
            IF (VERTICAL_INTENT NE 0)
                THEN SET HITFLG;
                ELSE IF (ITF.VALREP EQ 3 OR ITF.VALREP EQ 5 OR ITF.VALREP EQ
                        THEN IF (ITF.TAUR LT P.FRTHR(ITF.LEV, ITF.IFIRM))
                                THEN SET HITFLG;
                                    PERFORM Avoid_TCAS_TCAS_crossings;
                                    IF (HITFLG EQ $TRUE)
                                        THEN PERFORM Alt_separation_test;
                                ELSE PERFORM Model_worst_rate_errors;
                                    PERFORM Evaluate_low_firmness_separation;
            IF (HITFLG EQ $TRUE)
                THEN ITF.LEVELWAIT = 0;
        ELSE IF (ITF.TAUR LT P.FRTHR(ITF.LEV, ITF.IFIRM))
                THEN SET HITFLG;
                    PERFORM Alt_separation_test;
                ELSE PERFORM Model_worst_rate_errors;
                    PERFORM Evaluate_low_firmness_separation;
END Track_firmness_test;
```

```
PROCESS Avoid_TCAS_TCAS_crossings;

    IF (ITF.TPTR EQ $NULL)
        THEN IF ((ABS(ITF.ZDINT) GT P.OLEV) AND
                  (ABS(G.ZDOWN) LE P.OLEV))
                THEN IF ((ITF.RZ GT P.MINSEP AND ITF.VMD LE 0) OR
                          (ITF.RZ LT -P.MINSEP AND ITF.VMD GE 0))
                        THEN IF (ITF.LEVELWAIT LE P.WTTHR)
                                THEN ITF.LEVELWAIT = ITF.LEVELWAIT + 1;
                                    CLEAR HITFLG;

END Avoid_TCAS_TCAS_crossings;
```

```
PROCESS Alt_separation_test;

    IF (ABS(G.ZDOWN) LE P.OLEV AND ITF.A GT P.MAXALTDIFF)
        THEN CLEAR CLIMB_SENSE, DESCEND_SENSE;
            CALL MODEL_MANEUVERS
              IN (ITF entry)
              OUT (ZMPCLM, ZMPDES);
            IF (ZMPCLM GT ZMPDES)
                THEN SET CLIMB_SENSE;
            ELSEIF (G.CLIMBINHIB EQ $TRUE AND (ZMPCLM+P.NOZCROSS) GT ZMPDES)
                THEN SET CLIMB_SENSE;
            OTHERWISE SET DESCEND_SENSE;
            IF (ITF.RZ GE P.MINSEP)
                THEN IF (DESCEND_SENSE EQ $TRUE AND ZMPCLM GE G.ALIM)
                        THEN CLEAR DESCEND_SENSE;
                            SET CLIMB_SENSE;
            IF (ITF.RZ LE -P.MINSEP)
                THEN IF (CLIMB_SENSE EQ $TRUE AND ZMPDES GE G.ALIM)
                        THEN CLEAR CLIMB_SENSE;
                            SET DESCEND_SENSE;
            IF ((CLIMB_SENSE EQ $TRUE AND ITF.RZ LT 0) OR
                (DESCEND_SENSE EQ $TRUE AND ITF.RZ GT 0))
                THEN CLEAR HITFLG;

END Alt_separation_test;
```

```
PROCESS Model_worst_rate_errors;

    TRTLIM = MIN(ITF.TRTRU, P.TVPETBL(ITF.LEV));
    TAULIM = MIN(ITF.TAUR, P.TVPETBL(ITF.LEV));
    TAULIM = MAX(P.MINTAUM, TAULIM);
    ZDIHI = MAX(ITF.ZDINR, ITF.ZDOUTR);
    ZDILO = MIN(ITF.ZDINR, ITF.ZDOUTR);
    CLEAR All bits in VRAC, VRAD;
    IF (G.CLIMBINHIB EQ $TRUE OR G.RA(6) EQ $TRUE)
        THEN SET VRAC(2); <Don't descend>
        ELSE SET VRAC(1); <Climb>
    CALL PROJECT_OVER_INTERVAL <Chapter 6>
      IN (G.ZOWN, G.ZDOWN, VRAC, P.TV1, TRTLIM, TAULIM, ITF.REVERSE)
      OUT (ZCLM1, ZCLM2);
    IF (G.RA(1) EQ $TRUE)
        THEN SET VRAD(7); <Don't climb>
        ELSE SET VRAD(6); <Descend>
    CALL PROJECT_OVER_INTERVAL
      IN (G.ZOWN, G.ZDOWN, VRAD, P.TV1, TRTLIM, TAULIM, ITF.REVERSE)
      OUT (ZDES1, ZDES2);
    CALL PROJECT_OVER_INTERVAL
      IN (ITF.ZINT, ZDIHI, 0, 0, TRTLIM, TAULIM, ITF.REVERSE)
      OUT (ZDINTC1, ZDINTC2);
    CALL PROJECT_OVER_INTERVAL
      IN (ITF.ZINT, ZDILO, 0, 0, TRTLIM, TAULIM, ITF.REVERSE)
      OUT (ZDINTD1, ZDINTD2);
    CALL SEPARATION_OVER_INTERVAL <Chapter 6>
      IN (ZCLM1, ZCLM2, ZDINTC1, ZDINTC2, ZDINTD1, ZDINTD2, ZDES1, ZDES2)
      OUT (ZMPCLM, ZMPDES);

END Model_worst_rate_errors;
```

```
PROCESS Evaluate_low_firmness_separation;

    IF (MAX(ZMPCLM, ZMPDES) LE G.SENSFIRM)
        THEN CLEAR HITFLG;
            ITF.PREWARN = ITF.PREWARN + 1;
        ELSE SET HITFLG;

            IF (ZMPCLM GT ZMPDES)
                THEN ITF.BADFOK = +1;
                    ITF.SECH = ZMPDES;
                ELSE IF (G.CLIMBINHIB EQ $TRUE AND (ZMPCLM+P.NOZCROSS) GT ZMPDES)
                        THEN ITF.BADFOK = +1;
                            ITF.SECH = ZMPDES;
                        ELSE ITF.BADFOK = -1;
                            ITF.SECH = ZMPCLM;
            IF (((ITF.BADFOK EQ 1) AND (ITF.RZ LT -P.LOWFIRMRZ))
                OR ((ITF.BADFOK EQ -1) AND (ITF.RZ GT P.LOWFIRMRZ)))
                THEN CLEAR HITFLG;
                    ITF.PREWARN = ITF.PREWARN + 1;
                    ITF.BADFOK = 0;
                    ITF.SECH = 0;

END Evaluate_low_firmness_separation;
```

```
PROCESS Set_up_working_list;

    IF (HITFLG EQ $TRUE)
        THEN ITF.TACODE = $RA;
            IF (ITF.KHIT EQ 0)
                THEN add to list with WL.STATUS = $NEW;
                ELSE add to list with WL.STATUS = $CONT:
            ITF.KHIT = 3;
        ELSE ITF.TACODE = $NOTAPA;
            IF (ITF.KHIT GT 0)
                THEN IF (ITF.KHIT EQ 1)
                        THEN IF (G.TCUR - ITF.TCMD GT P.TMIN-1)
                                THEN add to list with WL.STATUS = $TERM;
                                    ITF.KHIT = 0;
                                ELSE add to list with WL.STATUS = $CONT;
                                    ITF.TACODE = $RA;
                                    CLEAR ITF.CLEAR_CONFLICT;
                        ELSE add to list with WL.STATUS = $CONT;
                            ITF.KHIT = 1;
                            ITF.TACODE = $RA;

    < Note:  when entry added to WL, ITF.WPTR should be set to its address.  >
    < Entries must be ordered with WL.STATUS = $TERM first, = $NEW next, >
    < and $CONT last.  >

END Set_up_working_list;
```

```
PROCESS Test_for_multiaircraft_conflict;

    CLEAR G.MACFLG;
    COUNT = 0;
    REPEAT WHILE (more entries on Working List);
        IF (WL.STATUS EQ $NEW OR WL.STATUS EQ $CONT)
            THEN COUNT = COUNT + 1;
        Select next WL entry;
    ENDREPEAT;

    IF (COUNT GT 1)
        THEN SET G.MACFLG;

END Test_for_multiaircraft_conflict;
```

```
ROUTINE VERTICAL_MISS_DISTANCE_CALCULATION
  IN (RELZ, RELZD, TAU1, TAU2, CLIP)
  OUT (VMD);

<TAU1 is the TRTRU argument.  TAU2 is the TAUR argument.>

    VMD1 = RELZ + RELZD*MIN(CLIP,TAU1);
    VMD2 = RELZ + RELZD*MIN(CLIP,TAU2);
    IF (VMD1 * VMD2 LE 0)
        THEN VMD = 0;
    ELSEIF (VMD1 GT 0)
        THEN VMD = MIN(VMD1, VMD2);
    OTHERWISE VMD = MAX(VMD1, VMD2);

END VERTICAL_MISS_DISTANCE_CALCULATION;
```

```
STRUCTURE RESVAR
<***RESOLUTION LOCAL VARIABLES***>
  GROUP sense
    FLT ALT_DIFF <Difference between intruder's projected and
    own's current altitude>
    INT COMPLEMENT <Numeric value of complement of intent>
    FLT DELAYC <Pilot delay modeled for climb sense>
    FLT DELAYD <Pilot delay modeled for descend sense>
    INT DIRECTION
    FLT OWNLEV
    FLT RELALT
    BIT RVSFLAG
    BIT SKIP_CALC
    FLT TACC <Time to accelerate>
    FLT TAGR <Time at goal rate>
    FLT TAIR <Time at initial rate (delay time)>
    FLT TAULIM
    FLT TAUM
    FLT TDGR
    FLT TESC
    FLT TRTLIM <Time to closest approach (with ceiling)>
    FLT TUGR
    FLT VMDO
    BITS VRAC(10)
    BITS VRAD(10)
    FLT ZDACC
    FLT ZDGOAL
    FLT ZDMAX
    FLT ZDMIN
    FLT ZMPCLM <Separation given climb sense>
    FLT ZMPDES <Separation given descend sense>
    FLT ZMPSECH <Separation given second-choice sense>
    FLT ZPINT
    FLT ZPOWN
    FLT ZPROJ
```

```
<***RESOLUTION LOCAL VARIABLES CONTINUED***>
  GROUP select_advisory
    FLT ALIMOD <Modification to ALIM (hysteresis)>
    FLT AVSL
    BIT CONSIDER_INCREASE <Flag set by reversal logic to consider increase rat
    BIT INTHR <TRUE => separation falls within given threshold>
    FLT NOWEAKEN_TIME <Time advisory not allowed to weaken>
    FLT OLDGOAL
    BITS OWNTENT(12) <Advisory for WL (input) threat>
    FLT POSITN
    FLT RELZ
    FLT RELZD
    FLT SEP
    FLT THRHLD
    FLT VO
    BIT VSLOK <TRUE => given VSL or neg RA is safe>
    FLT VVMD
    FLT ZDOW
  GROUP multiaircraft
    BIT ABOVE <TRUE => intruder to pass above own (DESC sense)>
    BIT BELOW <TRUE => intruder to pass below own (CLIMB sense)>
    INT MACNPTR(20)
    INT MACOPTR(20)
    INT N
    INT NUMBPOI <Number of advisory changes due to multiaircraft logi
    BIT SENSE
    BIT SENSE_SET
    BIT TYPE <TRUE => threats are resolved in both senses>
  GROUP threat_file_update
    INT IND1
    INT IND2
    INT OLDPOI <Advisory to delete>
    INT OPTR <Advisory to add>
    INT RAIND
    BIT SUCCESS
```

```
<***RESOLUTION LOCAL VARIABLES CONTINUED***>
  GROUP coordination
    INT SIT
    INT TRY <Number of failed coordination attempts>
  GROUP modeling
    FLT ACCEL <Vertical acceleration to model>
    FLT A1
    FLT A2
    FLT A3
    FLT A4
    FLT A5
    FLT A6
    FLT A7
    FLT A8
    FLT DELAY
    FLT HVEQ
    FLT PROJ_ZDINT <Int.  projected alt.  at CPA using ZDINT>
    FLT TAUARG
    FLT TPROJ
    FLT TV <Pilot response delay time to model>
    FLT T1
    FLT VARG
    FLT VPROJ
    BITS VRA(10)
    FLT Z
    FLT ZCLM1
    FLT ZCLM2
    FLT ZD
    FLT ZDES1
    FLT ZDES2
    FLT ZDI
    FLT ZPIN1
    FLT ZPIN2
```

```
<***RESOLUTION LOCAL VARIABLES CONTINUED***>

    FLT ZPR1
    FLT ZPR2

ENDSTRUCTURE;
```

```
TASK RESOLUTION_AND_COORDINATION
  IN (WL entry);
    Pointer to ITF = WL.IPTR, pointer to TF = ITF.TPTR;
    REPEAT WHILE (G.COLOCK EQ $TRUE);
        <Loop while waiting for coordination lock state to end.  Performance
         Monitor should recognize when TCAS has been locked for more than
         P.TUNLOCK seconds and take appropriate action.>
    ENDREPEAT;
    SET G.COLOCK using uninterruptible test and set instruction;
    G.TLOCK = TCLOCK;
    IF (WL.STATUS EQ $TERM)
        THEN PERFORM Update_threat_file_own;
            CALL DELETE_RESOLUTION_ADVISORY
              IN (OLDPOI);
            CLEAR ITF.REVERSE, ITF.INCREASE;
    ELSEIF (WL.STATUS EQ $NEW)
        THEN CLEAR All bits in OWNTENT;
            SET OWNTENT(4);
            PERFORM New_threat_file_entry;
            PERFORM Select_sense;
    OTHERWISE OWNTENT = TF.PERMTENT;
    IF (WL.STATUS EQ $NEW OR WL.STATUS EQ $CONT)
        THEN PERFORM Process_new_or_continuing_threat;
    IF (WL.STATUS EQ $TERM OR TF.DEFER_DISPLAY EQ $FALSE)
        THEN send Coordination Update msg to transponder with:  ARA=G.RA,RAC=G.INT
    IF (G.OPFLG EQ $TRUE AND ITF.EQP EQ $TCAS)
        THEN PERFORM Send_initial_intent;
    CALL COORDINATION_UNLOCK
    IF (G.OPFLG EQ $TRUE AND ITF.EQP EQ $TCAS)
        THEN PERFORM Complete_send_intent;
END RESOLUTION_AND_COORDINATION;
```

```
PROCESS Update_threat_file_own;

    IF (WL.STATUS EQ $TERM)
        THEN OWNTENT = TF.PERMTENT;
            OLDPOI = TF.POOWRAR;
            OPTR = 0;
            IF (TF.POTHRAR(1) NE 0 OR TF.POTHRAR(2) NE 0)
                THEN TF.POOWRAR = 0;
                    CLEAR All bits in TF.PERMTENT;
                    TF.TLRCMD = G.TCUR;
                ELSE ITF.TPTR = $NULL;
                    Clear all variables and flags in TF entry
                        and delete TF entry;
        ELSE IF (OWNTENT NE TF.PERMTENT)
                THEN TF.PERMTENT = OWNTENT;
                    ITF.TCMD = G.TCUR;
            OPTR = RAMAP(OWNTENT);
            OLDPOI = TF.POOWRAR;
            TF.POOWRAR = OPTR;
            TF.TLRCMD = G.TCUR;

END Update_threat_file_own;
```

```
PROCESS New_threat_file_entry;

    CLEAR SUCCESS;
    IF (ITF.EQP NE $ATCRBS)
        THEN REPEAT WHILE (more entries in TF AND SUCCESS EQ $FALSE);
                IF (ITF.IDINT EQ TF.ID)
                    THEN SET SUCCESS;
                    ELSE select next TF entry;
            ENDREPEAT;
    IF (SUCCESS EQ $FALSE)
        THEN create new TF entry;
            TF.ID = ITF.IDINT;
            TF.POOWRAR, TF.POTHRAR(1), TF.POTHRAR(2) = 0;
            TF.TTHLRCM = P.TINIT;
            CLEAR TF.DEFER_DISPLAY;
            TF.DEFER_COUNT = 0;
            CLEAR All bits in TF.TEMPRA;
            CLEAR All bits in TF.PERMTENT;
    TF.IPTR = ITF.IROW;
    TF.TLRCMD = G.TCUR;
    SET TF.NEW;
    CLEAR ITF.REVERSE, ITF.INCREASE, ITF.INC_ENC, ITF.TIEBREAKER_REVERSAL;
    ITF.INCTIME = 0;
    ITF.TCMD = G.TCUR;
    ITF.TPTR = address of TF entry;

END New_threat_file_entry;
```

```
PROCESS Select_sense;
    IF (ITF.EQP EQ $TCAS AND TF.POTHRAR(1) EQ 1 or 2)
        THEN PERFORM Form_complement; <complement of threat's sense>
            CLEAR ITF.DCFLG;
        ELSE IF (ITF.BADFOK EQ 0)
                THEN CALL MODEL_MANEUVERS
                      IN (ITF entry)
                      OUT (ZMPCLM, ZMPDES);
                    IF (ZMPCLM GT ZMPDES)
                        THEN CLEAR OWNTENT(7);
                            ZMPSECH = ZMPDES;
                    ELSEIF (G.CLIMBINHIB EQ $TRUE AND (ZMPCLM+P.NOZCROSS) GT ZMPDES)
                        THEN CLEAR OWNTENT(7);
                            ZMPSECH = ZMPDES;
                    OTHERWISE SET OWNTENT(7);
                            ZMPSECH = ZMPCLM;
                    IF (ITF.RZ GE P.MINSEP)
                        THEN IF ((OWNTENT(7) EQ $TRUE) AND (ZMPCLM GE G.ALIM))
                                THEN CLEAR OWNTENT(7);
                                    ZMPSECH = ZMPDES;
                    IF (ITF.RZ LE -P.MINSEP)
                        THEN IF ((OWNTENT(7) EQ $FALSE) AND (ZMPDES GE G.ALIM))
                                THEN SET OWNTENT(7);
                                    ZMPSECH = ZMPCLM;
            ELSEIF (ITF.BADFOK EQ 1)
                THEN CLEAR OWNTENT(7);
                    ZMPSECH = ITF.SECH;
            OTHERWISE SET OWNTENT(7);
                    ZMPSECH = ITF.SECH;
            IF (ITF.EQP EQ $TCAS)
                THEN PERFORM TCAS_threat_processing;
                ELSE PERFORM Don't_care_test;
END Select_sense;
```

```
PROCESS Form_complement;

    COMPLEMENT = TF.POTHRAR(1) - 1;
    IF (COMPLEMENT EQ 1)
        THEN SET OWNTENT(7);
        ELSE CLEAR OWNTENT(7);

END Form_complement;
```

```
PROCESS TCAS_threat_processing;

    CLEAR ITF.DCFLG;
    IF (G.IDOWN GT ITF.IDINT)
        THEN SET TF.DEFER_DISPLAY;
            TF.DEFER_COUNT = 0;
            ITF.TACODE = $TAMC;
            TF.TEMPRA = OWNTENT;
            TF.POOWRAR = 0;
            CLEAR All bits in TF.PERMTENT;

END TCAS_threat_processing;
```

```
PROCESS Don't_care_test;

    IF (ZMPSECH GE G.ALIM)
        THEN SET ITF.DCFLG;
        ELSE CLEAR ITF.DCFLG;
            IF ( (G.RA(1) EQ $TRUE AND OWNTENT(7) EQ $TRUE)
            OR (G.RA(6) EQ $TRUE AND OWNTENT(7) EQ $FALSE))
                THEN OWNLEV = G.ZOWN + P.OWNDEL*G.ZDOWN;
                    Save current TF pointer;
                    REPEAT WHILE (more entries in TF AND ITF.DCFLG EQ $FALSE);
                        IF (TF.PERMTENT(4,5) EQ '10' AND TF.PERMTENT(7) NE
                            OWNTENT(7) AND TF.IPTR NE $NULL)
                            THEN RELALT = OWNLEV - TF.IPTR->ITF.ZINT;
                                TAUM = MIN(MAX(ITF.TAUR,P.MINTAUM), ITF.TRTRU);
                                CALL VERTICAL_MISS_DISTANCE_CALCULATION <Chap 5>
                                  IN (RELALT, TF.IPTR->ITF.ZDINT, TAUM,
                                  ITF.TRTRU, P.TVPETBL(ITF.LEV))
                                  OUT (VMDO);
                                IF (ABS(VMDO) LT ZMPSECH)
                                    THEN SET ITF.DCFLG;
                        Select next TF entry;
                    ENDREPEAT;
                    Restore current TF pointer;

END Don't_care_test;
```

```
PROCESS Process_new_or_continuing_threat;

    IF (WL.STATUS EQ $CONT AND ITF.KHIT EQ 3)
        THEN PERFORM Reversal_check;
    IF (TF.DEFER_DISPLAY EQ $FALSE)
        THEN PERFORM Select_advisory;
            IF (G.MACFLG EQ $TRUE)
                THEN PERFORM Multiaircraft_processing;
                ELSE PERFORM Update_threat_file_own;
                    Save current TF pointer;
                    CALL RESOLUTION_UPDATE
                        IN (OLDPOI, OPTR);
                    Restore current TF pointer;
            IF (WL.STATUS EQ $CONT AND ITF.KHIT EQ 3)
                THEN PERFORM Increase_check;
        ELSE TF.TLRCMD = G.TCUR;

END Process_new_or_continuing_threat;
```

```
PROCESS Reversal_check;

    CLEAR CONSIDER_INCREASE;
    IF (ITF.EQP EQ $TCAS)
        THEN IF (G.IDOWN GT ITF.IDINT)
            THEN IF (TF.DEFER_DISPLAY EQ $TRUE)
                    THEN OWNTENT = TF.TEMPRA;
                        TF.DEFER_COUNT = TF.DEFER_COUNT + 1;
                IF ((TF.POTHRAR(1) EQ 1 AND OWNTENT(7) EQ $TRUE) OR
                      (TF.POTHRAR(1) EQ 2 AND OWNTENT(7) EQ $FALSE))
                    THEN PERFORM Form_complement;
                        SET ITF.TIEBREAKER_REVERSAL;
                        IF (TF.DEFER_DISPLAY EQ $FALSE)
                            THEN SET ITF.REVERSE;
                IF ((TF.DEFER_DISPLAY EQ $TRUE AND (TF.DEFER_COUNT GT
                      P.WTTHR OR TF.POTHRAR(1) NE 0))
                      OR (ITF.REVERSE EQ $TRUE))
                    THEN PERFORM Set_up_for_advisory;
                        CLEAR TF.DEFER_DISPLAY;
                IF (TF.DEFER_DISPLAY EQ $TRUE)
                    THEN ITF.TACODE = $TAMC;
        ELSE IF (ITF.TRTRU GT P.MIN_RI_TIME AND ITF.TAURISE LT 3)
                THEN IF (ITF.INC_ENC EQ $FALSE
                          AND (ITF.INT_CROSS EQ $TRUE OR ITF.OWN_CROSS EQ $TRUE))
                        THEN PROJ_ZDINT = ITF.ZINT + (ITF.TRTRU * ITF.ZDINT);
                            PERFORM Reversal_proj_check;
                            IF (ITF.REVERSE EQ $FALSE AND
                                  ITF.TRTRU LE P.MINRVSTIME)
                                THEN SET CONSIDER_INCREASE;
                        ELSE PERFORM Cross_through_check;

END Reversal_check;
```

```
PROCESS Set_up_for_advisory;

    CLEAR OWNTENT(5,6,11,12);
    TF.PERMTENT = OWNTENT;
    ITF.TCMD = G.TCUR;
    IF (ITF.EQP NE $TCAS)
        THEN PERFORM Dont_care_test;

END Set_up_for_advisory;
```

```
PROCESS Reversal_proj_check;
    IF ((OWNTENT(7) EQ $FALSE AND ITF.RZ LE -P.AVEVALT) OR
          (OWNTENT(7) EQ $TRUE AND ITF.RZ GE P.AVEVALT) OR (ITF.TRTRU GT
          P.MINRVSTIME AND ((OWNTENT(7) EQ $FALSE AND ITF.RZ LE -P.CROSST
          OR (OWNTENT(7) EQ $TRUE AND ITF.RZ GE P.CROSSTHR))))
        THEN IF (ITF.INT_CROSS EQ $TRUE)
                THEN IF ((OWNTENT(7) EQ $FALSE AND G.ZOWN LT PROJ_ZDINT)
                          OR (OWNTENT(7) EQ $TRUE AND G.ZOWN GT PROJ_ZDINT))
                        THEN SET ITF.REVERSE;
            ELSEIF (ITF.OWN_CROSS EQ $TRUE)
                THEN IF ((OWNTENT(7) EQ $FALSE AND G.ZOWN GT PROJ_ZDINT)
                          OR (OWNTENT(7) EQ $TRUE AND G.ZOWN LT PROJ_ZDINT))
                        THEN CLEAR ITF.OWN_CROSS, SET ITF.INT_CROSS;
                        ELSE IF ((G.MACFLG EQ $FALSE) OR
                                  (ABS(ITF.ZDINT) GE P.OLEV AND
                                  (G.ZDMODEL * ITF.ZDINT) LT 0))
                                THEN CALL MODEL_MANEUVERS
                                      IN (ITF entry)
                                      OUT (ZMPCLM,ZMPDES);
                                    IF (ZMPCLM GT ZMPDES)
                                        THEN IF (OWNTENT(7) EQ $TRUE AND
                                            ZMPCLM GT (ZMPDES+P.NOZCROSS))
                                                THEN SET ITF.REVERSE;
                                    ELSEIF (G.CLIMBINHIB EQ $TRUE AND
                                        (ZMPCLM+P.NOZCROSS) LT ZMPDES)
                                        THEN IF (OWNTENT(7) EQ $FALSE AND
                                            ZMPDES GT (ZMPCLM+2*P.NOZCROSS))
                                                THEN SET ITF.REVERSE;
                                    OTHERWISE IF (OWNTENT(7) EQ $FALSE AND
                                            ZMPDES GT (ZMPCLM+P.NOZCROSS))
                                                THEN SET ITF.REVERSE;
            OTHERWISE;
            IF (ITF.REVERSE EQ $TRUE)
                THEN PERFORM Reversal_modeling;
END Reversal_proj_check;
```

```
PROCESS Reversal_modeling;

    CALL MODEL_MANEUVERS
      IN (ITF entry)
      OUT (ZMPCLM,ZMPDES);
    IF (OWNTENT(7) EQ $FALSE AND ZMPDES GT 0)
        THEN ZMPSECH = ZMPCLM;
            SET OWNTENT(7);
            PERFORM Set_up_for_advisory;
    ELSEIF (OWNTENT(7) EQ $TRUE AND ZMPCLM GT 0)
        THEN ZMPSECH = ZMPDES;
            CLEAR OWNTENT(7);
            PERFORM Set_up_for_advisory;
    OTHERWISE CLEAR ITF.REVERSE;

END Reversal_modeling;
```

```
PROCESS Cross_through_check;

    IF (ITF.INT_CROSS EQ $FALSE AND ITF.OWN_CROSS EQ $FALSE)
        THEN IF ((OWNTENT(5,6,7) EQ '000' AND ITF.RZ LT -P.CROSSTHR)
                  OR (OWNTENT(5,6,7) EQ '001' AND ITF.RZ GT P.CROSSTHR))
                THEN CLEAR ITF.INCREASE;
                    CLEAR ITF.INC_ENC;
                    ITF.INCTIME = 0;
                    IF (OWNTENT(7) EQ $TRUE)
                        THEN CLEAR OWNTENT(7);
                        ELSE SET OWNTENT(7);
                    SET ITF.REVERSE;
                    PERFORM Set_up_for_advisory;

END Cross_through_check;
```

```
PROCESS Select_advisory;

    IF ((ITF.KHIT EQ 1) OR (WL.STATUS EQ $CONT AND ITF.TRTRU LE P.QU
        THEN OWNTENT = TF.PERMTENT;
        ELSE IF (ABS(ITF.ZDINT) LT P.ILEV)
                THEN CALL CHECK_PROJECTION
                      IN (ITF.RZ, G.ALIM, OWNTENT)
                      OUT (INTHR);
                    IF (INTHR EQ $FALSE)
                        THEN PERFORM Try_vsl;
                        ELSE CALL CHECK_PROJECTION
                              IN (ITF.VMD, G.ALIM, OWNTENT)
                              OUT (INTHR);
                            IF (INTHR EQ $FALSE OR ABS(G.ZDOWN) GT P.OLEV)
                                THEN PERFORM Try_vsl;
                                ELSE CLEAR OWNTENT(5,6,11,12);
                ELSE CALL CHECK_PROJECTION
                      IN (ITF.VMD, G.ALIM, OWNTENT)
                      OUT (INTHR);
                    IF (INTHR EQ $FALSE OR ABS(G.ZDOWN) GT P.OLEV)
                        THEN PERFORM Try_vsl;
                        ELSE CLEAR OWNTENT(5,6,11,12);
            PERFORM No_weaken_test;
            PERFORM Extreme_altitude_check;

END Select_advisory;
```

```
PROCESS Try_vsl;

   IF (ITF.RD GT 0)
    THEN IF (WL.STATUS EQ $NEW)
          THEN OWNTENT(5,6,11,12) = '10','00';
          ELSE OWNTENT = TF.PERMTENT;
    ELSE CALL VSL_OVER_INTERVAL
          IN (P.V2000, ITF entry)
          OUT (VSLOK);
        IF (VSLOK EQ $TRUE)
            THEN OWNTENT(5,6,11,12) = '11','11';
            ELSE CALL VSL_OVER_INTERVAL
                  IN (P.V1000, ITF entry)
                  OUT (VSLOK);
                IF (VSLOK EQ $TRUE)
                    THEN OWNTENT(5,6,11,12) = '11','10';
                    ELSE CALL VSL_OVER_INTERVAL
                          IN (P.V500, ITF entry)
                          OUT (VSLOK);
                        IF (VSLOK EQ $TRUE)
                            THEN OWNTENT(5,6,11,12) = '11','01';
                            ELSE CALL VSL_OVER_INTERVAL
                                  IN (0, ITF entry)
                                  OUT (VSLOK);
                                IF (VSLOK EQ $TRUE)
                                    THEN OWNTENT(5,6,11,12) = '10','00';
                                    ELSE OWNTENT(5,6,11,12) = '00','00';

END Try_vsl;
```

```
PROCESS No_weaken_test;

    IF (WL.STATUS NE $NEW)
        THEN IF (ITF.REVERSE EQ $TRUE)
                THEN NOWEAKEN_TIME = P.TRVSNOWEAK;
                ELSE NOWEAKEN_TIME = P.TNOWEAK;
            IF (EVAL(TF.PERMTENT) LE EVAL(OWNTENT))
                THEN IF (ITF.TAUR GT MAX(P.STROFIR, P.FRTHR(ITF.LEV,ITF.IFIRM))
                        THEN OWNTENT = TF.PERMTENT;
            ELSEIF (TF.PERMTENT(4,5) EQ '10')
                THEN CALL CHECK_PROJECTION
                      IN (ITF.RZ, G.ALIM, OWNTENT)
                      OUT (INTHR);
                    IF (INTHR EQ $TRUE)
                        THEN OWNTENT = TF.PERMTENT;
                        ELSE IF ((G.TCUR-ITF.TCMD) LT NOWEAKEN_TIME
                                OR ITF.IFIRM LT P.MINFIRM)
                                THEN OWNTENT = TF.PERMTENT;
            OTHERWISE IF ((G.TCUR-ITF.TCMD) LT NOWEAKEN_TIME OR
                        ITF.IFIRM LT P.MINFIRM)
                        THEN OWNTENT = TF.PERMTENT;

END No_weaken_test;
```

```
PROCESS Extreme_altitude_check;

    IF (OWNTENT(7) EQ $FALSE AND OWNTENT(5) EQ $FALSE)
        THEN IF ((G.CLIMBINHIB EQ $TRUE) AND (ITF.REVERSE = $FALSE))
                THEN OWNTENT(5,6,11,12) = '10','00';

    ELSEIF (OWNTENT(7) EQ $TRUE AND OWNTENT(5) EQ $FALSE)
        THEN IF (G.NODESCENT EQ $TRUE)
                THEN OWNTENT(5,6,11,12) = '10','00';
                    CLEAR ITF.REVERSE;

END Extreme_altitude_check;
```

```
PROCESS Multiaircraft_processing;

    PERFORM Multiaircraft_resolution;
    PERFORM Multiaircraft_threat_file_update;
    N = 1;
    Save current TF pointer;
    REPEAT UNTIL (N GT NUMBPOI);
        CALL RESOLUTION_UPDATE
          IN (MACOPTR(N), MACNPTR(N));
        N = N + 1;
    ENDREPEAT;
    Restore current TF pointer;

END Multiaircraft_processing;
```

```
PROCESS Multiaircraft_resolution;

    CLEAR ABOVE, BELOW;
    Save current TF pointer;
    REPEAT WHILE (more TF entries); <loop through entire TF>
        SET SENSE_SET;
        IF (TF.PERMTENT(4) EQ $TRUE) <vertical RA>
            THEN SENSE = TF.PERMTENT(7) using threat's TF entry;
        ELSEIF (TF threat is same threat as current threat>
            THEN SENSE = OWNTENT(7);
        OTHERWISE CLEAR SENSE_SET;

        IF (SENSE_SET EQ $TRUE)
            THEN IF (TF.IPTR->ITF.DCFLG EQ $TRUE)
                    THEN ;
                    ELSE IF (SENSE EQ $TRUE)
                            THEN SET ABOVE;
                            ELSE SET BELOW;
        Select next TF entry;
    ENDREPEAT;
    Restore current TF pointer;

END Multiaircraft_resolution;
```

```
PROCESS Multiaircraft_threat_file_update;

    TF.PERMTENT = OWNTENT;
    NUMBPOI = 0;
    IF (ABOVE EQ $TRUE AND BELOW EQ $TRUE)
        THEN SET TYPE;
        ELSE CLEAR TYPE;
            IF (ABOVE EQ $FALSE AND BELOW EQ $FALSE)
                THEN SENSE = TF.PERMTENT(7);
                ELSE SENSE = ABOVE;
    PERFORM Multiaircraft_loop_on_threat_file;
    OWNTENT = TF.PERMTENT;

END Multiaircraft_threat_file_update;
```

```
PROCESS Multiaircraft_loop_on_threat_file;

    Save current Working List pointer;
    Save current TF pointer;
    REPEAT WHILE (more threats in TF); <loop through entire TF>
      IF (TF.PERMTENT(4) EQ $TRUE) <provision for future horizontal type
        THEN IF (TYPE NE TF.PERMTENT(5))
                THEN TF.PERMTENT(5) = TYPE;
                    CLEAR TF.PERMTENT(6,11,12);
            IF (TYPE EQ $FALSE AND SENSE NE TF.PERMTENT(7))
                THEN TF.PERMTENT(7) = SENSE;
                    CLEAR TF.PERMTENT(6,11,12);
            IF ((TYPE EQ $FALSE) AND ((TF.PERMTENT(7) EQ $FALSE AND
                  G.CLIMBINHIB EQ $TRUE) OR (TF.PERMTENT(7) EQ $TRUE AND
                  G.NODESCENT EQ $TRUE)))
                THEN SET TF.PERMTENT(5);
            IF (TF.POOWRAR NE RAMAP(TF.PERMTENT))
                THEN NUMBPOI = NUMBPOI + 1;
                    MACOPTR(NUMBPOI) = TF.POOWRAR;
                    TF.POOWRAR = RAMAP(TF.PERMTENT);
                    MACNPTR(NUMBPOI) = TF.POOWRAR;
                    IF (TF threat is in Working List)
                        THEN TF.IPTR -> ITF.TCMD = G.TCUR;
            IF (TF threat is in Working List)
                THEN TF.TLRCMD = G.TCUR;
        Select next TF entry;
    ENDREPEAT;
    Restore current Working List pointer;
    Restore current TF pointer;

END Multiaircraft_loop_on_threat_file;
```

```
PROCESS Increase_check;
    IF ((OWNTENT(5,6,7) EQ '000' AND G.CLSTROLD EQ 8) OR
        (OWNTENT(5,6,7) EQ '001' AND G.DESTROLD EQ 8))
        THEN IF (ITF.TAURISE LT 3 AND
                ((ITF.EQP EQ $TCAS) OR (CONSIDER_INCREASE EQ TRUE) OR
                (ITF.INT_CROSS EQ $FALSE AND ITF.OWN_CROSS EQ $FALSE)))
                THEN CLEAR INTHR;
                    IF (CONSIDER_INCREASE EQ $TRUE)
                        THEN CLEAR CONSIDER_INCREASE;
                            ALT_DIFF = PROJ_ZDINT - G.ZOWN;
                            IF (OWNTENT(7) EQ $TRUE AND ALT_DIFF LE P.AVEVALT)
                                THEN SET INTHR;
                            ELSEIF (OWNTENT(7) EQ $FALSE AND ALT_DIFF GE -P.AVEVALT)
                                THEN SET INTHR;
                        ELSE PERFORM Increase_proj_check;
                    IF (INTHR EQ $TRUE)
                        THEN IF (ITF.INCREASE EQ $TRUE OR
                                 ITF.TRTRU GT P.MIN_RI_TIME)
                                THEN IF ((OWNTENT(7) EQ $TRUE AND
                                          G.RADAROUT EQ 0 AND
                                          O.ZRADAR LE P.ZNO_INCDES) OR
                                          (OWNTENT(7) EQ $FALSE AND
                                          G.INC_CLMINHIB EQ $TRUE))
                                        THEN;
                                        ELSE ITF.INCTIME = G.TCUR;
                                            IF (ITF.INCREASE EQ $FALSE)
                                                THEN ITF.TCMD = G.TCUR;
                                            SET ITF.INCREASE, ITF.INC_ENC;
            IF (ITF.INCREASE EQ $TRUE AND (G.TCUR - ITF.INCTIME) GE P.TNOWEAK)
                THEN CLEAR ITF.INCREASE;
                    ITF.INCTIME = 0;
        ELSE CLEAR ITF.INCREASE; <RA not positive>
            ITF.INCTIME = 0;
END Increase_check;
```

```
PROCESS Increase_proj_check;

    IF (ITF.IFIRM GE P.MINFIRM AND
        ITF.TRTRU LE P.AVEVTAU(ITF.LEV))
        THEN IF (OWNTENT(7) EQ $FALSE)
                THEN ZDOW = MAX(+P.CLMRT, G.ZDOWN);
                ELSE ZDOW = MIN(-P.DESRT, G.ZDOWN);
            RELZ = G.ZOWN - ITF.ZINT;
            RELZD = ZDOW - ITF.ZDINT;
            CALL VERTICAL_MISS_DISTANCE_CALCULATION
              IN (RELZ, RELZD, ITF.TRTRU,
                ITF.TAUR, P.TVPETBL(ITF.LEV))
              OUT (SEP);
            CALL CHECK_PROJECTION
              IN (SEP, P.AVEVALT, OWNTENT)
              OUT (INTHR);

END Increase_proj_check;
```

```
PROCESS Send_initial_intent;

    <form TCAS Resolution Message>

    Place G.MACFLG in MTB field;
    <Form complement from OWNTENT(7)>
    IF (OWNTENT(7) EQ $TRUE)
        THEN COMPLEMENT = 1;
        ELSE COMPLEMENT = 2;
    IF (WL.STATUS EQ $TERM)
        THEN place COMPLEMENT in CVC field;
            CLEAR VRC field;
        ELSE place COMPLEMENT in VRC field;
            CLEAR CVC field;
            IF (ITF.TIEBREAKER_REVERSAL EQ $TRUE)
                THEN IF (OWNTENT(7) EQ $TRUE)
                        THEN COMPLEMENT = 2;
                        ELSE COMPLEMENT = 1;
                    Place COMPLEMENT in CVC field;
    VSB subfield of message = P.PTABLE(BITS 43-46(CVC,VRC)of message);
    Send initial TCAS Resolution (Intent) message to threat
        identified by ITF.IDINT;

END Send_initial_intent;
```

```
PROCESS Complete_send_intent;

    Wait until surveillance delivers reply or decides no reply
        for initial Resolution (Intent) message;
    IF (coordination reply message not received)
          <a coordination reply message is identified by DF=16 and VDS1=3 and VDS2
        THEN TRY = 1;
            CLEAR SUCCESS;
            REPEAT UNTIL ((SUCCESS EQ $TRUE) OR (TRY GT P.TRYMAX));
                    Send TCAS Resolution (Intent) message to threat identified by
                        ITF.IDINT;
                    Wait until Surveillance delivers reply or decides no reply;
                    IF (Coordination reply message is received)
                          <a coordination reply message is identified
                             by DF=16 and VDS1=3 and VDS2=0>
                        THEN SET SUCCESS;
                        ELSE TRY=TRY + 1;
            ENDREPEAT;

END Complete_send_intent;
```

```
ROUTINE CHECK_PROJECTION
  IN (POSITN, THRHLD, OWNTENT)
  OUT (INTHR);

    CLEAR INTHR;
    IF (OWNTENT(7) EQ $FALSE)
        THEN IF (POSITN LT THRHLD)
                THEN SET INTHR;

        ELSE IF (POSITN GT -THRHLD)
                THEN SET INTHR;
<only set flag if position has desired sign>

END CHECK_PROJECTION;
```

```
ROUTINE COORDINATION_UNLOCK

    CLEAR G.COLOCK;
    IF (any messages in Resolution message queue)
        THEN CALL RESOLUTION_MESSAGE_PROCESSING;

END COORDINATION_UNLOCK;
```

```
ROUTINE DELETE_RESOLUTION_ADVISORY
  IN (RAIND);

    IF (RAIND NE 0)
        THEN CLEAR SUCCESS;
            REPEAT WHILE (more entries in TF AND SUCCESS EQ $FALSE);
                IF (RAIND EQ TF.POOWRAR)
                    THEN SET SUCCESS;
                    ELSE select next TF entry;
            ENDREPEAT;

            IF (SUCCESS EQ $TRUE)
                THEN; <cannot delete res.  adv.  which applies to another threat>
                ELSE CLEAR G.RA(RAIND);

END DELETE_RESOLUTION_ADVISORY;
```

```
ROUTINE MODEL_MANEUVERS
  IN (ITF entry)
  OUT (ZMPCLM, ZMPDES);

    CLEAR SKIP_CALC;
    CLEAR All bits in VRAC, VRAD;
    TRTLIM = MIN(ITF.TRTRU, P.TVPETBL(ITF.LEV));
    TAULIM = MIN(ITF.TAUR, P.TVPETBL(ITF.LEV));
    TAULIM = MAX(P.MINTAUM, TAULIM);
    PERFORM Set_up_maneuvers;
    IF (SKIP_CALC EQ $FALSE)
        THEN PERFORM Modeling_calculations;

END MODEL_MANEUVERS;
```

```
PROCESS Set_up_maneuvers;
    DELAYC, DELAYD, TV = P.TV1;
    IF (ITF.REVERSE EQ $TRUE)
        THEN DELAYC, DELAYD, TV = P.QUIKREAC;
    IF (TRTLIM LE TV)
        THEN CALL PROJECT_VERTICAL_GIVEN_BITS
              IN (TRTLIM, G.ZOWN, G.ZDOWN, 0, 0, ITF.REVERSE)
              OUT (ZPOWN);
            CALL PROJECT_VERTICAL_GIVEN_BITS
              IN (TRTLIM, ITF.ZINT, ITF.ZDINT, 0, 0, ITF.REVERSE)
              OUT (ZPINT);
            ZMPCLM = ZPOWN - ZPINT;
            ZMPDES = -ZMPCLM;
            SET SKIP_CALC;
        ELSE SET VRAC(1), VRAD(6); <climb, descend>
            IF (ITF.REVERSE EQ $FALSE AND ITF.OWN_CROSS EQ $FALSE)
                THEN IF (G.RA(6,7,8,9, or 10) EQ $TRUE) <any descend sense>
                        THEN CLEAR VRAC(1);
                            SET VRAC(2); <don't descend>
                            DELAYC, DELAYD = P.QUIKREAC;
                    IF (G.RA(1,2,3,4, or 5) EQ $TRUE) <any climb sense>
                        THEN CLEAR VRAD(6);
                            SET VRAD(7); <don't climb>
                            DELAYC, DELAYD = P.QUIKREAC;

END Set_up_maneuvers;
```

```
PROCESS Modeling_calculations;

    IF (VRAC(1) EQ $TRUE AND G.CLIMBINHIB EQ $TRUE AND ITF.REVERSE EQ
        THEN VRAC(1,2) = '01'; <change climb to don't descend>

    CALL PROJECT_OVER_INTERVAL
      IN (G.ZOWN, G.ZDOWN, VRAC, DELAYC, TRTLIM, TAULIM, ITF.REVERSE)
      OUT (ZCLM1, ZCLM2);
    CALL PROJECT_OVER_INTERVAL
      IN (G.ZOWN, G.ZDOWN, VRAD, DELAYD, TRTLIM, TAULIM, ITF.REVERSE)
      OUT (ZDES1, ZDES2);
    IF (ITF.REVERSE EQ $FALSE)
        THEN CALL PROJECT_OVER_INTERVAL
              IN (ITF.ZINT, ITF.ZDINT, 0, 0, TRTLIM, TAULIM, ITF.REVERSE)
              OUT (ZPIN1, ZPIN2);
        ELSE IF ((ITF.INT_CROSS EQ $TRUE) OR (ITF.ZDINT EQ 0 AND ITF.RZ
                 (ITF.ZDINT * G.ZDMODEL LT 0))
                THEN CALL PROJECT_OVER_INTERVAL
                      IN (ITF.ZINT, ITF.ZDOUTR, 0, 0, TRTLIM, TAULIM,
                         ITF.REVERSE)
                      OUT (ZPIN1, ZPIN2);
                ELSE CALL PROJECT_OVER_INTERVAL
                      IN (ITF.ZINT, ITF.ZDINR, 0, 0, TRTLIM, TAULIM,
                         ITF.REVERSE)
                      OUT (ZPIN1, ZPIN2);

    CALL SEPARATION_OVER_INTERVAL
      IN (ZCLM1, ZCLM2, ZPIN1, ZPIN2, ZPIN1, ZPIN2, ZDES1, ZDES2)
      OUT (ZMPCLM, ZMPDES);

END Modeling_calculations;
```

```
ROUTINE PROJECT_OVER_INTERVAL
  IN (Z, ZD, VRA, DELAY, TRTLIM, TAULIM, RVSFLAG)
  OUT (ZPR1, ZPR2);

    CALL PROJECT_VERTICAL_GIVEN_BITS
      IN (TRTLIM, Z, ZD, VRA, DELAY, RVSFLAG)
      OUT (ZPR1);

    CALL PROJECT_VERTICAL_GIVEN_BITS
      IN (TAULIM, Z, ZD, VRA, DELAY, RVSFLAG)
      OUT (ZPR2);

END PROJECT_OVER_INTERVAL;
```

```
ROUTINE PROJECT_VERTICAL_GIVEN_BITS
  IN (TPROJ, Z, ZD, VRA, DELAY, RVSFLAG)
  OUT (ZPROJ);

    ZDGOAL, TACC = 0;
    TAIR = TPROJ;
    ACCEL = P.VACCEL;
    IF (RVSFLAG EQ $TRUE)
        THEN ACCEL = P.RACCEL;
    IF (VRA NE 0)
        THEN IF (VRA(1,2,3,4, or 5) EQ $TRUE)
                THEN ZDMIN = P.ZDMINTB(MMINDEX(VRA));
                    IF (ZD LT ZDMIN)
                        THEN DIRECTION = +1;
                            ZDGOAL = ZDMIN;
            ELSEIF (VRA(6,7,8,9, or 10) EQ $TRUE)
                THEN ZDMAX = P.ZDMAXTB(MMINDEX(VRA) - 5);
                    IF (ZD GT ZDMAX)
                        THEN DIRECTION = -1;
                            ZDGOAL = ZDMAX;
            OTHERWISE; <can't happen without horizontal RA's>
    IF (ZDGOAL NE 0)
        THEN PERFORM Calculate_acceleration;
    ZPROJ = Z + ZD*TAIR;
    IF (TACC GT 0)
        THEN ZPROJ = ZPROJ + ZDACC*TACC;
            IF (TAGR GT 0)
                THEN ZPROJ = ZPROJ + ZDGOAL*TAGR;
END PROJECT_VERTICAL_GIVEN_BITS;
```

```
PROCESS Calculate_acceleration;

    TUGR = ABS(ZDGOAL-ZD)/ACCEL;
    TESC = TPROJ - DELAY;
    TAIR = DELAY;
    IF (TUGR GT TESC)
        THEN TACC = TESC;
            ZDACC = ZD + 0.5*DIRECTION*ACCEL*TACC;
            TAGR = 0;
        ELSE TACC = TUGR;
            ZDACC = (ZD+ZDGOAL)/2;
            TAGR = TESC - TUGR;

END Calculate_acceleration;
```

```
ROUTINE PROJECT_VERTICAL_GIVEN_ZDGOAL
  IN (TPROJ, Z, ZD, ZDGOAL, DELAY, DIRECTION)
  OUT (VPROJ);

    TDGR = ABS(ZDGOAL - ZD)/P.VACCEL;
    TESC = TPROJ - DELAY;

    IF (TDGR GT TESC)
        THEN TACC = TESC;
            ZDACC = ZD + 0.5*DIRECTION*P.VACCEL*TACC;
            TAGR = 0.;
        ELSE TACC = TDGR;
            ZDACC = (ZD + ZDGOAL)/2;
            TAGR = TESC - TDGR;
    VPROJ = Z + ZD*DELAY;
    IF (TACC GT 0)
        THEN VPROJ = VPROJ + ZDACC*TACC;
            IF (TAGR GT 0)
                THEN VPROJ = VPROJ + ZDGOAL*TAGR;

END PROJECT_VERTICAL_GIVEN_ZDGOAL;
```

```
ROUTINE RESOLUTION_UPDATE
  IN (IND1, IND2);

    CALL DELETE_RESOLUTION_ADVISORY
      IN (IND1);
    IF (IND2 NE 0)
        THEN SET G.RA(IND2);

END RESOLUTION_UPDATE;
```

```
ROUTINE SEPARATION_OVER_INTERVAL
  IN (A1, A2, A3, A4, A5, A6, A7, A8)
  OUT (ZMPCLM, ZMPDES);

    A3 = MAX(A3, G.ZGROUND);
    A4 = MAX(A4, G.ZGROUND);
    A5 = MAX(A5, G.ZGROUND);
    A6 = MAX(A6, G.ZGROUND);

    A7 = MAX(A7, G.ZGROUND + P.ZDESBOT);
    A8 = MAX(A8, G.ZGROUND + P.ZDESBOT);

    ZMPCLM = MIN(A1-A3, A2-A4);
    ZMPDES = MIN(A5-A7, A6-A8);

END SEPARATION_OVER_INTERVAL;
```

```
ROUTINE VSL_OVER_INTERVAL
  IN (VARG, ITF entry)
  OUT (VSLOK);

    ALIMOD = 0;
    ZDI = ITF.ZDINT;

    IF (ITF.WPTR->WL.STATUS EQ $NEW AND VARG GT 0)
        THEN ALIMOD = -P.NEWVSL;
    ELSEIF (ITF.WPTR->WL.STATUS = $NEW)
        THEN;
    OTHERWISE OLDGOAL = P.DESGOAL(EVAL(ITF.TPTR -> TF.PERMTENT));
            IF (OLDGOAL LT VARG)
                THEN ALIMOD = -P.NEWVSL;
    IF (ITF.WPTR->WL.STATUS EQ $NEW AND ITF.BADFOK NE 0)
        THEN IF (OWNTENT(7) = $FALSE)
                THEN ZDI = MAX(ITF.ZDINR, ITF.ZDOUTR);
                ELSE ZDI = MIN(ITF.ZDINR, ITF.ZDOUTR);
    CALL VSL_TEST
      IN (VARG, ITF.TRTRU, ALIMOD, ZDI)
      OUT (VSLOK)
      INOUT (ITF entry);

    IF (VSLOK EQ $TRUE)
        THEN CALL VSL_TEST
              IN (VARG, ITF.TAUR, ALIMOD, ZDI)
              OUT (VSLOK)
              INOUT (ITF entry);

<Return VSLOK = $TRUE only if both tests passed>

END VSL_OVER_INTERVAL;
```

```
ROUTINE VSL_TEST
  IN (AVSL, TAUARG, HVEQ, ZDI)
  OUT (VSLOK)
  INOUT (ITF entry);

    TRTLIM = MIN(TAUARG, P.TVPETBL(ITF.LEV));
    IF (OWNTENT(7) EQ $FALSE)
        THEN DIRECTION = +1;
            IF (G.ZDOWN GE -AVSL)
              THEN VVMD = ITF.RZ + (-AVSL-ZDI)*TRTLIM;
              ELSE IF (ITF.WPTR->WL.STATUS EQ $CONT)
                    THEN T1 = MAX(P.BACKDELAY, P.TV1-(G.TCUR-ITF.TCMD));
                    ELSE T1 = P.TV1;
                  CALL PROJECT_VERTICAL_GIVEN_ZDGOAL
                    IN (TRTLIM, G.ZOWN, G.ZDOWN, -AVSL, T1, DIRECTION)
                    OUT (VO);
                  VVMD = (VO - ITF.ZINT) - ZDI*TRTLIM;
        ELSE DIRECTION = -1;
            IF (G.ZDOWN LE AVSL)
              THEN VVMD = ITF.RZ + (AVSL-ZDI)*TRTLIM;
              ELSE IF (ITF.WPTR->WL.STATUS EQ $CONT)
                    THEN T1 = MAX(P.BACKDELAY, P.TV1 - (G.TCUR-ITF.TCMD));
                    ELSE T1 = P.TV1;
                  CALL PROJECT_VERTICAL_GIVEN_ZDGOAL
                    IN (TRTLIM, G.ZOWN, G.ZDOWN, AVSL, T1, DIRECTION)
                    OUT (VO);
                  VVMD = (VO - ITF.ZINT) - ZDI*TRTLIM;
    CALL CHECK_PROJECTION
      IN (VVMD, G.ALIM-HVEQ, OWNTENT)
      OUT (INTHR);
    IF (INTHR EQ $TRUE)
        THEN CLEAR VSLOK;
        ELSE SET VSLOK;
END VSL_TEST;
```

```
FUNCTION EVAL
  IN (ARG) <bit string>
  OUT (EVAL); <integer>

    IF (ARG(4) EQ $FALSE)
        THEN EVAL = 0;
    ELSEIF (ARG(6) EQ $FALSE) <positive or negative>
        THEN IF (ARG(5) EQ $FALSE)
                THEN EVAL = 8;
                ELSE EVAL = 4;
    OTHERWISE EVAL = 1; <VSL>
            IF (ARG(11) EQ 0) <VSL 500>
                THEN EVAL = EVAL + 2;
            IF (ARG(12) EQ 0) <VSL 1000>
                THEN EVAL = EVAL + 1;

END EVAL;
```

```
FUNCTION MMINDEX
  IN (ARG) <bit string>
  OUT (MMINDEX); <integer>

    MMINDEX = position of first "1" in input bit string;
    <Will be from 1-10>

END MMINDEX;
```

```
FUNCTION RAMAP
  IN (ARG) <bit string>
  OUT (RAMAP); <integer>

    RAMAP = index value associated with input bit string, as
    defined in Table 6-3.

END RAMAP;
```

```
TASK HOUSEKEEPING;

    REPEAT WHILE (G.COLOCK EQ $TRUE);
        <Loop while waiting for coordination lock state to end.  Performance
         Monitor should recognize when TCAS has been locked for more than
         P.TUNLOCK seconds and take appropriate action.>
    ENDREPEAT;

    SET G.COLOCK using uninterruptible test and set instruction;
    G.TLOCK = TCLOCK;
    Null the Delete_RA List and Delete_Intent List;
    PERFORM Threat_file_housekeeping;
    PERFORM Resolution_advisory_housekeeping;
    PERFORM Sensitivity_level_housekeeping;
    Set up fields in Coordination Update message:
        ARA = G.RA, RAC = G.INTENT;
    Send Coordination Update message to transponder;
    CALL COORDINATION_UNLOCK

END HOUSEKEEPING;
```

```
PROCESS Threat_file_housekeeping;
    REPEAT WHILE (more entries in TF);
        IF (TF.DEFER_DISPLAY EQ $FALSE AND TF.POOWRAR EQ 0
              AND TF.POTHRAR(1) EQ 0 AND TF.POTHRAR(2) EQ 0)
            THEN IF (TF.IPTR NE $NULL)
                    THEN TF.IPTR->ITF.TPTR = $NULL;
                Clear TF entry variables and delete entry;
        Select next entry;
    ENDREPEAT;
    REPEAT WHILE (more entries in TF);
        IF (TF.TLRCMD GE 0)
            THEN IF (G.TCUR - TF.TLRCMD GT P.TCATRES)
                    THEN put TF.POOWRAR on Delete_RA List;
                        IF ((TF.POTHRAR(1) EQ 0) AND (TF.POTHRAR(2) EQ 0))
                            THEN IF (TF.IPTR NE $NULL)
                                    THEN TF.IPTR->ITF.TPTR = $NULL;
                                Clear TF entry variables and delete entry;
                            ELSE TF.POOWRAR = 0;
                                TF.DEFER_DISPLAY = $FALSE;
                                TF.IPTR = $NULL;
                                CLEAR All bits in TF.PERMTENT;
        Select next entry;
    ENDREPEAT;
    REPEAT WHILE (more entries in TF);
US         IF (TF.TTHLRCM GE 0)
            THEN IF (TCLOCK - TF.TTHLRCM GT P.TCATRES)
                    THEN put TF.POTHRAR(1 & 2) on Delete_Intent List;
                        IF (TF.POOWRAR EQ 0 AND TF.DEFER_DISPLAY EQ $FALSE)
                            THEN IF (TF.IPTR NE $NULL)
                                    THEN TF.IPTR->ITF.TPTR = $NULL;
                                Clear TF entry variables and delete entry;
                            ELSE TF.POTHRAR(1), TF.POTHRAR(2) = 0;
        Select next entry;
    ENDREPEAT;
    PERFORM Threat_file_ITF_linkup;
END Threat_file_housekeeping;
```

```
PROCESS Threat_file_ITF_linkup;

    REPEAT WHILE (more entries in ITF);
        IF (ITF.EQP EQ $TCAS AND ITF.TPTR EQ $NULL)
            THEN CLEAR SUCCESS;
                REPEAT WHILE (more entries in TF AND SUCCESS EQ $FALSE);
                    IF (ITF.IDINT EQ TF.ID)
                        THEN SET SUCCESS;
                        ELSE select next TF entry;
                ENDREPEAT;
                IF (SUCCESS EQ $TRUE)
                    THEN TF.IPTR = ITF.IROW;
                        ITF.TPTR = address of TF entry;
        Select next ITF entry;
    ENDREPEAT;

END Threat_file_ITF_linkup;
```

```
PROCESS Resolution_advisory_housekeeping;

<Process deletion list of Resolution Advisories and Complements
received from Threat_file_housekeeping>

    REPEAT WHILE (more entries on Delete_RA List);
        CALL DELETE_RESOLUTION_ADVISORY <delete advisory if not needed for other
          IN (index to Res.  Advisory);
        Select next list entry;
    ENDREPEAT;

    REPEAT WHILE (more entries on Delete_Intent List);
        CALL DELETE_INTENT <Chapter 3>
          IN (index to RA complement);
        Select next list entry;
    ENDREPEAT;

END Resolution_advisory_housekeeping;
```

```
PROCESS Sensitivity_level_housekeeping;

    SIT = 1;
    REPEAT UNTIL (SIT GT P.SITMAX);
        IF (TCLOCK - G.LEVELTIM(SIT) GT P.STIMOUT)
            THEN G.LEVELTIM(SIT), G.LEVELSIT(SIT) = 0;
        SIT = SIT + 1;
    ENDREPEAT;

END Sensitivity_level_housekeeping;
```

```
STRUCTURE TRAFVAR

  GROUP settable
    FLT DMODTA <Traffic Advisory version of DMOD>
    FLT H1TA <Traffic Advisory version of H1>
    FLT RTHRTA <Traffic Advisory version of range threshold>
    FLT TRTHRTA <Traffic Advisory version of TRTHR>
    FLT TVTHRTA <Traffic Advisory version of TVTHR>

  GROUP flags
    BIT DISPROX <TRUE => prox advisories to be displayed, this cycle>
    BIT PROX_TEST <TRUE => prox test req'd (failed TA & RA tests)>
    BIT PRXHITA <Proximity criteria satisfied for traffic adv>
    BIT RHITA <Range criteria satisfied for traffic adv>
    BIT ZHITA <Altitude criteria satisfied for traffic adv>

  GROUP temp_calculation
    FLT RDTA
    FLT SCORFACTR
    FLT TAURTA
    FLT TAUVTA

ENDSTRUCTURE;
```

```
TASK TRAFFIC_ADVISORY;
```

↑ *Supervisory-Interface* ▷ *Pilot-Displays* ▷ *Traffic-Advisories[i]* ▷ *Status*

```
    Set TDV to null;

    IF (G.TAMODE EQ $TRUE)
        THEN REPEAT WHILE (more entries in ITF);
                PERFORM Traffic_advisory_detection;
                Select next entry;
            ENDREPEAT;

            PERFORM Traffic_display;

END TRAFFIC_ADVISORY;
```

```
PROCESS Traffic_advisory_detection;
    CLEAR PROX_TEST;
    IF (ITF.TACODE EQ $RA)
        THEN ITF.TATIME = P.MINTATIME; <no change, set min TA disp.  time >
    ELSEIF (ITF.IOGROUN EQ $TRUE)
        THEN ITF.TACODE = $NOTAPA;
            ITF.TATIME = 0;
    OTHERWISE PERFORM Traffic_parameters;
            PERFORM Traffic_range_test;
            IF (RHITA EQ $TRUE)
                THEN PERFORM Range_hit_processing;
                ELSE IF (ITF.TATIME NE 0)
                        THEN ITF.TATIME = ITF.TATIME - 1;
                            IF (ITF.MODC EQ $FALSE)
                                THEN ITF.TACODE = $TANMC;
                                ELSE ITF.TACODE = $TAMC;
                        ELSE SET PROX_TEST;
            IF (PROX_TEST EQ $TRUE)
                THEN PERFORM Proximity_test;
                    IF (PRXHITA EQ $TRUE)
                        THEN ITF.TACODE = $PENDPA;
                        ELSE ITF.TACODE = $NOTAPA;
END Traffic_advisory_detection;
```

```
PROCESS Traffic_parameters;

    ITF.LEV = MAX(G.INDEX,ITF.PLINT);
    TRTHRTA = P.TRTHRTA_TBL(ITF.LEV);
    TVTHRTA = P.TVTHRTA_TBL(ITF.LEV);
    RTHRTA = P.RTHRTA_TBL(ITF.LEV);
    DMODTA = P.DMODTA_TBL(ITF.LEV);
    H1TA = P.H1TA_TBL(ITF.LEV);

    IF (ITF.MODC EQ $FALSE)
        THEN IF (ITF.LEV EQ 2)
                THEN TRTHRTA = P.TRTUTBL(3);
                ELSE TRTHRTA = P.TRTUTBL(ITF.LEV);
    ELSEIF (G.RAMODE EQ $TRUE)
        THEN; <use relative altitude values in track file>
    OTHERWISE ITF.RZ = G.ZOWN - ITF.ZINT;
            ITF.RZD = G.ZDOWN - ITF.ZDINT;
            ITF.A = ABS(ITF.RZ);
            ITF.ADOT = ITF.RZD * SIGN(ITF.RZ);

END Traffic_parameters;
```

```
PROCESS Traffic_range_test;

    RDTA = ITF.RD;
    IF (ITF.R LT RTHRTA)
        THEN SET RHITA;
    ELSEIF (ITF.RD GT P.RDTHRTA)
        THEN IF (ITF.R * ITF.RD GT H1TA OR ITF.R GT DMODTA)
                THEN CLEAR RHITA;
                ELSE SET RHITA;
    OTHERWISE IF (ITF.RD GE -P.RDTHRTA)
                THEN RDTA = -P.RDTHRTA;
            IF (ITF.R GT 0)
                THEN TAURTA = -(ITF.R-((DMODTA**2)/ITF.R))/RDTA;
                ELSE TAURTA = P.MINTAU;
            IF (TAURTA LT TRTHRTA)
                THEN SET RHITA;
                ELSE CLEAR RHITA;

END Traffic_range_test;
```

```
PROCESS Range_hit_processing;

    IF (ITF.MODC EQ $FALSE)
        THEN IF ((ITF.TACODE NE $TANMC) AND
                (ITF.BEAROK EQ $FALSE OR ITF.RFLG EQ $FALSE)
                THEN ITF.TACODE = $NOTAPA;
                ELSE IF (G.ZOWN GE P.ABOVNMC)
                        THEN IF (ITF.TATIME EQ 0)
                                THEN ITF.TACODE = $NOTAPA;
                                ELSE ITF.TACODE = $TANMC;
                                    ITF.TATIME = ITF.TATIME - 1;
                        ELSE ITF.TACODE = $TANMC;
                            ITF.TATIME = P.MINTATIME;
        ELSE PERFORM Traffic_altitude_test;
            IF (ZHITA EQ $TRUE)
                THEN ITF.TACODE = $TAMC;
                    ITF.TATIME = P.MINTATIME;
                ELSE IF (ITF.TATIME NE 0)
                        THEN ITF.TACODE = $TAMC;
                            ITF.TATIME = ITF.TATIME - 1;
                        ELSE SET PROX_TEST;

END Range_hit_processing;
```

```
PROCESS Traffic_altitude_test;

    IF (ITF.A LT P.ZTHRTA)
        THEN SET ZHITA;
    ELSEIF (ITF.ADOT GE P.ZDTHRTA)
        THEN CLEAR ZHITA;
    OTHERWISE TAUVTA = (-ITF.A/ITF.ADOT);
            IF (TAUVTA LT TVTHRTA)
                THEN SET ZHITA;
                ELSE CLEAR ZHITA;

END Traffic_altitude_test;
```

```
PROCESS Proximity_test;

    IF (ITF.R GE P.PROXR)
        THEN CLEAR PRXHITA;
    ELSEIF (ITF.MODC EQ $FALSE)
        THEN IF (G.ZOWN GE P.ABOVNMC)
                THEN CLEAR PRXHITA;
                ELSE IF ((ITF.TACODE NE $TANMC AND
                        ITF.TACODE NE $PA) AND
                        (ITF.BEAROK EQ $FALSE OR
                        ITF.RFLG EQ $FALSE))
                        THEN CLEAR PRXHITA;
                        ELSE SET PRXHITA;
    OTHERWISE IF (ITF.A LT P.PROXA)
                THEN SET PRXHITA;
                ELSE CLEAR PRXHITA;

END Proximity_test;
```

```
PROCESS Traffic_display;

    CLEAR DISPROX;
    IF (O.ALLPROX EQ $TRUE)
        THEN G.ALLPROXTIME = P.ALLPROXDUR;
    IF (G.ALLPROXTIME GE 0)
        THEN SET DISPROX;
            G.ALLPROXTIME = G.ALLPROXTIME - 1;
    REPEAT WHILE (more entries in ITF AND DISPROX EQ $FALSE);
        IF (ITF.TACODE GE P.NEEDPROX) <currently $RA, $TAMC, or $TANMC>
            THEN SET DISPROX;
            ELSE select next ITF entry;
    ENDREPEAT;

    REPEAT WHILE (more entries in ITF);
        IF (ITF.TACODE EQ $PENDPA AND DISPROX EQ $TRUE)
            THEN ITF.TACODE = $PA;
        PERFORM Traffic_score;
        Select next ITF entry;
    ENDREPEAT;
    Sort pointers on ITF.TASCORE;
    Move sorted ITF pointers with TASCORE>0 to TDV;

END Traffic_display;
```

```
PROCESS Traffic_score;

    DMODTA = P.DMODTA_TBL(ITF.LEV);

    IF (ITF.MODC EQ $TRUE)
        THEN SCORFACTR = 2;
        ELSE SCORFACTR = 1;

    IF (ITF.TACODE LT P.DISPTHR)
        THEN ITF.TASCORE = 0;
    ELSEIF (ITF.TACODE EQ $RA)
        THEN ITF.TASCORE = P.HISCORE;
    ELSEIF (ITF.TACODE EQ $TAMC or $TANMC)
        THEN IF (ITF.R LE DMODTA)
                THEN ITF.TASCORE = (P.MEDHISCORE * SCORFACTR) - ITF.R;
            ELSEIF (ITF.RD LT -P.RDTHRTA)
                THEN ITF.TASCORE = (P.MEDSCORE * SCORFACTR) + (ITF.R/ITF.RD);
            OTHERWISE ITF.TASCORE = (P.MEDLOSCORE * SCORFACTR) - ITF.R;
    ELSEIF (ITF.TACODE EQ $PA)
        THEN ITF.TASCORE = (P.LOSCORE * SCORFACTR) - ITF.R;
    OTHERWISE ITF.TASCORE = 0;

END Traffic_score;
```

```
STRUCTURE DISPVAR

  GROUP cor_prev
    FLT GOALCL
    FLT GOALDES
    INT RAIND

  GROUP crossing_RA
    FLT PROJ_ZDINT

ENDSTRUCTURE;
```

```
TASK DISPLAY_ADVISORIES;
<FUNCTION EVAL found in Chapter 6>

    REPEAT WHILE (G.COLOCK EQ $TRUE);
        <Loop while waiting for coordination lock state to end.  Performance
         Monitor should recognize when TCAS has been locked for more than
         P.TUNLOCK seconds and take appropriate action.>
    ENDREPEAT;
    SET G.COLOCK using uninterruptible test and set instruction;
    G.TLOCK = TCLOCK;

    G.CLSTROLD = G.CLSTRONG;
    G.DESTROLD = G.DESTRONG;
    CLEAR G.ANYNEWTHR, G.ANYPRECOR, G.ANYINCREASE;
    G.DESTRONG, G.CLSTRONG = 0;
    REPEAT WHILE (more TF entries);
        IF (TF.POOWRAR NE 0)
            THEN IF (TF.PERMTENT(7) EQ $FALSE)
                    THEN G.CLSTRONG = MAX(G.CLSTRONG, EVAL(TF.PERMTENT));
                    ELSE G.DESTRONG = MAX(G.DESTRONG, EVAL(TF.PERMTENT));
                IF (TF.NEW EQ $TRUE)
                    THEN SET G.ANYNEWTHR;
                        CLEAR TF.NEW;
        Select next TF entry;
    ENDREPEAT;

    PERFORM Set_up_goal_rate;
    PERFORM Corrective_preventive_test;
    PERFORM Set_up_global_flags;
    CALL COORDINATION_UNLOCK;

END DISPLAY_ADVISORIES;
```

```
PROCESS Set_up_goal_rate;

    GOALCL = -P.HUGE, GOALDES = +P.HUGE;
    REPEAT WHILE (more TF entries);
        RAIND = EVAL(TF.PERMTENT);
        IF (RAIND NE 0)
            THEN PERFORM Determine_goal_rate;
        Select next TF entry;
    ENDREPEAT;
END Set_up_goal_rate;
```

```
PROCESS Determine_goal_rate;
    IF (TF.PERMTENT(7) EQ $FALSE)
        THEN GOALCL = MAX(GOALCL, P.CLIMBGOAL(RAIND));
            IF (RAIND EQ 8) <positive RA>
                THEN IF (G.CLSTROLD LT 8 OR G.ANYNEWTHR EQ $TRUE)
                        THEN G.ZDMODEL = MAX(GOALCL, G.ZDOWN, G.ZDMODEL);
                    IF (TF.IPTR -> ITF.INCREASE EQ $TRUE AND
                          (G.ZDOWN LE P.INC_CLMRATE OR
                           G.CORINC EQ $TRUE))
                        THEN SET G.ANYINCREASE;
                            IF (G.CORINC EQ $FALSE)
                                THEN SET G.INC_ENCOUNTER,G.CORINC,
                                    G.CORRECTIVE_CLM, G.ANYPRECOR;
                                    G.ZDMODEL = P.INC_CLMRATE;
                        ELSE IF (TF.IPTR -> ITF.INCREASE EQ $FALSE
                                  AND G.CORINC EQ $TRUE)
                                THEN G.ZDMODEL = MAX(GOALCL, G.ZDOWN);
                    GOALCL = MAX(GOALCL, G.ZDMODEL);
        ELSE GOALDES = MIN(GOALDES, P.DESGOAL(RAIND));
            IF (RAIND EQ 8) <positive RA>
                THEN IF (G.DESTROLD LT 8 OR G.ANYNEWTHR EQ $TRUE)
                        THEN G.ZDMODEL = MIN(GOALDES, G.ZDOWN,G.ZDMODEL);
                    IF (TF.IPTR -> ITF.INCREASE EQ $TRUE AND
                          (G.ZDOWN GE P.INC_DESRATE OR
                           G.CORINC EQ $TRUE))
                        THEN SET G.ANYINCREASE;
                            IF (G.CORINC EQ $FALSE)
                                THEN SET G.INC_ENCOUNTER,G.CORINC,
                                    G.CORRECTIVE_DES, G.ANYPRECOR;
                                    G.ZDMODEL = P.INC_DESRATE;
                        ELSE IF (TF.IPTR -> ITF.INCREASE EQ $FALSE
                                  AND G.CORINC EQ $TRUE)
                                THEN G.ZDMODEL = MIN(GOALDES, G.ZDOWN);
                    GOALDES = MIN(GOALDES, G.ZDMODEL);
END Determine_goal_rate;
```

```
PROCESS Corrective_preventive_test;
    IF (GOALCL EQ 0 AND GOALDES EQ 0)
        THEN IF (G.CORRECTIVE_CLM EQ $TRUE AND G.ZDOWN LT P.SMALLRATE AND
                  (G.CLSTRONG LT G.CLSTROLD OR (G.CORINC EQ $TRUE AND
                  G.ANYINCREASE EQ $FALSE)))
                THEN CLEAR G.CORRECTIVE_CLM, G.CORINC;
                ELSE IF (G.CORRECTIVE_DES EQ $TRUE AND G.ZDOWN GT -P.SMALLRATE A
                          (G.DESTRONG LT G.DESTROLD OR (G.CORINC EQ $TRUE
                          AND G.ANYINCREASE EQ $FALSE)))
                        THEN CLEAR G.CORRECTIVE_DES, G.CORINC;
            IF (G.CORRECTIVE_CLM EQ $FALSE AND G.ZDOWN LT -(P.SMALLRATE+P.HYST
                THEN SET G.CORRECTIVE_CLM, G.ANYPRECOR;
                ELSE IF (G.CORRECTIVE_DES EQ $FALSE AND
                          G.ZDOWN GT (P.SMALLRATE + P.HYSTERCOR))
                        THEN SET G.CORRECTIVE_DES, G.ANYPRECOR;
        ELSE IF (G.CORRECTIVE_CLM EQ $TRUE AND G.ZDOWN GE GOALCL AND
                  (G.CLSTRONG LT G.CLSTROLD OR (G.CORINC EQ $TRUE AND
                  G.ANYINCREASE EQ $FALSE)))
                THEN CLEAR G.CORRECTIVE_CLM, G.CORINC;
                ELSE IF (G.CORRECTIVE_DES EQ $TRUE AND G.ZDOWN LE GOALDES AND
                          (G.DESTRONG LT G.DESTROLD OR (G.CORINC EQ $TRUE
                          AND G.ANYINCREASE EQ $FALSE)))
                        THEN CLEAR G.CORRECTIVE_DES, G.CORINC;
            IF (G.CORRECTIVE_CLM EQ $FALSE AND G.ZDOWN LT GOALCL-P.HYSTERCOR)
                THEN SET G.CORRECTIVE_CLM, G.ANYPRECOR;
                ELSE IF (G.CORRECTIVE_DES EQ $FALSE AND
                          G.ZDOWN GT GOALDES+P.HYSTERCOR)
                        THEN SET G.CORRECTIVE_DES, G.ANYPRECOR;
END Corrective_preventive_test;
```

```
PROCESS Set_up_global_flags;
```

↑ *Supervisory-Interface* ▹ *Pilot-Controls* ▹ *Aural-Alarm*
↑ *Supervisory-Interface* ▹ *Pilot-Controls* ▹ *Resolution-Advisory* ▹ *Combined-Control*
↑ *Supervisory-Interface* ▹ *Pilot-Controls* ▹ *Resolution-Advisory* ▹ *Vertical-Control*

```
    CLEAR G.ALARM,G.ANYFIRMDEL,G.ANYCORCHANG,G.ANYCROSS,G.ALLCLEAR;
    CLEAR G.ANYREVERSE, G.ANYTRACKDROP, G.ANYALTLOST, G.MAINTAIN;
    IF (G.CORRECTIVE_CLM EQ $FALSE AND G.CORRECTIVE_DES EQ $FALSE
          AND G.RA(1 or 6) EQ $TRUE)
        THEN SET G.MAINTAIN;
        ELSE IF (G.RA(1 and 6) EQ $FALSE)
                THEN G.ZDMODEL = 0;
    REPEAT WHILE (more ITF entries);
        CLEAR ITF.DCFLG;
        IF (ITF.PREWARN EQ 1)
            THEN ITF.PREWARN = ITF.PREWARN + 1;
                SET G.ANYFIRMDEL;
        IF (G.RA(1-10) EQ $FALSE)
            THEN CLEAR G.INC_ENCOUNTER;
                IF (ITF.ALTITUDE_LOST EQ $TRUE)
                    THEN SET G.ANYALTLOST, CLEAR ITF.ALTITUDE_LOST;
                ELSEIF (ITF.TRACK_DROP EQ $TRUE)
                    THEN SET G.ANYTRACKDROP, CLEAR ITF.TRACK_DROP;
                ELSEIF (ITF.CLEAR_CONFLICT EQ $TRUE)
                    THEN SET G.ALLCLEAR, CLEAR ITF.CLEAR_CONFLICT;
            ELSE PERFORM Crossing_flag_check;
                IF (ITF.REVERSE EQ $TRUE AND G.RA(1 or 6) EQ $TRUE AND
                      ITF.INC_ENC EQ $FALSE)
                    THEN SET G.ANYREVERSE;
        Select next ITF entry;
    ENDREPEAT;
    PERFORM Set_up_display_outputs;
    IF (G.CORRECTIVE_CLM EQ $TRUE OR G.CORRECTIVE_DES EQ $TRUE)
        THEN IF (G.CLSTRONG NE G.CLSTROLD OR G.DESTRONG NE G.DESTROLD)
                THEN SET G.ANYCORCHANG;
    IF (G.ANYNEWTHR EQ $TRUE OR G.ANYPRECOR EQ $TRUE OR G.ANYCORCHANG
        THEN SET G.ALARM;
END Set_up_global_flags;
```

```
PROCESS Crossing_flag_check;
    IF (ITF.TACODE EQ $RA AND ITF.TPTR NE $NULL)
        THEN IF ((ITF.TPTR->TF.PERMTENT(7) EQ $FALSE AND
                  ITF.RZ LE -P.CROSSTHR) OR (ITF.TPTR->TF.PERMTENT(7) EQ $TRUE
                  AND ITF.RZ GE P.CROSSTHR))
                THEN IF (ITF.TPTR->TF.PERMTENT(5,6) EQ '00')
                        THEN SET G.ANYCROSS;
                    IF (ITF.INT_CROSS EQ $FALSE AND ITF.OWN_CROSS EQ $FALSE)
                        THEN PROJ_ZDINT = ITF.ZINT + (ITF.ZDINT * ITF.TRTRU);
                            IF (G.ZDMODEL EQ 0)
                                THEN SET ITF.INT_CROSS;
                            ELSEIF (ABS(ITF.ZDINT) GE P.OLEV AND
                                ((G.ZDMODEL GT 0 AND G.ZOWN GT PROJ_ZDINT) OR
                                (G.ZDMODEL LT 0 AND G.ZOWN LT PROJ_ZDINT)))
                                THEN SET ITF.INT_CROSS;
                            OTHERWISE SET ITF.OWN_CROSS;
                ELSE IF (ITF.INT_CROSS EQ $TRUE OR ITF.OWN_CROSS EQ $TRUE)
                        THEN IF ((ITF.TPTR->TF.PERMTENT(7) EQ $FALSE AND
                                  ITF.RZ GE P.CROSSTHR) OR
                                  (ITF.TPTR->TF.PERMTENT(7) EQ $TRUE
                                  AND ITF.RZ LE -P.CROSSTHR))
                                THEN CLEAR ITF.INT_CROSS;
                                    CLEAR ITF.OWN_CROSS;
                                ELSE; <Don't clear cross flags until own
                                      aircraft has actually crossed through
                                      the altitude of the intruder.>

                <NOTE: Cross flags for a non-crossing encounter are already
                      cleared to begin with.>

        ELSE CLEAR ITF.INT_CROSS;
            CLEAR ITF.OWN_CROSS;
END Crossing_flag_check;
```

```
PROCESS Set_up_display_outputs;
↑ Supervisory-Interface ▹ Pilot-Displays ▹ Resolution-Advisory ▹ Combined-Control
↑ Supervisory-Interface ▹ Pilot-Displays ▹ Resolution-Advisory ▹ Vertical-Control
    IF (G.ANYINCREASE EQ $TRUE)
        THEN CLEAR G.ANYREVERSE, G.MAINTAIN, G.ANYCROSS;
            IF (G.PREVINCREASE EQ $FALSE)
                THEN SET G.ANYCORCHANG, G.PREVINCREASE;
        ELSE CLEAR G.PREVINCREASE;
            IF (G.MAINTAIN EQ $TRUE OR G.ANYREVERSE EQ $TRUE)
                THEN CLEAR G.ANYCROSS;
                    IF (G.MAINTAIN EQ $TRUE AND G.ANYREVERSE EQ $TRUE)
                        THEN CLEAR G.ANYREVERSE;
                            IF (G.CLSTRONG NE G.CLSTROLD OR
                                  G.DESTRONG NE G.DESTROLD)
                                THEN SET G.ANYCORCHANG;
    IF (G.RA(1-10) EQ $FALSE)
        THEN IF (G.ANYALTLOST EQ $TRUE)
                THEN CLEAR G.ANYTRACKDROP, G.ALLCLEAR;
                ELSE IF (G.ANYTRACKDROP EQ $TRUE)
                        THEN CLEAR G.ALLCLEAR;
                        ELSE IF (G.ALLCLEAR EQ $FALSE)
                                THEN indicate "No Advisory" in DITS Word 270;
        ELSE IF (G.CORRECTIVE_CLM EQ $FALSE AND G.CORRECTIVE_DES EQ $FALSE)
                THEN indicate "Preventive RA" in DITS Word 270;
                ELSE indicate "Corrective RA" in DITS Word 270;
                        <Climb or Descend sense>
    Quantize G.ZDMODEL to 100 ft/min for use as signed rate to display;
      <Note:  The actual rate that is shown on the RA display is dependent upon
       quantization and segmentation of the instrument's "eyebrow" lights, and
       could be different than the rate specified in DITS Word Label 270.>
    Formulate DITS VERTICAL RA DATA OUTPUT WORD FOR TCAS (Label 270)
     to be sent to RA display, TA display and aural annunciation subsystem;
      <Includes rate to display for "Maintain Vertical Speed" RA,
       if any, as well as flags for combined control and vertical control
       fields, and bit designations for climb and descend sense RAs.  Note
       that the logic specified above precludes the setting of multiple flags.>
END Set_up_display_outputs;
```

Hardware Assembly and Installation Instructions

This section might include (or include references to) the assembly and installation instructions for the equipment.

Training Requirements (Plan)

A training plan would be included here or appropriate pointers supplied instead. Some of the results of the simulator studies should be useful here and may involve links with the information about simulator studies in level 2. Again, only a few simple examples of the information that might be specified is included here.

> The training must encourage pilots to use all the information available to them to maintain a safe distance from other aircraft, but the inaccuracy of bearing and altitude information on traffic advisory displays must be emphasized.
>
> The likelihood that anticipatory avoidance maneuvers may compromise the performance of the TCAS in computing the most effective avoidance maneuver should also be discussed in training.
>
> Pilots must learn to use the system the way the designers and its logic intend it to be used, though they also must remember to use their training and experience to evaluate situations and take appropriate action to ensure safety of flight.

Appendix A

Constant Definitions

ABOVNMC = 15500 ft.

AVEVALT = 200 ft.

ALFAO = 0.58.

ALFAR = 0.4.

BACKDELAY = −2.5 s.

BETAO = 0.25.

BETAR = 0.15.

CLMRT = 1500 ft/min.

CROSSTHR = 100 ft.

HISCORE = 1200.

HUGE = 10000 ft/min.

HYSTERCOR = 300 ft/min.

ILEV = 1000 ft/min.

INCCLMRATE = 2500 ft/min.

INCDESRATE = -2500 ft/min.

KNOWGROL = 1650 ft.

LOSCORE = 100.

LOWFIRMRZ = 150 ft.

MAXALTDIFF = 600 ft.

MEDHISCORE = 500.

MEDLOSCORE = 300.

MEDSCORE = 400.

MINFIRM = 2.

MINRITIME = 4.0 s.

MINRVSTIME = 10 s.

MINSEP = 300 ft.

MINTATIME = 8 s.

MINTAUM = 10 s.

MINTAU = 0 s.

NAFRANGE = 1.5 nmi.

NEARGROH = 200 ft.

NEARGROL = 180 ft.

NEWVSL = 75 ft.

NODESHI = 1200ft.

NODESLO = 1000ft.

NOZCROSS = 100ft.

OLEV = 600 ft/min.

OWNDEL = 4 s.

PROXA = 1200 ft.

PROXR = 6.0 nmi.

QUIKREAC = 2.5s.

RACCEL = 11.2 ft/s^2.

RADARLOST = 10.

RDTHRTA = 10 ft/s.

RDTHR = 10 ft/s.

RMAX = 12.0 nmi.

SMALLRATE = 500 ft/min.

STIMOUT = 240s.

STROFIR = 20.

TCATRES = 6 s.

TMIN = 5.5 s.

NOWEAK = 10s.

TRVSNOWEAK = 5s.

VACCEL = 8 ft/s^2.

TV1 = 5 s.

V500 = 500 ft/min.

V1000 = 1000 ft/min.

V2000 = 2000 ft/min.

V0 = 0 ft/min.

WTTHR = 2 s.

ZDESBOT = 900 ft.

ZDTHRTA = -1 ft/s.

ZLARGE = 1000000 ft.

ZNOAURALLO = 400 ft.

ZNOAURALHI = 600 ft.

ZNOINCDES = 1450ft.

ZSL5TO4 = 2150 ft.

ZSL5TO6 = 10500 ft.

ZSL4TO5 = 2550 ft.

ZSL4TO2 = 400 ft.

ZSL7TO6 = 19500 ft.

ZSL6TO5 = 9500 ft.

ZSL6TO7 = 20500 ft.

ZSL2TO4 = 600 ft.

ZTHRTA = 1200 ft.

Appendix B

Table Definitions

Increase-Tau-Threshold

Mops Reference: AVEVTAU
Description:
Unit: s
Index: Effective Sensitivity Level
Definition:

3	4	5	6	7
13	18	20	24	26

Potential-Threat-Minimum-Range-Threshold

Mops Reference: DMODTA-TBL
Description:
Unit: nmi
Index: Effective Sensitivity Level
Definition:

2	3	4	5	6	7
.30	.40	.55	.75	1.0	1.3

Potential-Threat-Minimum-Divergence-Threshold

Mops Reference: H1TA-TBL
Description:
Unit: nm^2/s
Index: Effective Sensitivity Level
Definition:

2	3	4	5	6	7
.00160	.002	.00278	.00278	.00278	.004

Potential-Threat-Maximum-Range-Threshold

Mops Reference: RTHRTA-TBL
Description:
Unit: nmi
Index: Effective Sensitivity Level
Definition:

2	3	4	5	6	7
.55	.65	.85	1.10	1.30	1.70

Potential-Threat-Modified-Tau-Threshold

Mops Reference: TRTHRTA-TBL
Description:
Unit: s
Index: Effective Sensitivity Level
Definition:

2	3	4	5	6	7
20	30	35	40	45	48

XTVPCTBLX

Mops Reference: TVPC-TBL
Description:
Unit: s
Index: Intruder Sensitivity Level
Definition:

3	4	5	6	7
35	40	40	45	48

XTVPETBLX

Mops Reference: TVPE-TBL
Description:
Unit: s
Index: Intruder Sensitivity Level
Definition:

3	4	5	6	7
25	30	30	35	40

Potential-Threat-Time-to-Co-Altitude-Threshold

Mops Reference: TVTHRTA-TBL
Description:
Unit: s
Index: Effective Sensitivity Level
Definition:

2	3	4	5	6	7
20	30	35	40	45	48

Threat-Minimum-Range-Threshold

Mops Reference: DMOD-TBL
Description:
Unit: nmi
Index: Effective Sensitivity Level
Definition:

3	4	5	6	7
.20	.35	.55	.80	1.10

Threat-Minimum-Divergence-Threshold

Mops Reference: H1-TBL
Description:
Unit: nm^2/s
Index: Effective Sensitivity Level
Definition:

3	4	5	6	7
.002	.00278	.00278	.00278	.004

Threat-Time-To-CPA-Threshold-E

Mops Reference: TRTE-TBL
Description:
Unit: s
Index: Effective Sensitivity Level
Definition:

3	4	5	6	7
15	20	25	30	35

Threat-Time-To-Co-Alt-Threshold-U

Mops Reference: TVTU-TBL
Description:
Unit: s
Index: Effective Sensitivity Level
Definition:

3	4	5	6	7
15	20	25	30	35

Threat-Modified-Tau-Threshold

Mops Reference: TRTHR-TBL
Description:
Unit: s
Index: Effective Sensitivity Level
Definition:

3	4	5	6	7
15	20	25	30	35

Time-To-CPA-Firmness-Dependent

Mops Reference: FRTHR-TBL
Description:
Unit: s
Index: (Effective Sensitivity Level × Other-Track-Firmness$_{\text{f-431}}$)
Definition:

	3	**4**	**5**	**6**	**7**
0	0	0	0	0	0
1	0	0	0	0	0
2	15	20	25	30	35
3	15	20	25	30	35

Positive-RA-Altitude-Limit-Threshold

Mops Reference: AL
Description:
Unit: Feet
Index: Altitude Layer
Definition:

1	2	3	4
400	500	640	740

Threat-Alt-Threshold

Mops Reference: ZT
Description:
Unit: Feet
Index: Altitude Layer
Definition:

1	2	3	4
750	750	850	950

Low-Firmeness-Alt-Threshold

Mops Reference: SF
Description:
Unit: Feet
Index: Altitude Layer
Definition:

1	2	3	4
200	240	400	480

Alt-Layer-Thr-Top

Mops Reference: TOP
Description:
Unit: Feet
Index: Altitude Layer
Definition:

1	2	3	4
10500	20500	30500	600 000

Alt-Layer-Thr-Bot

Mops Reference: BOT
Description:
Unit: Feet
Index: Altitude Layer
Definition:

1	2	3	4
6000	9500	19500	29500

Descend-Goal

Mops Reference: DESGOAL
Description:
Unit: fpm
Index: RAIND
Definition:

1	2	3	4	8
2000	1000	500	0	-1500

Climb-Goal

Mops Reference: CLIMBGOAL
Description:
Unit: fpm
Index: RAIND
Definition:

1	2	3	4	8
-2000	-1000	-500	0	1500

Descend-Goal

Mops Reference: DESCENDGOAL
Description:
Unit: fpm
Index: RAIND
Definition:

1	2	3	4	8
2000	1000	500	0	-1500

Min-Alt-Rate

Mops Reference: ZDMINTB
Description:
Unit: fpm
Index: VRA
Definition:

1	2	3	4	5
1500	0	-500	-1000	-2000

Max-Alt-Rate

Mops Reference: ZDMAXTB
Description:
Unit: fpm
Index: VRA
Definition:

1	2	3	4	5
-1500	0	500	1000	2000

Appendix C

Reference Algorithms

Nonlinear Tracker

Note: We were not able to include the SARPS document, but it contains the nonlinear tracker reference algorithm that should go here.

Appendix D

Physical Measurement Conventions

Pulse Amplitude is measured at the pulse peak.

Pulse Duration is measured between the half voltage points of the leading and trailing edges.

Pulse Decay Time is measured as the time interval between 90 percent of peak amplitude and 10 percent of peak amplitude on the trailing edge of the pulse.

Pulse-to-Pulse Intervals are measured between the half voltage points of their leading edges.

Phase Reversal Location is the 90-degree point of the phase transition.

Phase Reversal Duration is measured between the 10-degree and the 170-degree points of the transition.

Phase Reversal Intervals are measured between the 90-degree points of the transitions.

Unless otherwise specified, the signal levels specified in this document are defined at an RF reference point at the antenna end of the cable that connects the TCAS interrogator/receiver equipment to its antenna. Specification values in this document are based upon an antenna transmission line loss equal to the maximum for which the TCAS equipment is designed.

> **Note:** The TCAS may be installed with less than the designed maximum transmission line loss. Nevertheless, the standard conditions of

this document are based on the maximum design value. Insertion loss internal to the antenna should be included as part of the net antenna gain.

These performance standards, where applicable are specified for an avionics configuration that includes both a Mode S transponder and TCAS equipment. Design specifications that may exist at a possible interface between the Mode S transponder and the TCAS equipment are not covered in detail.

Appendix E

Performance Requirements on Equipment Interacting with TCAS

This appendix would additional performance requirements on the other aircraft systems or references to where this information could be found.

Appendix F

Glossary

ACAS Airborne Collision Avoidance System. An independent aircraft equipment designed to detect potential conflicting aircraft that are equipped with SSR (Secondary Surveillance Radar) transponders. In this context the term 'independent' means that ACAS operates independently of othe r systems used by air traffic services except for communication with Mode S ground stations.

ACAS Broadcast A long Mode S air-air surveillance interrogation (UF=16) with the broadcast address.

Advisory Message, containing information relevant to collision avoidance, provided to assist pilots in the safe conduct of flight.

Advisory Invalid Indication that RA is no longer valid.

Age of data Time that has elapsed since the measurement time of this data.

Air Traffic Aircraft operating in the air or on an airport surface other than loading ramps and parking areas.

Air traffic control (ATC) A joint civil/military for promoting safe, orderly and expeditious flow of air traffic.

? Aircraft-of-Interest An intruder whose elevation is within 10 degrees and whose relative altitude is within +/- 2000 ft., and for which both the intruder aircraft and own aircraft are at least 600 ft. above ground level.

Air traffic control radar beacon system (ATCRBS) A secondary surveillance radar system having ground-based interrogators and airborne transponders capable of operation on Modes A and C.

Alarm An aural signal to the pilot that recommends immediate attention to the display(s).

Alert Indicator (visual or auditory) which provides information to the flight crew in a timely manner about an abnormal traffic situation. This term is synonymous with 'advisory'.

Altitude Crossing (Crossover) Encounters in which own aircraft and the threat aircraft are projected to cross in altitude prior to reaching closest point of approach.

Altitude Crossing RA A resolution advisory is altitude crossing if own TCAS II aircraft is currently more than 31 m(100 ft) below or above the threat aircraft for upward or downward sense advisories, respectively.

Altitude, relative The altitude of own aircraft measured upward from the intruder aircraft ie., relative altitude is positive when own is higher and negative when own is lower.

Altitude, separation The absolute value of relative altitude.

ATCRBS Air Traffic Control Radar Beacon System compatible transponder. A type of SSR transponder.

Bearing The angle of the intruder aircraft in the horizontal plane, measured clockwise from the longitudinal axis of own aircraft.

Broadcast Unsolicited transmission to a non-specific destination.

Cancel Vertical Resolution Advisory Complement (CVC) Information sent from one TCAS to another via a coordination interrogation to cancel the Vertical Resolution Advisory Complement (VRC) previously sent.

CAS See Collision Avoidance System.

Closest point of approach (CPA) The occurrence of minimum range between own TCAS aircraft and the intruder. Thus range at closest approach is the smallest range between the two aircraft and time of closest approach is the time at which this occurs.

Coasted Track A track that is continued based on previous track characteristics in the absence of surveillance data.

Collision Avoidance System (CAS) Collision avoidance logic subsystem within TCAS.

Coordination The process by which two TCAS equipped aircraft select compatible (nonconflicting) Resolution Advisories by the exchange of Resolution Advisory Complements.

Coordination interrogation A Mode S interrogation (uplink transmission) radiated by TCAS II or III and containing a resolution message.

Coordination reply A Mode S reply (downlink transmission) acknowledging the receipt of a coordination interrogation by the Mode S transponder that is part of a TCAS II or III installation.

Corrective Resolution Advisory A Resolution Advisory that advises the pilot to deviate from current vertical speed, e.g., CLIMB when the aircraft is in level flight.

CPA See Closest point of approach.

Critical Interval Critical interval is the period of time when the horizontal separation between own and the threat aircraft is minimal. True TAU marks the beginning and modified TAU marks the end of this period. Normally the critical interval is very short, only a second or two. However, in cases of slow closure, as a tail chase scenario, horizontal separation may be minimum for a long time. Vertical separation is critical throughout this interval.

Critical NMAC see Critical Near Midair Collision.

Critical Near Midair Collision Situation of aircraft coming within 100 ft of vertical separation and 500 ft of horizontal separation.

Crossover See Altitude crossing.

Cycle The term "cycle" used in this document refers to one complete computation cycle through the sequence of functions executed by TCAS II or TCAS III, nominally one second.

Deferred Resolution Advisory An RA that is deferred from being displayed.

Displayed Resolution Advisory An RA already displayed to the flight crew.

Distance Modification applied to range measurements to account for possible lateral maneuvers. The value of distance modification varies with the sensitivity level for this own-intruder set. The value is chosen such that a sustained acceleration of g/3 will produce a displacement of in threshold time before time to CPA.

Divergence Threshold divergence threshold at and beyond which the aircraft are considered to be diverging from each other, and hence do not warrant an alert.

Effective Sensitivity Level Sensitivity level of an aircraft derived from a composite of its pilot input, any ground station (up to 15) uplink, and altitude based sensitivity level. This composite level is derived using the set of rules specified by the TCAS requirements. This level is used in determining the sensitivity of performance of collision avoidance logic.

Elapsed time of alert Time since issuance of this alert.

Escape Maneuver See Resolution Maneuver.

Established threat An intruder that has been declared a threat and still merits a Resolution Advisory.

Established Track A track generated by TCAS air-to-air surveillance that is treated as the track of actual aircraft.

FL see Flight Level.

Failure The inability of a system, subsystem, unit, or part to satisfactorily perform within the specified constraints.

False advisory An advisory caused by a false track or a TCAS malfunction.

False track A track created by erroneous surveillance data.

Flight Level (FL) Flight level refers to a level of constant atmospheric pressure, usually at an altitude above 18,000 feet above mean sea level, for which a pressure of 29.92 inches of mercury is the zero reference level.

FRUIT See Garble, nonsynchronous.

Garble, nonsynchronous Reply pulses received from a transponder that is being interrogated from some other source. Also called FRUIT.

Garble, synchronous An overlap of the reply pulses received from two or more transponders answering the same interrogation.

Horizontal Miss Distance (HMC) see Range at co-altitude.

IMC see Instrument Meteorological Conditions.

Incorrect Resolution Advisory A Resolution Advisory which occurs when a threat is present, but, because of a failure of the installed TCAS II, Mode S transponder, or associated sensors, commands a maneuver which reduces separation to the threat.

Increased rate Resolution Advisory A resolution advisory that advises the pilot to increase the altitude rate to a value exceeding that of a previous positive RA.

Individual Resolution Advisory Resolution advisory derived considering only one single threat.

Instrument Meteorological Conditions (IMC) are ceilings, visibilities, or cloud clearances less than the minima specified for VFR (Visual Meteorological Rules) flight.

Intruder A transponder-equipped aircraft within the surveillance range of TCAS for which TCAS has an established track.

Mode A Reply to a type of secondary surveillance radar (SSR) equipment known as Mode A transponder which provides a reply with Mode A identification data, and with open brackets for altitude.

Mode C Reply to a type of secondary surveillance radar (SSR) equipment known as ATCRBS which provides a reply with both identification and altitude information when interrogated.

Mode S Type of secondary surveillance radar (SSR) equipment which provides replies to Mode A and Mode C interrogations and discrete address interrogations from the ground or air.

Mode S discrete address or id A unique 24 bit address assigned to each Mode S equipped aircraft. This allows the Mode S beacon to interrogate one aircraft at a time (by specifying that aircraft's Mode S address). The 24 bit address allows over 16,000,000 unique aircraft addresses.

Modified TAU Time to closest point of approach assuming lateral closure of intruder aircraft. See Distance Modification.

Negative Advisory One of the following resolution advisories: DON'T CLIMB, DON'T DESCEND. A negative advisory can be either preventive or corrective.

Operational Mode TCAS mode of operation. TCAS may be in one of the three modes; standby, TA only or TA and RA mode.

Own aircraft The TCAS equipped reference aircraft.

Performance Monitoring A feature of the TCAS equipment that implements the function of measuring critical physical or software TCAS quantities to determine the operating capability of the TCAS equipment. The performance monitoring function is initiated routinely and automatically by the TCAS equipment; no flight crew or external stimulation is required. The performance monitor feature of the TCAS equipment also provides to the pilot an indication of the operating status of the equipment.

Positive Resolution Advisory One of the following resolution advisories: CLIMB, DESCEND. A positive advisory can be either preventive or corrective.

Potential threat An intruder that has passed the Potential Threat classification criteria, and does not meet the Threat Classification criteria.

Preventive Resolution Advisory A Resolution Advisory that advises the pilot to avoid certain deviations from the vertical rate because certain vertical speed restriction exist. Example: a DON'T CLIMB when the aircraft is level.

? Protected volume A volume of airspace enclosing the TCAS aircraft which, when penetrated by or containing an intruder, will normally result in the generation of a Traffic Advisory or a Resolution Advisory. The protected volume is generated from a larger volume produced by a range test which is then reduced in size by the application of an altitude test.

Proximate traffic Nearby aircraft within 1200 feet and 6 NMI which do not meet either the threat or the potential threat classification criteria.

Range At Co-altitude Also called HMD or Horizontal Miss Distance. Range at co-altitude is a measure of horizontal proximity when own and intruder aircraft are predicted to be at the same altitude.

Relative Altitude see Altitude, Relative.

Resolution Advisory (RA or Vertical Resolution Advisory) Oral and visual information given to the flight crew recommending a maneuver intended to provide separation from all threats, or a maneuver restriction to maintain existing separation, in order to avoid a potential collision. Positive, negative and vertical speed limit (VSL) advisories constitute the resolution advisories. A resolution advisory may also be corrective or preventive.

Resolution Advisory Complement (RAC) Information returned by one TCAS to a TCAS or ground station interrogation in a coordination reply message via Mode S. TCAS uses the reply only to verify receipt of its coordination interrogation by the other TCAS.

Resolution display A display which depicts caution or warning areas above or below the TCAS equipped aircraft.

Resolution Maneuver Maneuver in the vertical direction resulting from compliance with a Resolution Advisory.

Resolution message The message containing the Resolution Advisory Complement.

Risk ratio The measure of comparison of the risk of encountering a critical NMAC with TCAS to that without is called Risk ratio.

Self-test Tests of the TCAS equipment and displays which are initiated by the flight crew and are used to determine the operational status of the equipment. Self-test differs from the Performance Monitoring in that it is initiated by the flight crew, and may use external stimuli and is not performed continually or automatically.

Sense A direction that a resolution advisory may take the aircraft. The sense of a TCAS II Resolution Advisory is "upward" if it requires climb or limitation of descent rate and "downward" if it requires descent or limitation of climb rate. I.e., either up or down relative to the existing flight path of the aircraft.

Sense reversal A change to the sense opposite that of the original Resolution Advisory.

Sensitivity level (S or SL) An integer defining a set of parameters used to specify the sizes of the protected volumes around the TCAS-equipped aircraft for traffic advisory detection and collision avoidance.

Sensitivity level command (SLC) An instruction given to the TCAS equipment for control of parameters used in threat detection (and hence the level of collision avoidance protection).

Squitter Spontaneous transmission generated once per cycle by Mode S transponders.

Standby mode An operational mode of TCAS. TCAS does not issue any advisories nor does it issue any interrogations in this mode.

TA-Only mode A TCAS mode of operation in which TAs are displayed and RAs are not displayed; any threats that would normally qualify for an RA are displayed as TAs.

TA-RA mode A TCAS operation mode in which both TAs and RAs are issued as specified.

TCAS Traffic Alert and Collision Avoidance System. An ACAS system that uses the transponder onboard the intruder aircraft as the basis for detecting the presence of an intruder and estimating its position. In this document TCAS and TCAS II are used interchangeably.

TCAS I A TCAS which provides Traffic Advisories as an aid to initiate "see and avoid" action but does not include the capability for generating Resolution Advisories.

TCAS II A TCAS which provides vertical Resolution Advisories in addition to Traffic Advisories.

TCAS III A TCAS which provides vertical and horizontal Resolution Advisories in addition to Traffic Advisories.

TCAS Broadcast A long Mode S air-to-air surveillance interrogation (UF = 16) with the broadcast address.

TCAS Capability An indication of whether there is an operating TCAS with at least a vertical resolution advisory capability on board the reference aircraft.

TCAS Computation Cycle see Cycle.

Threat An intruder that has passed the Threat Classification criteria.

Track A sequence of at least three measurements (range and altitude) representing positions that could reasonably have been occupied by an aircraft.

Traffic An aircraft within the surveillance range of TCAS.

Traffic advisory (TA) Information given to the flight crew identifying the approximate positions, relative to the TCAS-equipped aircraft, of aircraft meeting the Traffic Advisory criteria. No resolution advisory information is conveyed by a Traffic Advisory.

Traffic display Information given to the pilot pertaining to the position of another aircraft in the immediate vicinity. The information contains no suggested maneuver.

Transponder, ATCRBS See ATCRBS.

Transponder, Mode S See Mode S.

Transponder Equipage Indication of type of transponder, whether ATCRBS or Mode S.

True TAU Time to closest point of approach assuming straight flight of the intruder.

VMD see Vertical Miss Distance.

Vertical Miss Distance The relative altitude between own and intruder aircraft at closest point of approach.

Vertical Resolution Advisory Complement (VRC) Information provided by one TCAS to another TCAS via a coordination interrogation in order to ensure complementary maneuvers by restricting the choice of maneuvers available to the TCAS receiving the Vertical Resolution Advisory Complement.

VRC See Vertical Resolution Advisory Complement.

VSI Vertical Speed Indicator.

Vertical Speed Limit (VSL) advisory Resolution advisories such as DON'T CLIMB > 500 FPM, which limit the vertical speed of the aircraft. A VSL advisory may be preventive or corrective.

Warning time The time interval projection between penetration of the protected volume by the intruder, i.e., the intruder meets the detection criteria, and closest approach when neither aircraf t accelerates aircraft accelerates.

Whisper-shout (WS) A method of controlling synchronous garble from ATCRBS transponders, through the combined use of variable power levels and suppression pulses.

Index

Appendix G

Index

Index