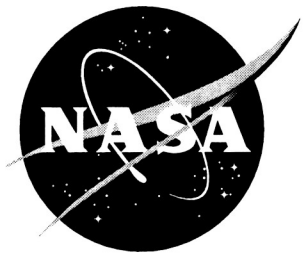
NASA/CR-2003-212426



Flight Guidance System Requirements Specification

Steven P. Miller, Alan C. Tribble, Timothy M. Carlson, and Eric J. Danielson
Rockwell Collins, Cedar Rapids, Iowa

June 2003

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

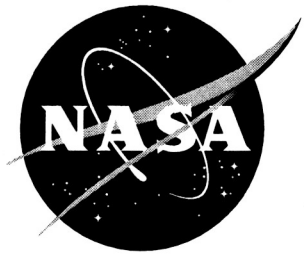
- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Phone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/CR-2003-212426



Flight Guidance System Requirements Specification

Steven P. Miller, Alan C. Tribble, Timothy M. Carlson, and Eric J. Danielson
Rockwell Collins, Cedar Rapids, Iowa

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Cooperative Agreement NCC1-01001

June 2003

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Abstract

This report describes a requirements specification written in the RSML^{-e} language for the mode logic of a Flight Guidance System of a typical regional jet aircraft. This model was created as one of the first steps in five-year project sponsored by the NASA Langley Research Center, Rockwell Collins Inc., and the Critical Systems Research Group of the University of Minnesota to develop new methods and tools to improve the safety of avionics designs. This model will be used to demonstrate the application of a variety of methods and techniques, including safety analysis of system and subystem requirements, verification of key properties using theorem provers and model checkers, identification of potential sources mode confusion in system designs, partitioning of applications based on the criticality of system hazards, and autogeneration of avionics quality code. While this model is representative of the mode logic of a typical regional jet aircraft, it does not describe an actual or planned product. Several aspects of a full Flight Guidance System, such as recovery from failed sensors, have been omitted, and no claims are made regarding the accuracy or completeness of this specification.

Contents

1 Introduction **5**

1.1 How Requirements Analysis Relates to Aircraft Safety 5
1.2 Overview of a Flight Guidance System 9
1.3 Fundamentals of the Mode Logic 12

2 The FGS Requirements Specification **15**

2.1 Overview of the FGS Specification 16
2.2 Prioritization of Events 20
2.3 Basic Definitions 23
2.4 Flight Director (FD) 25
2.5 Pilot Flying (PF) 29
2.6 Independent Mode 30
2.7 Flight Modes 32
2.7.1 Lateral Modes 35
2.7.1.1 Roll Hold (ROLL) Mode 37
2.7.1.2 Heading Select (HDG) Mode 40
2.7.1.3 Navigation (NAV) Mode 44
2.7.1.4 Lateral Approach (LAPPR) Mode 50
2.7.1.5 Lateral Go Around (LGA) Mode 56
2.7.2 Vertical Modes 60
2.7.2.1 Pitch Hold (PITCH) Mode 62
2.7.2.2 Vertical Speed (VS) Mode 65
2.7.2.3 Flight Level Change (FLC) Mode 69
2.7.2.4 Altitude Hold (ALT) Mode 73
2.7.2.5 Altitude Select (ALTSEL) Mode 77
2.7.2.6 Vertical Approach (VAPPR) Mode 85
2.7.2.7 Vertical Go Around (VGA) Mode 91
2.8 Flight Control Laws 95
2.9 Flight Control Panel (FCP) 103
2.10 Navigation Sources 127
2.11 Air Data System 128
2.12 References 129

2.13 Autopilot (AP) 130
2.14 Offside FGS 133
2.15 FGS Inputs 152
2.16 FGS Outputs 156
References 161
Index 163

Chapter 1

Introduction

This report describes a requirements specification written in the RSML^{-e} [8, 6] language for the mode logic of a Flight Guidance System of a typical regional jet aircraft. This model was created as one of the first steps in a project sponsored by the NASA Langley Research Center, Rockwell Collins, Inc., and the Critical Systems Research Group of the University of Minnesota to develop new methods and tools to improve the safety of avionics designs. This model will be used to demonstrate the application of a variety of methods and techniques, including:

- Safety analysis of system and subsystem requirements
- Verification of key properties using theorem provers and model checkers
- Identification of potential sources mode confusion in system designs
- Partitioning of applications based on the criticality of system hazards
- Autogeneration of avionics quality code

While this model is representative of the mode logic of a typical regional jet aircraft, it does not describe an actual or planned product. Several aspects of a full Flight Guidance System, such as recovery from failed sensors, have been omitted, and no claims are made regarding the accuracy or completeness of this specification.

The report is organized as follows. In this chapter, Section 1.1 discusses why the modeling and analysis of requirements is important to aircraft safety, Section 1.2 provides an overview of a modern Flight Guidance System, and Section 1.3 discusses the the fundamentals of the mode logic found in the FGS. Chapter 2 deals with the requirements specification itself. Section 2.1 describes how the specification is organized. The remaining sections of Chapter 2 are the specification itself.

1.1 How Requirements Analysis Relates to Aircraft Safety

Aircraft safety has improved steadily over the last few decades and much of this improvement can be attributed to the introduction of advanced automation in the cockpit. The predicted

Accident Category	Fatalities	Accidents
Controlled Flight Into Terrain	2111	28
Loss of Control in Flight	2011	29
In Flight Fire	600	3
Midair Collision	506	2
Fuel Tank Explosion	238	2
Landing	~200	14
Takeoff	~144	3
Ice/Snow	~60	3
Fuel Exhaustion	~40	5
Wind Shear	~30	2
Runway Incursion	~10	4
Misc. Fatality	~10	3
On Ground	~10	3
Refused Takeoff	~10	1
Turbulence	~10	3
Unknown	482	7
Total	6465	112

Table 1.1: Fatal Accidents - Worldwide Commercial Jet Fleet, 1990-1999 - From [14]

ten-fold increase in air traffic by 2016 will force much greater reliance on automated systems. New technologies, such as: Communication, Navigation, and Surveillance for Air Traffic Management (CNS/ATM); Global Air Traffic Management (GATM); Global Positioning System (GPS); and fly-by-wire, will be introduced to meet enhanced safety and performance goals. Table 1.1 classifies the fatal accidents of the worldwide commercial jet fleet for the period of 1990 to 1999. Categories in which automated systems can play a role in reducing the accident rate still further are emphasized. These account for almost 80 percent of the accidents and fatalities.

However, the growing complexity and integration issues associated with these advanced technologies also increase the potential for errors that could have a direct impact on safety. As systems become more tightly integrated, even the engineers that design them will find it difficult to anticipate every possible interaction. Unfortunately, it is precisely these interactions that will have the greatest impact on safety:

> *Today more accidents result from dangerous design characteristics and interactions among components.[10]*

> *Complex and highly integrated avionics present greater risk for development error. With non-traditional human-machine interfaces, there is also the potential for operational flight crew errors. Moreover, integration of systems may result in a greater*

likelihood of undesirable and unintended effects.[2]

In spite of the increased safety concerns, manufacturers are being asked to design, develop and validate the systems of the future under the budgets of the past. It is imperative that new methods to improve system safety be developed, and that these methods work in concert with the goals of reducing cost and cycle time.

One of the best ways to improve product quality while reducing costs is to develop a complete, consistent, and well organized set of requirement at the start of the product life cycle. As stated by Fred Brooks:

> *The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements... No other part of the work so cripples the resulting system if done wrong. No other part is as difficult to rectify later.[4]*

The majority of software development errors stem from logic errors made during requirements analysis. Most of these errors are not found until the later phases of a project. The amount of rework that has to be done to fix a requirements error, and the cost of doing so, grows dramatically the later it is detected.[1] In one well-known study, it was found that it costs ten times as much to correct a requirements error during unit testing as during requirements analysis. Correcting a requirements error after a product had been deployed increased the cost by 100 to 200 times.[3] Moreover, requirements errors are often the most serious errors. Investigators focusing on safety-critical systems have found that requirements errors are more likely to affect the safety of an embedded system than errors introduced during design or implementation.[12]

One of the most successful approaches to finding requirements errors is to create a precise model of the system's externally visible behavior that can be executed. Several notations have been developed over the last few years for creating a model of the system's functional behavior that is both readable and mathematically precise. Among the better known are SCR [7], RSML [8, 6], SpecTRM [10], and Statecharts [5]. Creating models based on these notations have been shown to find a wealth of errors in textual specifications [6, 13]. Moreover, such models can be connected to a mock-up of the user interface and executed with the customer in the same way as a simulation. In the best approaches, the underlying notation has been carefully designed to support automated analyses. These make possible a variety of consistency and completeness checks that find many errors, as well as the ability to check for properties specific to the application being modeled. Finally, the requirements model itself becomes a detailed statement of the desired behavior. This enhances design and testing, and makes it far more feasible to outsource the software development.

In short, creation of a precise model of a system's behavior not only finds errors early in the life-cycle when they can be most economically addressed, it enables a variety of downstream activities tradionally associated with the quality of the system, including design, coding, and verification. Particularly relevant to this report, it lays the foundation for improving system safety through analysis:

> *Every hazard analysis requires some type of model of the system, which may range from a fuzzy idea in the analyst's mind to a complex and carefully specified mathematical*

model. The model may range from a high-level abstraction to a low-level and detailed prototype. Nevertheless, information about the system must exist in some form, and that constitutes the system model upon which the analysis is performed.[11]

The model of a Flight Guidance System described in this report will be used for the demonstration of a wide variety of techniques developed to improve the quality and safety of embedded systems, thereby reducing the accident rate shown in Table 1.1.

1.2 Overview of a Flight Guidance System

A Flight Guidance System (FGS) is a component of the overall Flight Control System (FCS). It compares the measured state of an aircraft (position, speed, and attitude) to the desired state and generates pitch and roll guidance commands to minimize the difference between the measured and desired state. A simplified overview of an FCS that emphasizes the role of the FGS is shown in Figure 1.1.

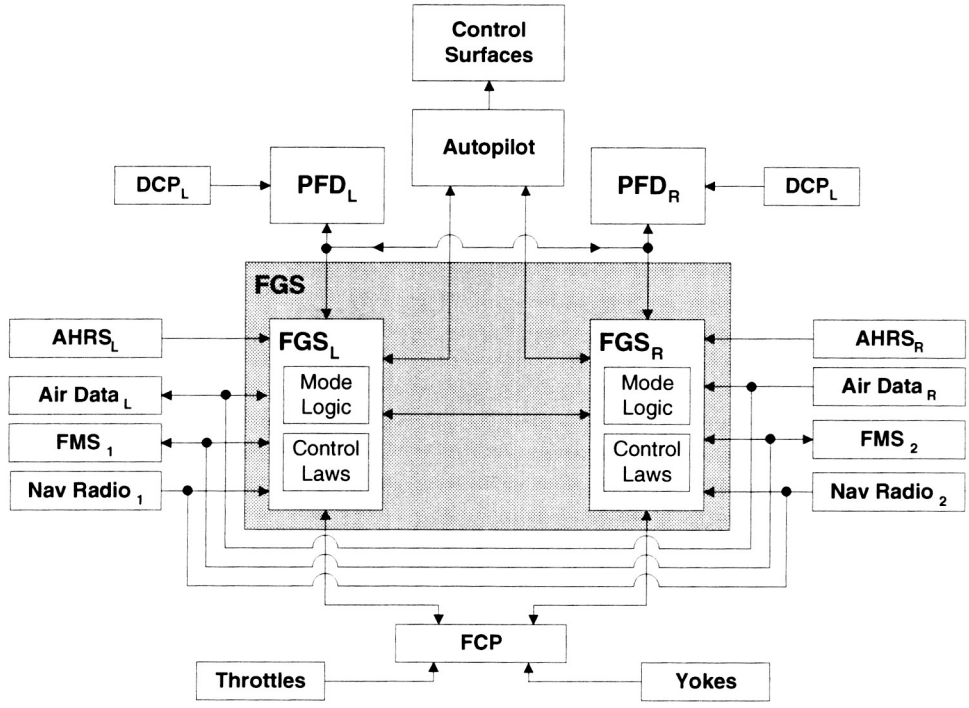


Figure 1.1: Flight Control System Overview

As shown in Figure 1.1, the FGS subsystem accepts input about the aircraft's state from the Attitude Heading Reference System (AHRS), Air Data System (ADS), Flight Management System (FMS), and Navigation Radios. Using this information, it computes pitch and roll guidance commands that are provided to the Autopilot (AP). When engaged, the Autopilot translates these commands into movement of the aircraft's control surfaces necessary to achieve the commanded changes about the lateral and vertical axes.

The flight crew interacts with the FGS primarily through the Flight Control Panel (FCP), shown in more detail in Figure 1.2. The FCP includes switches for turning the Flight Director (FD) on and off, switches for selecting the different flight modes such as vertical speed (VS), lateral navigation (NAV), heading select (HDG), altitude hold (ALT), and approach (APPR), the Vertical Speed/Pitch Wheel, and the autopilot disconnect bar. The FCP also supplies feedback to the crew, indicating selected modes by lighting lamps on either side of a selected mode's button. Figure 1.2 depicts a configuration in which Heading Select (HDG) mode is selected.

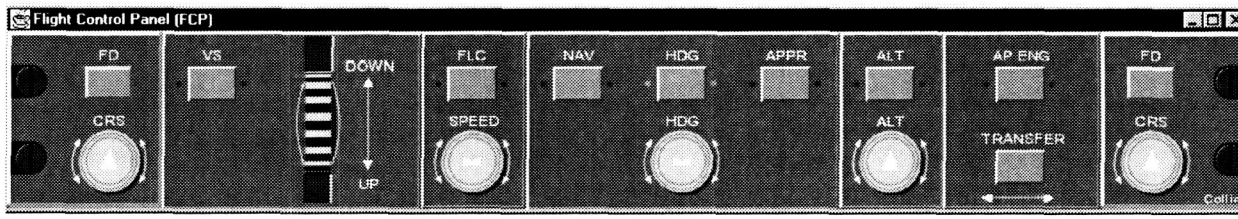


Figure 1.2: Flight Control Panel

A few key controls, such as the Go Around button and the Autopilot Disengage switch, are provided on the control yokes and throttles and routed through the FCP to the FGS. Navigation sources are selected through the Display Control Panel (DCP), with the selected navigation source routed through the PFD to the FGS.

The FGS has two physical sides, or channels, one on the left side and one on the right side of the aircraft (see Figure 1.1). These provide redundant implementations that communicate with each other over a cross-channel bus. Each channel of the FGS can be further broken down into the mode logic and the flight control laws. The flight control laws accept information about the aircraft's current and desired state and compute the pitch and roll guidance commands. The mode logic determines which lateral and vertical modes of operation are active and armed at any given time. These in turn determine which flight control laws are active and armed. These are annunciated, or displayed, on the Primary Flight Displays (PFD) along with a graphical depiction of the flight guidance commands generated by the FGS.

A simplified image of a Primary Flight Display (PFD) is shown in Figure 1.3. The PFDs display essential information about the aircraft, such as airspeed, vertical speed, attitude, the horizon, and heading. The active lateral and vertical modes are displayed (annunciated) at the top of the display. The annunciations in Figure 1.3 indicate that the current active lateral mode is Heading Select (HDG), the active vertical mode is Pitch (PTCH), and that Altitude Select (ALTS) mode is armed.

The large sphere in the center of the PFD is the sky/groundball. The horizontal line across its middle is the artificial horizon. The current pitch and roll of the aircraft is indicated by a white wedge $\wedge$ representing the aircraft in the middle of the sky/ground ball. Figure 1.3 depicts an aircraft with zero degrees of roll and pitched up approximately five degrees.

The graphical presentation of the pitch and roll guidance commands on the PFD are referred to as the Flight Director (FD). [1] The pitch and roll guidance commands are shown as a magneta wedge $\wedge$ in the sky/ground ball. When the autopilot is not engaged, these are interpreted as guidance to the pilot. When the autopilot is engaged, these indicate the direction the aircraft is being steered by the autopilot. Figure 1.3 depicts an aircraft in which the autopilot is not engaged and the Flight Director is commanding the pilot to pitch up and roll to the right.

[1] The term Flight Director is also commonly used to refer to the logic that computes the pitch and roll guidance commands.

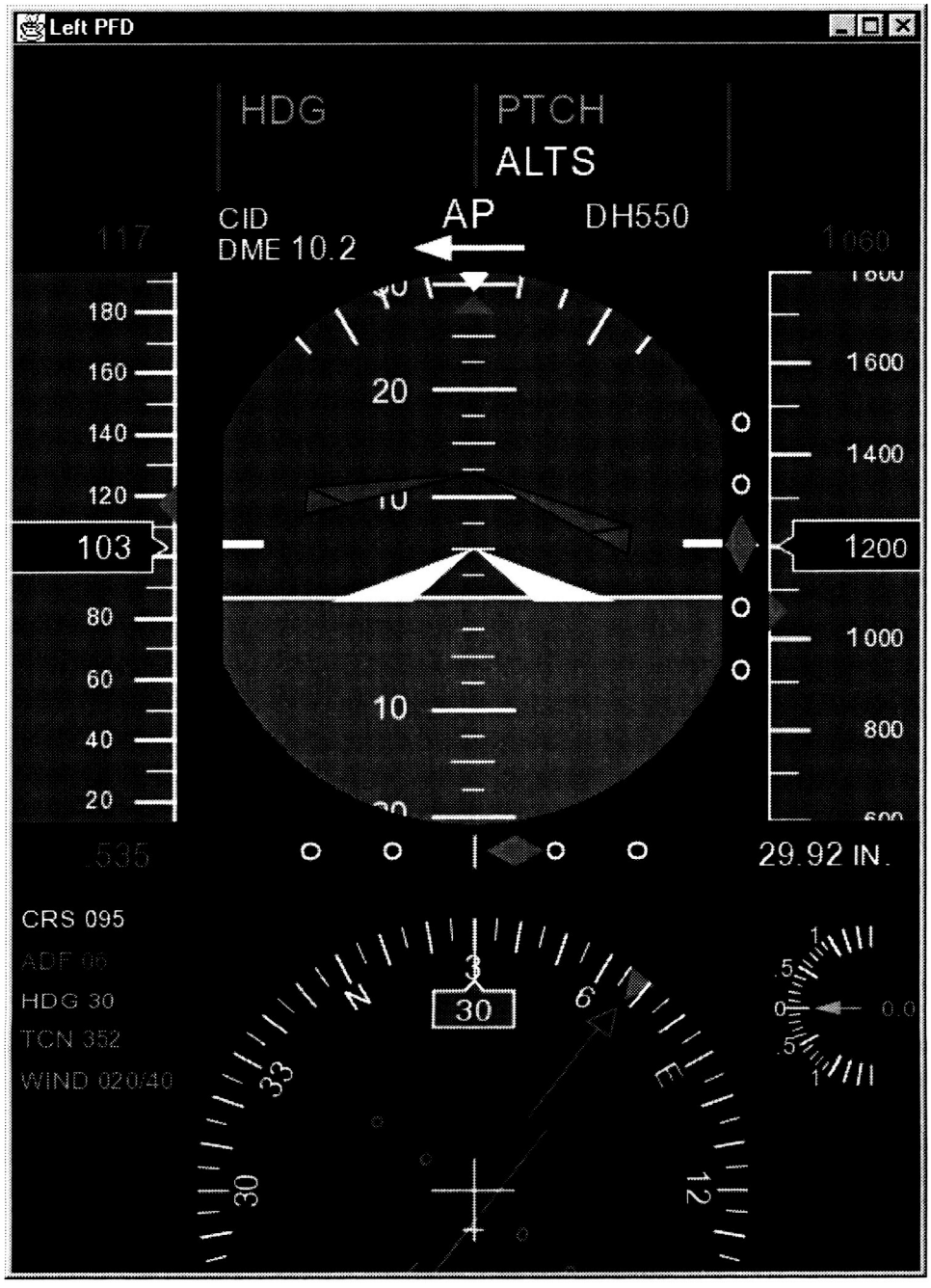


Figure 1.3: Primary Flight Display

1.3 Fundamentals of the Mode Logic

A *mode* is defined by Leveson as a "mutually exclusive set of system behaviors"[9]. Specifically as it relates to an FGS, advisory circular AC/ACJ 25.1329 defines a mode as "a system configuration that corresponds to a single (or set of) FGS behavior(s)" [2]. The primary modes of interest in an FGS are the lateral and vertical modes. The lateral modes control the behavior of the aircraft about the longitudinal, or roll, axis, while the vertical modes control the behavior of the aircraft about the vertical, or pitch, axis. In addition, there are a number of auxillary modes, such as half-bank mode, that control other aspects of the aircraft's behavior. Examples of FGS modes include Heading Hold (HDG) which holds the aircraft to a selected heading and Vertical Speed (VS) mode which holds the aircraft to selected vertical speed.

A mode is said to be *selected* if it has been manually requested by the flight crew or if it has been automatically requested by a subsystem such as the FMS. The simplest modes have only two states, *cleared* and *selected*, as shown in Figure 1.4. Such a mode becomes active immediately upon selection with it's associated flight control law providing guidance commands to the flight director and, if engaged, the autopilot. When cleared, a mode's associated flight control law is non-operational, i.e., it does not generate any outputs.

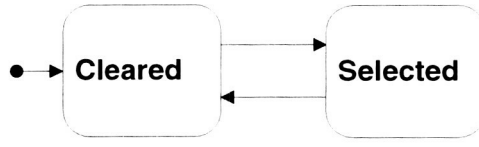


Figure 1.4: A Simple Mode

Some modes can be *armed* to become active when a criterion is met, such as the acquisition of a navigation source or proximity to a target reference such as a desired altitude. Such modes have three states as shown in Figure 1.5. The two states *armed* and *active* are substates of the *selected* state, i.e., when the mode is armed or active, it is also said to be selected. While in the armed state, the mode's flight control law is not generating guidance commands for the flight director or the autopilot, but it may be accepting inputs, accumulating state information, and helping to determine if the criterion for becoming active is met. Once the criterion is met, the mode transitions to the active state and its flight control law begins generating guidance for the flight director and autopilot. Note that the only way to exit the active state is to deselect the mode, i.e., it is not usually possible to revert directly from the active state to the armed state.

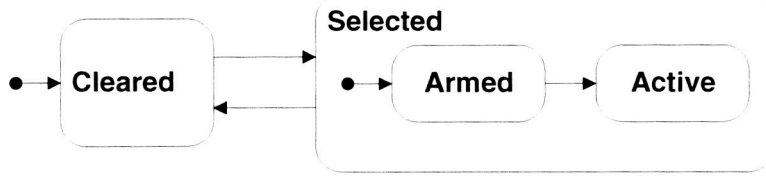


Figure 1.5: A Mode with Armed and Active Substates

Some modes also distinguish between capturing and tracking of the target reference or navigation source. Such a mode is shown in Figure 1.6. Once in the active state, such a mode's flight control law first *captures* the target by manuvering the aircraft to align it with the navigation source or reference. Once correctly aligned, the mode transitions to the *tracking* state in which it holds the aircraft on the target. Both the *capture* and *track* states are substates of the *active* state and the mode's flight control law is active in both states, i.e., generating guidance commands for the flight director and autopilot. Note that the only way to exit the active, track, or capture states is to deselect the mode, i.e., it is not possible to revert directly from the track state to the capture state or from the active state to the armed state.

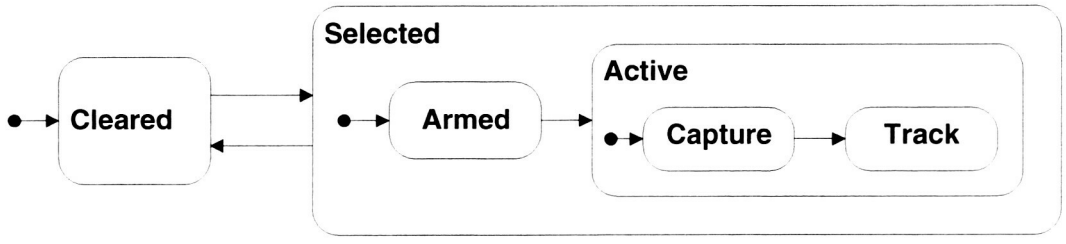


Figure 1.6: A Mode with Capture and Track Substates

The *mode logic* consists of all the available modes and the rules for transitioning between them. Figure 1.7 provides an overview of the modes described in this specification. Traditionally, aircraft modes are associated with a flight control law that determines the guidance provided to the flight director or autopilot. For example, in Figure 1.7, there are lateral modes of Roll Hold, Heading Hold, Navigation, Lateral Approach, and Lateral Go Around. These control the guidance about the longitudinal, or roll, axis. Guidance about the vertical, or pitch, axis is controlled by the vertical modes of Pitch, Vertical Speed, Altitude Hold, Altitude Select, Vertical Approach, and Vertical Go Around. Each of these are associated with one or more control laws.

Also shown in Figure 1.7 are several auxillary modes such as Pilot Flying, Independent Mode, FD, and AP, that describe the status of key state variables. While these have an important affect upon the behavior of the aircraft, they are not directly associated with specific flight control laws, and are not usually viewed as "modes" by system designers in the same way the lateral and vertical modes are.

In order to provide effective guidance of the aircraft, these modes are tightly synchronized so that only a small portion of their total state space is actually reachable. For example, to ensure that meaningful guidance is provided to the flight director and autopilot, only one lateral and one vertical mode can be active at any time. For the same reason, if the autopilot is engaged or the flight director is turned on, at least one lateral and one vertical mode must be active. Other constraints enforce sequencing of modes that are dictated by the characteristics of the aircraft and the airspace. For example, vertical approach mode is not usually allowed to become active until lateral approach mode has become active to ensure that the aircraft is horizontally centered on the localizer before tracking the glideslope. These constraints are clearly important to safe flight, and can become quite complex. The mode logic is responsible for enforcing all these constraints.

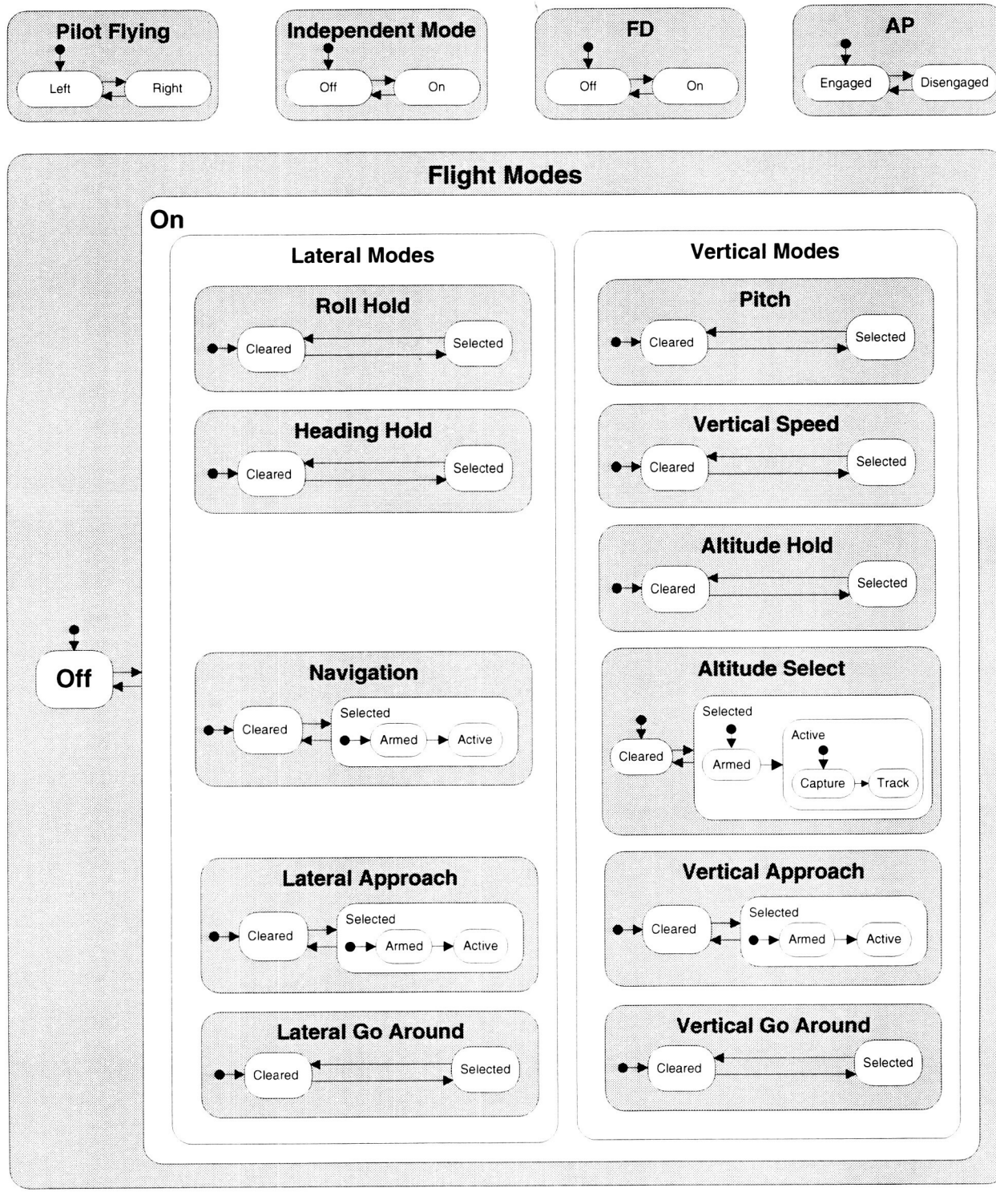


Figure 1.7: FGS Mode Logic

Chapter 2

The FGS Requirements Specification

A requirements specification must meet the needs of a diverse set of stakeholders, including but not limited to the customer, end users, program management, systems engineers, software engineers, hardware engineers, test engineers, and regulatory agencies. It must be clear enough that end users can understand it, yet complete and precise enough that the engineers can implement the system correctly and develop a comprehensive set of test cases. It must be robust in the face of change so that small changes only require small efforts. Ideally, the specification should support the development of a family of products, so that it can easily be reused to develop an entire line of similar products.

The FGS requirements specification described in this document is organized to meet these goals. Its structure is hierarchial, allowing the reader to start with the entire subsystem, then examine individual components in greater detail. To enhance readability, it makes heavy use of the tabular formats developed in RSML^{-e} to make the complex logic of the Traffic Collision Avoidance System (TCAS) accessible to pilots, engineers, and regulatory authorities.[6]

Some of these goals, particularly those related to robustness in the face of change and support for product families, are best achieved by organizing the specification into pieces that are logically related and likely to change together and by defining interfaces between these components that are unlikely to change. The current version of RSML^{-e} does not directly support these notions, though extensions are planned for future versions that will.

However, this specification has been written as though these features were available. Functions that are logically related are grouped together into components, with details that are likely to change encapsulated within these components. Rather than referring to these directly in the other parts of the specification, interfaces (e.g., functions or macros) are defined that should remain stable over time for use in the rest of the specification. Many of the architectural decisions are based on an extensive commonality analysis of several Flight Guidance Systems to determine which features change and which remain stable over a period of time. To facilitate such anticipated changes, the architecture implicitly defines a framework in which common patterns and interfaces are used across similar components.

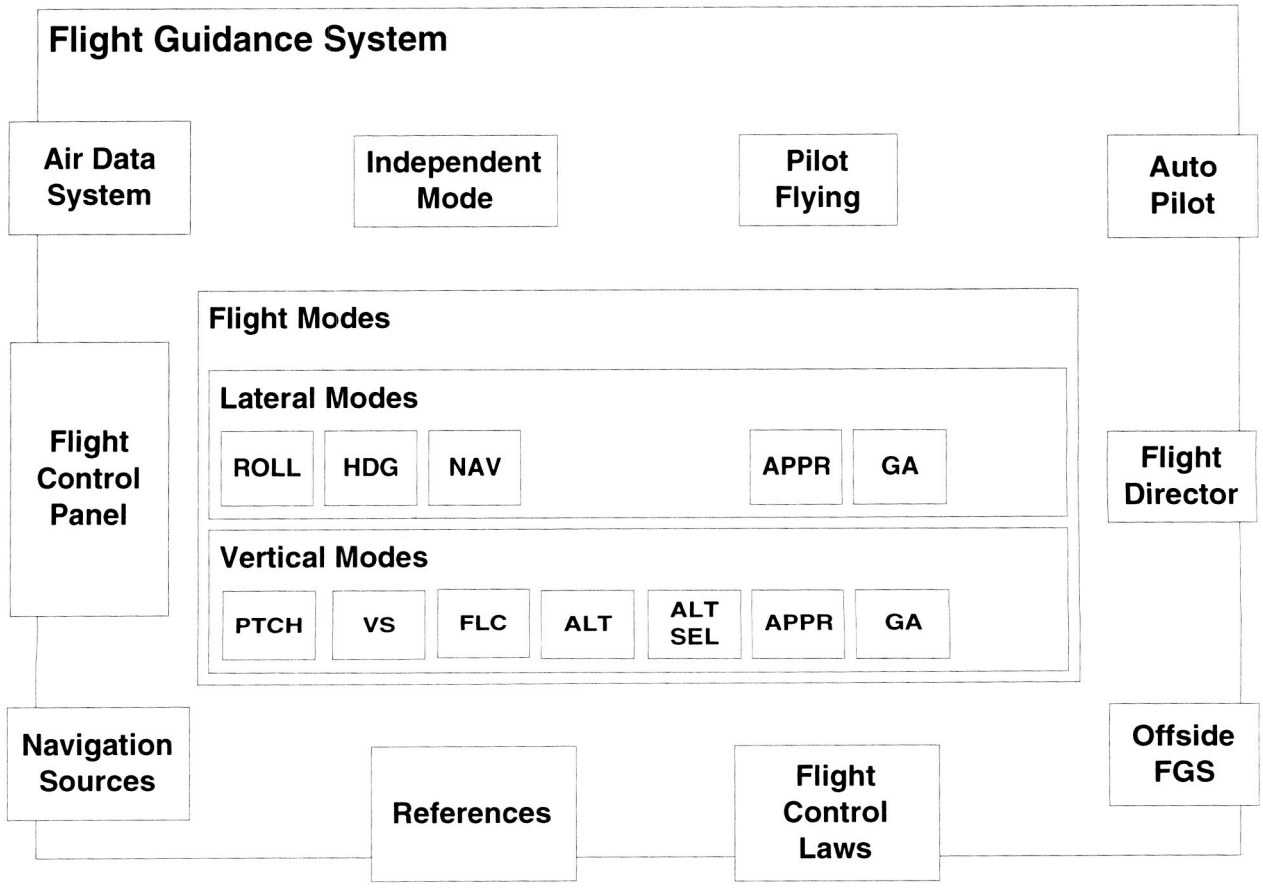


Figure 2.1: Flight Guidance Specification Overview

2.1 Overview of the FGS Specification

The top level structure of the FGS is shown in Figure 2.1. The enclosing box labeled *Flight Guidance System* represents the full specification. The components located on the edge of the Flight Guidance System are *boundary* components that define the system's view of its interfaces with the outside world. These do not define the behavior of the actual external subsystem. Rather, they define the behavior of that subsystem that the FGS must be aware of. For example, the component labeled *Flight Control Panel* (FCP) only defines what information is received by and sent to the Flight Control Panel and does not define the full behavior of that actual device.

Each component may define new events or values that are used by other components. For example, the FCP mentioned above defines and exports events for each switch that can be pressed on the panel. In fact, since several of these events can occur at the same time, the FCP component defines a prioritization of these events so that only the highest priority events are seen by the rest of the specification (see Section 2.2).

The heart of the specification is the definition of the mode logic contained in the *Modes*

component. This is further broken down into the *Lateral Modes* and *Vertical Modes*, which are in turn broken down into individual modes. The lateral and vertical modes all present standard interfaces to their parent modules, making it staightforward to add, remove, or substitute modes, a common source of variation in a FGS. In addition, many of the modes share a common behavior that can be specified once, then tailored to define the behavior of each individual mode.

The following sections provide an overview of each individual component. Detailed specifications of the components can be found in the remainder of this chapter.

Flight Control Panel (page 103) The Flight Control Panel (FCP) is the primary device with which the flight crew interacts with the FGS. This component defines the FGS's view of the FCP. This consists of the definition of the monitored variables representing the switches and dials of the FCP and the controlled variables representing the lamps displayed on the FCP. A few additional inputs, such as the SYNC and AP Disengage switches on the control yokes and the GA switches on the throttles, are routed through the FCP aand are defined here. This component also defines FCP specific events, such as the pressing of a switch, used in the rest of the specification. These events are prioritized so that if multiple events occur at the same time, only the most significant events are seen by the other parts of the specification.

Flight Director (page 25) The Flight Director (FD) displays the pitch and roll guidance commands to the pilot and copilot on the Primary Flight Display. This component defines when the Flight Director guidance cues are turned on and off.

Independent Mode (page 30) Normally, the FGS operates in dependent mode in which the pilot flying (PF) side of the FGS provides the mode annunciations and FD guidance commands to both PFDs. While in dependent mode, the pilot not flying (PNF) side of the FGS synchronizes its modes to the pilot flying side. Under final approach conditions and during takeoff or go around modes, the FGS operates in independent mode in which each side operates independently of the other side and provides mode annunciations and FD guidance commands to its own PFD. This component defines when the FGS enters independent mode.

Pilot Flying (page 29) The transfer switch on the FCP is used to determine which FGS provides mode annunciations and guidance commands to the PFDs while in dependent mode. This component defines which side is the Pilot Flying side. It also defines the notion of the "active" side of the FGS. A side of the FGS is said to be "active" if it is providing mode annunciations and guidance commands to its PFD.

Flight Control Laws (page 95) The Flight Control Laws generate the pitch and roll guidance commands provided to the Flight Director and the Autopilot. Different Flight Control Laws are armed or activated based on the current modes of the FGS. They also provide inputs to the mode logic that determine when transitions are made from the armed to the active state. While the Flight Control Laws are not specified in this model, this component defines the interface between them and the mode logic and defines the events provided by

the Flight Control Laws that the mode logic depends upon. As with events provided by the Flight Control Panel, some of these are prioritized so that if multiple events occur at the same time, only the most significant events are seen by the other parts of the specification.

Navigation Sources (page 127) The FGS receives navigation information from several sources, including VHF Omni-Directional Range (VOR) for lateral navigation, Localizer (LOC) for precision lateral approach, and Glideslope (GS) for precision vertical approach. This component defines the FGS's view of its navigation sources. This includes definition of the types of sources available, the currently selected source, and the information available from that source.

Air Data System (page 128) The Air Data System provides information about the aircraft state sensed from the surrounding air mass such as pressure altitude and indicated airspeed. This component defines the FGS's view of the Air Data System.

References (page 129)

References refer to target values used by several of the modes, such as the Preselected Altitude. These may be set by the flight crew through the Flight Control Panel, or set to the current aircraft value on entry to a mode or a synchronization request. The references may be physically maintained by a variety of systems. This component defines the FGS's view of the references. Details of how these values are obtained from other systems are encapsulated within this component.

Autopilot (page 130) The Autopilot (AP) commands the movement the control surfaces based on the pitch and roll commands generated by the FGS. This component defines the FGS's view of the AP.

Offside FGS (page 133) The FGS has two physical sides, or channels, on the left and right sides of the aircraft. These provide redundant implementations that communicate with each other over a cross-channel bus. This component defines this side of the FGS's view of the other side.

Flight Modes (page 32) The flight modes determine which modes of operation of the FGS are active and armed at any given moment. These in term determine which flight control laws are generating the commands directing the aircraft along the lateral (roll) and vertical (pitch) axes. This component encapsulates the definitions of the lateral and vertical modes and defines how they are synchronized.

Lateral Modes (page 35) The lateral modes select the control laws generating commands directing the aircraft along the lateral, or roll, axis. This component describes the specific lateral modes present in this aircraft and defines how they are synchronized.

Roll Hold (ROLL) Mode (page 37) In Roll Hold mode the FGS generates guidance commands to hold the aircraft at a fixed bank angle. Roll Hold mode is the basic

lateral mode and is always active when the modes are displayed and no other lateral mode is active.

Heading Select (HDG) Mode (page 40) In Heading Select mode, the FGS provides guidance commands to to track the *Selected Heading* displayed on the PFD.

Lateral Navigation (NAV) Mode (page 40) In Lateral Navigation mode, the FGS provides guidance commands to acquire and track lateral guidance for en route navigation and non-precision approaches. Lateral Navigation is an arming mode, i.e., after being selected, it must remained armed for a period of time before it can become active.

Lateral Approach (LAPPR) Mode (page 50) In Lateral Approach mode, the FGS provides guidance commands to acquire and track lateral guidance for precision and non-precision approaches. Lateral Approach is an arming mode, i.e., after being selected, it must remained armed for a period of time before it can become active.

Go Around (GA) Mode) (page 56) In Go Around mode the FGS generates guidance commands for the pilot after aborting an approach. The AP is never engaged during Go Around mode. Lateral Go Around mode provides guidance to maintain the reference heading.

Vertical Modes (page 60) The vertical modes select the control laws generating commands directing the aircraft along the vertical, or pitch, axis. This component describes the specific vertical modes present in this aircraft and defines how they are synchronized.

Pitch Hold (PITCH) Mode (page 62) In Pitch Hold mode the FGS generates guidance commands to hold the aircraft at a fixed pitch angle. Pitch Hold mode is the basic vertical mode and is always active when the modes are displayed and no other vertical mode is active.

Vertical Speed (VS) Mode (page 65) In Vertical Speed mode, the FGS provides pitch guidance commands to to hold the aircraft to the Vertical Speed (VS) reference.

Flight Level Change (FLC) Mode (page 69)

In Flight Level Change mode, the FGS provides guidance commands to acquire and track an Indicated Airspeed (IAS) or Mach Reference Airspeed, taking into account the need to climb or descend to bring the aircraft to the PreSelector Altitude (PSA).

Altitude Hold (ALT) Mode (page 73)

In Altitude Hold mode, the FGS provides pitch guidance commands to to acquire and track the Altitude reference, which is set to the current altitude when the mode is activated or upon a synchronization request by the flight crew.

Altitude Select (ALTSEL) Mode (page 77)

In Altitude Select mode, the FGS provides guidance commands to capture and track the Preselected Altitude. Altitude Select is an arming mode, i.e., after being selected, it must remained armed for a period of time before it can become active. Altitude Select mode is normally armed, although it is automically deselected by the activatation of

Vertical Approach, Go Around, or Altitude Hold mode. While in the armed state, the FGS monitors the aircraft closure rate towards the target altitude and determines the optimum capture point to transition to the capture state. In the capture state, the FGS generates vertical guidance commands to perform a smooth capture of the target altitude. Once it acquires the target altitude, it enters the track state, during which it generates vertical guidance commands to maintain the aircraft at the target altitude.

Vertical Approach (VAPPR) Mode (page 85)

In Vertical Approach mode, the FGS provides guidance commands to track vertical guidance for ILS precision Glideslope approaches. Vertical Approach is an arming mode, i.e., after being selected, it must remained armed for a period of time before it can become active.

Vertical Go Around (VGA) Mode (page 91)

In Go Around mode, the FGS provides guidance to the pilot after aborting an approach. The AP is never engaged during Go Around mode. Vertical Go Around mode provides guidance to maintain a fixed pitch angle.

2.2 Prioritization of Events

An issue that frequently arises in the specification of embedded systems is how to handle multiple events that occur at the same point in time. Here, we assume that an event corresponds to an instantaneous change in some input or state variable. One school argues that since an event occurs at a point in time, it is impossible for two events to occur at precisely the same moment, and simultaneous events should be prohibited in the underlying formal model. While theoretically appealing, many embedded systems are polling systems that periodically sample their inputs. It is difficult to imagine how a polling system could be built that could guarantee that no more than one input changes between any two samples. Even interrupt driven systems are ultimately implemented as digital systems where interrupts are only handled at well defined boundaries, such as the end of an instruction, and multiple interrupts can be observed at these boundaries.

From a practical standpoint, it seems that the required abstraction is to allow one or more events to occur at the same time, and to deal with this within the specification. RSML^{-e} makes the *one message assumption* that at most one message can arrive at any given point in time. However, within a message, several fields can change value, allowing more than one input event to occur at the same time. Since we did not know how the inputs to the mode logic would be implemented in a specific system, we decided to group the inputs into large messages in order to make as few assumptions as possible about the disjointness of input events.

In verifying the behavior of the mode logic, it became clear that the behavior of the system when two or more events occurred at the same time would indeed have to be carefully considered if the system was to have the correct behavior and satisfy even the most basic of safety properties (e.g., no more than one lateral mode or vertical mode active at the same time). Our approach was to define a prioritization of the input events such that only the highest priority event is seen by the main body of the specification and the lower priority events are discarded. For example,

it seems reasonable to respond to the pilot pressing the Go Around switch and ignore the copilot pressing the Flight Director switch, rather than the other way around.

This approach is reasonable for most combinations of events. However, a few events, such as the acquisition of a navigation beacon, do not need to be acted upon immediately, but should not be simply discarded. These were given the lowest possible priority, and the condition of their being true, rather than the event of their becoming true, was used to trigger the associated transitions. In this way, processing of the event might be delayed, but would always be completed as soon as no higher priority event blocked it.

Other approaches are of course feasible. For example, one could queue the input events and process them in the order received. This introduces considerable complexity into the model, and the simpler prioritization described here actually appears to be the more appropriate solution. One advantage of this approach is that it makes the solution explicit and easily reviewed.

We also attempted to localize this logic in the specification by introducing macros (predicates) of the form *When_Event_Name_Seen*. These macros are true only when the underlying event occurs and no higher priority event has occurred at the same time. By using only the *When_Event_Name_Seen* macros in the body of the specification, it is straightforward to change the prioritization of events, or even to implement a more complicated approach such as the queueing model discussed earlier.

As formal verification of the model proceeded, it became clear that it was possible to process some combinations of events in parallel. As a result we changed the prioritization from a total to a partial order, as shown in Figure 2.2. In this diagram, an event has a higher priority than all events that can be reached from it by following a downward chain of arrow and will supersede those events. If no path exists between two events, then the two events can be processed concurrently. Thus, pressing the SYNC switch supersedes all events except pressing the AP Disconnect or AP Engage switch, which can be processed in parallel with it. In like manner, pressing the HDG switch will supersede pressing the NAV switch, the Pilot Flying Switch, the Flight Director switch, and any of the capture or track conditions. However, a press of the HDG switch can be processed in parallel with any of the switches associated only with vertical modes, such as the ALT switch.

It is worth noting that this partial order could only have been verified with confidence through the use of formal analysis tools such as model checkers and theorem provers. A full description of this verification activity will be provided in a later report.

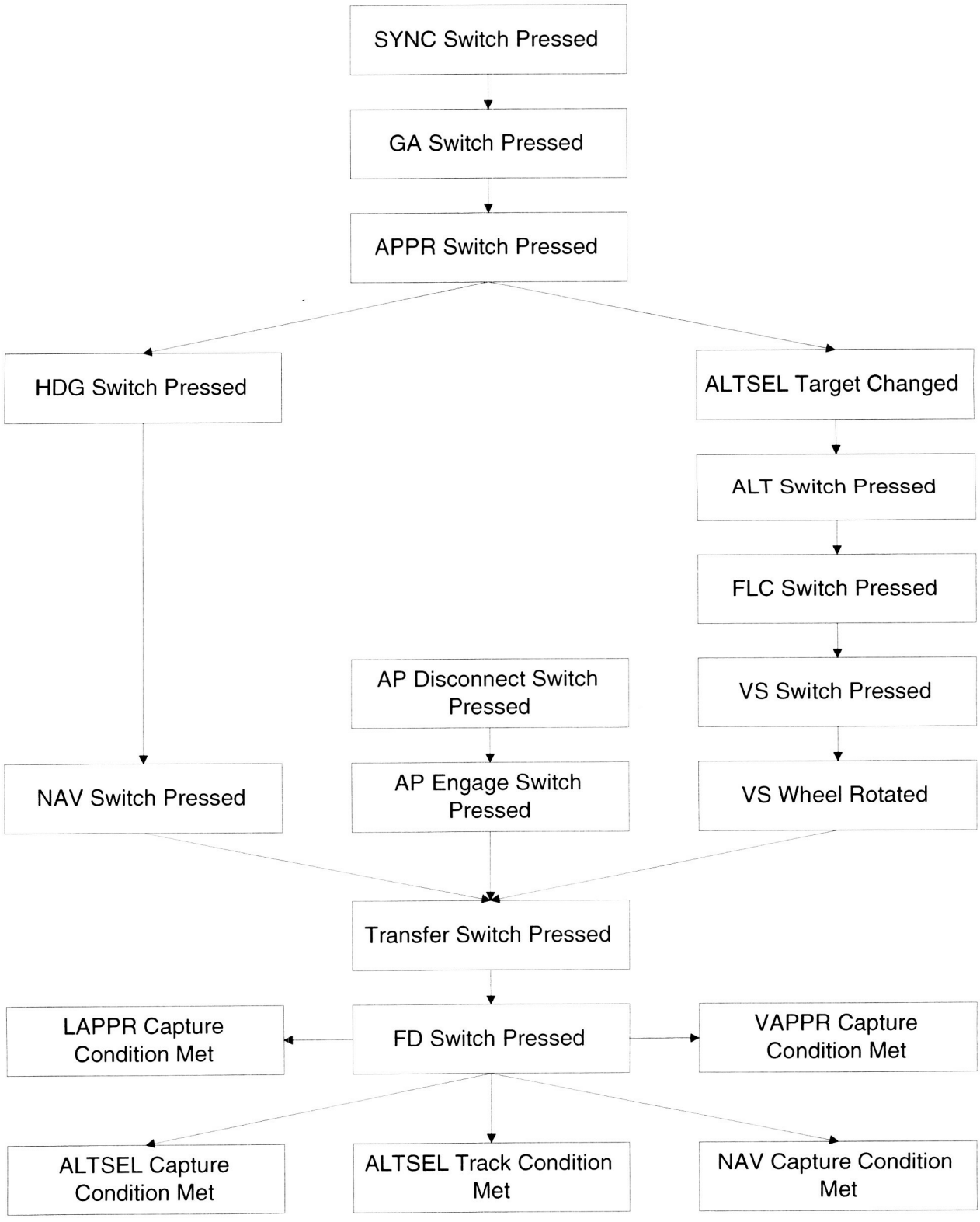


Figure 2.2: The Partial Order of Event Priorities

2.3 Basic Definitions

This section defines types and constants that are used throughout the specification.

TYPE

Side

Possible Values: LEFT, RIGHT

TYPE

On_Off

Possible Values: Off, On

TYPE

Base_State

Possible Values: Cleared, Selected

TYPE

Selected_State

Possible Values: Armed, Active

TYPE

Active_State

Possible Values: Capture, Track

CONSTANT

THIS_SIDE

Units: N/A

Value: LEFT

Purpose: This constant defines which side of the FGS (channel) is being specified. The specifications of both sides, left and right, are identical except for this constant.

2.4 Flight Director (FD)

The Flight Director (FD) displays the pitch and roll guidance commands to the pilot and copilot on the Primary Flight Display. This component defines when the Flight Director guidance cues are turned on and off.

Definitions of Values to be Imported

MACRO

When_Turn_FD_On

Condition:

		OR					
	When_FD_Switch_Pressed_Seen$_{\text{m-103}}$()	T	·	·	·	·	·
	When(AP$_{\text{v-132}}$ =Engaged)	·	T	·	·	·	·
	Overspeed_Condition$_{\text{m-128}}$()	·	·	T	·	·	·
AND	When_Lateral_Mode_Manually_Selected$_{\text{m-26}}$()	·	·	·	T	·	·
	When_Vertical_Mode_Manually_Selected$_{\text{m-27}}$()	·	·	·	·	T	·
	When_Pilot_Flying_Transfer$_{\text{m-29}}$()	·	·	·	·	·	T
	Pilot_Flying$_{\text{v-29}}$ =THIS_SIDE$_{\text{LEFT}}$	·	·	·	·	·	T
	PREVIOUS STEP(Mode_Annunciations_On$_{\text{v-33}}$)	·	·	·	·	·	T

Purpose: This event defines when the onside FD is to be turned on (i.e., displayed on the PFD).

MACRO

When_Turn_FD_Off

Condition:

AND	When_FD_Switch_Pressed_Seen$_{\text{m-103}}$()	T
	Overspeed_Condition$_{\text{m-128}}$()	F

Purpose: This event defines when the onside FD is to be turned off (i.e., removed from the PFD).

MACRO

When_Lateral_Mode_Manually_Selected

Condition:

		OR							
	PREVIOUS STEP(Mode_Annunciations_On$_{\text{v-33}}$)	F	F	F	F	·	·	·	·
	Is_This_Side_Active$_{\text{v-30}}$	·	·	·	·	T	T	T	T
AND	When_HDG_Switch_Pressed_Seen$_{\text{m-109}}$()	T	·	·	·	T	·	·	·
	When_NAV_Switch_Pressed_Seen$_{\text{m-108}}$()	·	T	·	·	·	T	·	·
	When_APPR_Switch_Pressed_Seen$_{\text{m-115}}$()	·	·	T	·	·	·	T	·
	When_GA_Switch_Pressed_Seen$_{\text{m-116}}$()	·	·	·	T	·	·	·	T

Purpose: This event defines when a lateral mode is manually selected.

MACRO

When_Vertical_Mode_Manually_Selected

Condition:

							OR						
	PREVIOUS STEP(Mode_Annunciations_On$_{v\text{-}33}$)	F	F	F	F	F	·	·	·	·	·	·	·
	Is_This_Side_Active$_{v\text{-}30}$	·	·	·	·	·	T	T	T	T	T	T	T
	When_VS_Switch_Pressed_Seen$_{m\text{-}111}$()	T	·	·	·	·	T	·	·	·	·	·	·
	When_FLC_Switch_Pressed_Seen$_{m\text{-}112}$()	·	T	·	·	·	·	T	·	·	·	·	·
	When_ALT_Switch_Pressed_Seen$_{m\text{-}113}$()	·	·	T	·	·	·	·	T	·	·	·	·
AND	When_APPR_Switch_Pressed_Seen$_{m\text{-}115}$()	·	·	·	T	·	·	·	·	T	·	·	·
AND	When_GA_Switch_Pressed_Seen$_{m\text{-}116}$()	·	·	·	·	T	·	·	·	·	T	·	·
	When_VS_Pitch_Wheel_Rotated_Seen$_{m\text{-}110}$()	·	·	·	·	·	·	·	·	·	·	T	·
	PREVIOUS STEP(Is_VS_Active$_{v\text{-}66}$)	·	·	·	·	·	·	·	·	·	·	F	·
	PREVIOUS STEP(Is_VAPPR_Active$_{v\text{-}87}$)	·	·	·	·	·	·	·	·	·	·	F	·
	PREVIOUS STEP(Overspeed$_{iv\text{-}128}$)	·	·	·	·	·	·	·	·	·	·	F	·
	When_ALTSEL_Target_Altitude_Changed_Seen$_{m\text{-}114}$()	·	·	·	·	·	·	·	·	·	·	·	T
	PREVIOUS STEP(Is_ALTSEL_Active$_{v\text{-}80}$)	·	·	·	·	·	·	·	·	·	·	·	T

Purpose: This event defines when a vertical mode is manually selected.

Definitions of Values to be Exported

STATE VARIABLE

Onside_FD_On

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= $\text{Onside\_FD}_{\text{v-28}}$ =On **if** TRUE

Purpose: Indicates if the FD Guidance cues should be displayed on the PFD.

Definitions of Values to be Encapsulated

STATE VARIABLE

Onside_FD

Parent: NONE

Type: $\{\text{Off, On}\}_{\text{ty-23}}$
Initial Value: Off
Classified as: State
Off ⟶ On **if** $\text{When\_Turn\_FD\_On}_{\text{m-25}}()$

On ⟶ Off **if** $\text{When\_Turn\_FD\_Off}_{\text{m-26}}()$

Purpose: This variable maintains the current state of the onside Flight Director.

2.5 Pilot Flying (PF)

The transfer switch on the FCP is used to determine which FGS provides mode annunciations and guidance commands to the PFDs while in dependent mode. This component defines which side is the Pilot Flying side. It also defines the notion of the "active" side of the FGS. A side of the FGS is said to be "active" if it is providing mode annunciations and guidance commands to its PFD.

Definitions of Values to be Exported

STATE VARIABLE

Pilot_Flying

Parent: NONE

Type: {LEFT, RIGHT}$_{\text{ty-23}}$
Initial Value: LEFT
Classified as: State

LEFT ⟶ RIGHT **if** When_Transfer_Switch_Pressed_Seen$_{\text{m-104}}$()

RIGHT ⟶ LEFT **if** When_Transfer_Switch_Pressed_Seen$_{\text{m-104}}$()

Purpose: This state variable maintains which side this side of the FGS believes is the Pilot Flying side.

MACRO

When_Pilot_Flying_Transfer

Condition: Changed(Pilot_Flying$_{\text{v-29}}$)

Purpose: This event occurs when the pilot flying side changes to the other side.

2.6 Independent Mode

Normally, the FGS operates in dependent mode in which the pilot flying (PF) side of the FGS provides the mode annunciations and FD guidance commands to both PFDs. While in dependent mode, the pilot not flying (PNF) side of the FGS synchronizes its modes to the pilot flying side. Under final approach conditions and during takeoff or go around modes, the FGS operates in independent mode in which each side operates independently of the other side and provides mode annunciations and FD guidance commands to its own PFD. This component defines when the FGS enters independent mode.

Definitions of Values to be Exported

STATE VARIABLE

Is_This_Side_Active

Parent: NONE

Type: BOOLEAN
Initial Value: TRUE
Classified as: State
:= TRUE **if**

		OR		
AND	PREVIOUS STEP(Independent_Mode$_{v\text{-}31}$) =Off	F	T	T
	Pilot_Flying$_{v\text{-}29}$ =THIS_SIDE$_{LEFT}$	·	T	F
	Offside_Modes$_{v\text{-}135}$ =On	·	·	F

:= FALSE **if**

AND	PREVIOUS STEP(Independent_Mode$_{v\text{-}31}$) =Off	T
	Pilot_Flying$_{v\text{-}29}$ =THIS_SIDE$_{LEFT}$	F
	Offside_Modes$_{v\text{-}135}$ =On	T

Purpose: A side of the FGS is said to be active if it is the source for guidance to the AP its FD. This occurs if the system is in independent mode, if this side of the FGS is the pilot flying side, or if the other side of the FGS is the pilot flying side but its mode annunciations are not turned on.

Definitions of Values to be Encapsulated

MACRO

Independent_Mode_Condition

Condition:

		OR	
AND	$\text{Is\_LAPPR\_Active}_{\text{v-52}}$	T	·
	$\text{Is\_VAPPR\_Active}_{\text{v-87}}$	T	·
	$\text{Is\_Offside\_LAPPR\_Active}_{\text{v-134}}$	T	·
	$\text{Is\_Offside\_VAPPR\_Active}_{\text{v-134}}$	T	·
	$\text{Is\_VGA\_Active}_{\text{v-92}}$	·	T
	$\text{Is\_Offside\_VGA\_Active}_{\text{v-135}}$	·	T

Purpose: This condition defines when the system is in independent mode. This occurs when lateral and vertical approach modes are active on both sides (channels) or vertical Go Around mode is active.

STATE VARIABLE

Independent_Mode

Parent: NONE

Type: $\{\text{Off, On}\}_{\text{ty-23}}$
Initial Value: Off
Classified as: State
:= On **if** $\text{Independent\_Mode\_Condition}_{\text{m-31}}()$

:= Off **if** **not** $\text{Independent\_Mode\_Condition}_{\text{m-31}}()$

Purpose: This variable maintains the state of whether this side is in independent mode.

Comment: This variable is introduced to make it easier to access its previous value and so that it can be displayed during simulation.

2.7 Flight Modes

The flight modes determine which modes of operation of the FGS are active and armed at any given moment. These in term determine which flight control laws are generating the commands directing the aircraft along the lateral (roll) and vertical (pitch) axes. This component describes the lateral and vertical modes and defines how they are synchronized.

Definitions of Values to be Imported

MACRO

When_Turn_Modes_On

Condition:

				OR
AND	Onside_FD$_{v\text{-}28}$ =On	T	·	·
	Offside_FD$_{v\text{-}133}$ =On	·	T	·
	Is_AP_Engaged$_{v\text{-}131}$	·	·	T

Purpose: This event defines when the flight modes are to be turned on and displayed on the PFD.

MACRO

When_Turn_Modes_Off

Condition:

AND	Onside_FD$_{v\text{-}28}$ =Off	T
	Offside_FD$_{v\text{-}133}$ =Off	T
	AP$_{v\text{-}132}$ =Disengaged	T

Purpose: This event defines when the flight modes are to be turned off and removed from the PFD.

Definitions of Values to be Exported

STATE VARIABLE

Mode_Annunciations_On

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= $\text{Modes}_{\text{v-34}}$ =On **if** TRUE

Purpose: Indicates if the mode annunications should be displayed on the PFD.

Definitions of Values to be Encapsulated

STATE VARIABLE

Modes

Parent: NONE

Type: $\{\text{Off, On}\}_{\text{ty-23}}$
Initial Value: Off
Classified as: State
:= $\text{Offside\_Modes}_{\text{v-135}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

Off ⟶ On **if**

AND	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T
	$\text{When\_Turn\_Modes\_On}_{\text{m-32}}()$	T

On ⟶ Off **if**

AND	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T
	$\text{When\_Turn\_Modes\_Off}_{\text{m-32}}()$	T

Purpose: This variable maintains the current state of whether the mode annunciations are turned on or off.

2.7.1 Lateral Modes

The lateral modes select the control laws generating commands directing the aircraft along the lateral, or roll, axis. This component describes the specific lateral modes present in this aircraft and defines how they are synchronized.

Definitions of Values to be Encapsulated

MACRO

When_Nonbasic_Lateral_Mode_Activated

Condition:

		OR			
AND	When_HDG_Activated$_{\text{m-42}}$()	T	·	·	·
	When_NAV_Activated$_{\text{m-47}}$()	·	T	·	·
	When_LAPPR_Activated$_{\text{m-53}}$()	·	·	T	·
	When_LGA_Activated$_{\text{m-58}}$()	·	·	·	T

Purpose: This event ocurrs when a new lateral mode other than the basic mode becomes active. It is used to deselect active or armed modes.

Comment: Basic mode is excluded to avoid a cyclic dependency in the definition of this macro.

MACRO

Is_No_Nonbasic_Lateral_Mode_Active

Condition:

	$\text{Is\_HDG\_Active}_{\text{v-41}}$	F
AND	$\text{Is\_NAV\_Active}_{\text{v-46}}$	F
AND	$\text{Is\_LAPPR\_Active}_{\text{v-52}}$	F
AND	$\text{Is\_LGA\_Active}_{\text{v-57}}$	F

Purpose: This condition indicates if no lateral mode except basic mode is active. It is used to trigger the activation of the basic lateral mode.

Comment: Basic mode is excluded to avoid a cyclic dependency in the definition of this macro.

2.7.1.1 Roll Hold (ROLL) Mode

In Roll Hold mode the FGS generates guidance commands to hold the aircraft at a fixed bank angle. Roll Hold mode is the basic lateral mode and is always active when the modes are displayed and no other lateral mode is active.

Definitions of Values to be Imported

MACRO

Select_ROLL

Condition:

AND	Is_No_Nonbasic_Lateral_Mode_Active$_{\text{m-36}}$()	T
	Modes$_{\text{v-34}}$ =On	T

Purpose: This event defines when Roll Hold mode is to be selected. Roll Hold mode is the basic, or default, mode and is selected whenever the mode annunciations are on and no other lateral mode is active.

Comment: To avoid cyclic dependencies, the only way to select Roll Hold mode is to deselect the active lateral mode, which will automatically activate Roll Hold.

MACRO

Deselect_ROLL

Condition:

		OR	
AND	When_Nonbasic_Lateral_Mode_Activated$_{\text{m-35}}$()	T	·
	When(Modes$_{\text{v-34}}$ =Off)	·	T

Purpose: The event defines when Roll Hold mode is to be deselected. This occurs when a new lateral mode is activated or the modes are turned off.

Definitions of Values to be Exported

STATE VARIABLE

Is_ROLL_Selected

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($ROLL_{v\text{-}39}$ =Selected) **if** TRUE

Purpose: Indicates if ROLL mode is selected.

STATE VARIABLE

Is_ROLL_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($ROLL_{v\text{-}39}$ =Selected) **if** TRUE

Purpose: Indicates if ROLL mode is selected.

Comment: Even though ROLL Selected and ROLL Active are the same value, this is defined to provide a common set of controlled variables across all modes.

Definitions of Values to be Encapsulated

STATE VARIABLE

ROLL

Parent: $\text{Modes}_{\text{v-34}}$.On

Type: $\{\text{Cleared, Selected}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_ROLL}_{\text{v-136}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED ⟶ Cleared **if**

AND	$\text{Select\_ROLL}_{\text{m-37}}()$	F
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

UNDEFINED ⟶ Selected **if**

AND	$\text{Select\_ROLL}_{\text{m-37}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Cleared ⟶ Selected **if**

AND	$\text{Select\_ROLL}_{\text{m-37}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Selected ⟶ Cleared **if**

AND	$\text{Deselect\_ROLL}_{\text{m-37}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Purpose: This variable maintains the current base state of Roll Hold mode, i.e., whether it is cleared or selected. Note that if this side of the FGS is not active, it obtains its value from the off side. The normal transitions occur only when this side is active.

2.7.1.2 Heading Select (HDG) Mode

In Heading Select mode, the FGS provides guidance commands to to track the Selected Heading displayed on the PFD.

Definitions of Values to be Imported

MACRO

Select_HDG

Condition: When_HDG_Switch_Pressed_Seen$_{\text{m-109}}$()

Purpose: This event defines when Heading Select mode is to be selected.

MACRO

Deselect_HDG

Condition:

		OR			
	When_HDG_Switch_Pressed_Seen$_{\text{m-109}}$()	T	·	·	·
A	When_Nonbasic_Lateral_Mode_Activated$_{\text{m-35}}$()	·	T	·	·
N	When_Pilot_Flying_Transfer$_{\text{m-29}}$()	·	·	T	·
D	When(Modes$_{\text{v-34}}$ =Off)	·	·	·	T

Purpose: This event defines when Heading Select mode is to be deselected.

Definitions of Values to be Exported

STATE VARIABLE

Is_HDG_Selected

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($HDG_{v\text{-}43}$ =Selected) **if** TRUE

Purpose: Indicates if HDG mode is selected.

STATE VARIABLE

Is_HDG_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($HDG_{v\text{-}43}$ =Selected) **if** TRUE

Purpose: Indicates if HDG mode is active.

Comment: Even though HDG Selected and HDG Active are the same value, this is defined to provide a common set of controlled variables across all modes.

MACRO

When_HDG_Activated

Condition:

AND	Select_HDG$_{m\text{-}40}$()	T
	PREVIOUS STEP(HDG$_{v\text{-}43}$) =Selected	F
	Is_This_Side_Active$_{v\text{-}30}$	T

Purpose: This signal occurs when Heading Select mode is activated.

Comment: This event is defined this way to avoid circular dependencies. It would be preferable to define it as When(HDG = Selected).

Definitions of Values to be Encapsulated

STATE VARIABLE

HDG

Parent: $\text{Modes}_{\text{v-34}}$.On

Type: $\{\text{Cleared, Selected}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
Purpose: This variable maintains the current base state of Heading Select mode, i.e., whether it is cleared or selected. Note that if this side of the FGS is not active, it obtains its value from the off side. The normal transitions occur only when this side is active.

:= $\text{Offside\_HDG}_{\text{v-136}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED ⟶ Cleared **if**

AND	$\text{Select\_HDG}_{\text{m-40}}()$	F
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

UNDEFINED ⟶ Selected **if**

AND	$\text{Select\_HDG}_{\text{m-40}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Cleared ⟶ Selected **if**

AND	$\text{Select\_HDG}_{\text{m-40}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Selected ⟶ Cleared **if**

AND	$\text{Deselect\_HDG}_{\text{m-40}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

2.7.1.3 Navigation (NAV) Mode

In Lateral Navigation mode, the FGS provides guidance commands to acquire and track lateral guidance for en route navigation and non-precision approaches. Lateral Navigation is an arming mode, i.e., after being selected, it must remained armed for a period of time before it can become active.

Definitions of Values to be Imported

MACRO

Select_NAV

Condition: When_NAV_Switch_Pressed_Seen$_{\text{m-108}}$()

Purpose: This event defines when Lateral Navigation mode is to be selected, i.e., to become armed.

MACRO

Activate_NAV

Condition: When_NAV_Track_Cond_Met_Seen$_{\text{m-95}}$()

Purpose: This event defines when Lateral Navigation mode is to transition from the armed to the active state.

MACRO

Deselect_NAV

Condition:

		OR				
	When_NAV_Switch_Pressed_Seen$_{m\text{-}108}$()	T	·	·	·	·
A	When_Selected_Nav_Source_Changed$_{v\text{-}127}$	·	T	·	·	·
N	When_Selected_Nav_Frequency_Changed$_{v\text{-}127}$	·	·	T	·	·
D	When_Pilot_Flying_Transfer$_{m\text{-}29}$()	·	·	·	T	·
	When(Modes$_{v\text{-}34}$ =Off)	·	·	·	·	T

Purpose: This event defines when Lateral Navigation mode is to be deselected, i.e., to be cleared regardless of whether it is armed or active.

MACRO

Dearm_NAV

Condition: When_LAPPR_Armed$_{m\text{-}53}$()

Purpose: This event defines when Lateral Navigation mode is to be cleared if it is armed, but to remain active if it is active.

MACRO

Deactivate_NAV

Condition: When_Nonbasic_Lateral_Mode_Activated$_{m\text{-}35}$()

Purpose: This event defines when Lateral Navigation mode is to be cleared if it is active, but to remain armed if it is armed.

Definitions of Values to be Exported

STATE VARIABLE

Is_NAV_Selected

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($NAV_{v\text{-}48}$ =Selected) **if** TRUE

Purpose: Indicates if NAV mode is selected.

STATE VARIABLE

Is_NAV_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($NAV\_Selected_{v\text{-}49}$ =Active) **if** TRUE

Purpose: Indicates if NAV mode is active.

MACRO

When_NAV_Armed

Condition:

AND	Select_NAV$_{m\text{-}44}$()	T
	PREVIOUS STEP(NAV$_{v\text{-}48}$) =Cleared	T
	Is_This_Side_Active$_{v\text{-}30}$	T

Purpose: This signal occurs when Lateral NAV mode is armed.

Comment: This event is defined this way to avoid circular dependencies. It would be preferable to define it as When(NAV_Selected = Armed).

MACRO

When_NAV_Activated

Condition:

AND	Activate_NAV$_{m\text{-}44}$()	T
	PREVIOUS STEP(NAV_Selected$_{v\text{-}49}$) =Armed	T
	Is_This_Side_Active$_{v\text{-}30}$	T

Purpose: This signal occurs when Lateral Navigation mode is activated.

Comment: This event is defined this way to avoid circular dependencies. It would be preferable to define it as When(NAV_Selected = Active).

Definitions of Values to be Encapsulated

STATE VARIABLE

NAV

Parent: $\text{Modes}_{\text{v-34}}$.On

Type: $\{\text{Cleared, Selected}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_NAV}_{\text{v-137}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED ⟶ Cleared **if**

AND	$\text{Select\_NAV}_{\text{m-44}}()$	F
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

UNDEFINED ⟶ Selected **if**

AND	$\text{Select\_NAV}_{\text{m-44}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Cleared ⟶ Selected **if**

AND	$\text{Select\_NAV}_{\text{m-44}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Selected ⟶ Cleared **if**

			OR	
AND	$\text{Deselect\_NAV}_{\text{m-45}}()$	T	·	·
	$\text{Dearm\_NAV}_{\text{m-45}}()$	·	T	·
	PREVIOUS STEP($\text{NAV\_Selected}_{\text{v-49}}$) =Armed	·	T	·
	$\text{Deactivate\_NAV}_{\text{m-45}}()$	·	·	T
	PREVIOUS STEP($\text{NAV\_Selected}_{\text{v-49}}$) =Active	·	·	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T	T	T

Purpose: This variable maintains the current base state of Lateral Navigation mode, i.e., whether it is cleared or selected. Note that if this side of the FGS is not active, it obtains its value from the off side. The normal transitions occur only when this side is active.

STATE VARIABLE

NAV_Selected

Parent: $\text{Modes}_{\text{v-34}}.\text{On} \triangleright \text{NAV}_{\text{v-48}}.\text{Selected}$

Type: $\{\text{Armed, Active}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_NAV\_Selected}_{\text{v-137}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED ⟶ Armed **if** TRUE

Armed ⟶ Active **if**

AND	$\text{Activate\_NAV}_{\text{m-44}}()$	T
AND	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Purpose: This variable maintains the current state of Lateral Navigation mode when it is selected, i.e., whether it is armed or active. Note that if this side of the FGS is not active, it obtains its value from the off side. The normal transitions occur only when this side is active.

2.7.1.4 Lateral Approach (LAPPR) Mode

In Lateral Approach mode, the FGS provides guidance commands to acquire and track lateral guidance for precision and non-precision approaches. Lateral Approach is an arming mode, i.e., after being selected, it must remained armed for a period of time before it can become active.

Definitions of Values to be Imported

MACRO

Select_LAPPR

Condition: When_APPR_Switch_Pressed_Seen$_{\text{m-115}}$()

Purpose: This event defines when Lateral Approach mode is to be selected, i.e., to become armed.

MACRO

Activate_LAPPR

Condition: When_LAPPR_Track_Cond_Met_Seen$_{\text{m-98}}$()

Purpose: This event defines when Lateral Approach mode is to transition from the armed to the active state.

MACRO

Deselect_LAPPR

Condition:

		OR				
	When_APPR_Switch_Pressed_Seen$_{\text{m-115}}$()	T	·	·	·	·
A	When_Selected_Nav_Source_Changed$_{\text{v-127}}$	·	T	·	·	·
N	When_Selected_Nav_Frequency_Changed$_{\text{v-127}}$	·	·	T	·	·
D	When_Pilot_Flying_Transfer$_{\text{m-29}}$()	·	·	·	T	·
	When(Modes$_{\text{v-34}}$ =Off)	·	·	·	·	T

Purpose: This event defines when Lateral Approach mode is to be deselected, i.e., to be cleared regardless of whether it is armed or active.

MACRO

Dearm_LAPPR

Condition: When_NAV_Armed$_{\text{m-47}}$()

Purpose: This event defines when Lateral Approach mode is to be cleared if it is armed, but to remain active if it is active.

MACRO

Deactivate_LAPPR

Condition: When_Nonbasic_Lateral_Mode_Activated$_{\text{m-35}}$()

Purpose: This event defines when Lateral Approach mode is to be cleared if it is active, but to remain armed if it is armed.

Definitions of Values to be Exported

STATE VARIABLE

Is_LAPPR_Selected

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($LAPPR_{v\text{-}54}$ =Selected) **if** TRUE

Purpose: Indicates if LAPPR mode is selected.

STATE VARIABLE

Is_LAPPR_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($LAPPR\_Selected_{v\text{-}55}$ =Active) **if** TRUE

Purpose: Indicates if LAPPR mode is active.

MACRO

When_LAPPR_Armed

Condition:

AND	Select_LAPPR$_{m\text{-}50}$()	T
	PREVIOUS STEP(LAPPR$_{v\text{-}54}$) =Cleared	T
	Is_This_Side_Active$_{v\text{-}30}$	T

Purpose: This signal occurs when Lateral Approach mode is armed.

Comment: This event is defined this way to avoid circular dependencies. It would be preferable to define it as When(LAPPR_Selected = Armed).

MACRO

When_LAPPR_Activated

Condition:

AND	Activate_LAPPR$_{m\text{-}50}$()	T
	PREVIOUS STEP(LAPPR_Selected$_{v\text{-}55}$) =Armed	T
	Is_This_Side_Active$_{v\text{-}30}$	T

Purpose: This signal occurs when Lateral Approach mode is activated.

Comment: This event is defined this way to avoid circular dependencies. It would be preferable to define it as When(LAPPR_Selected = Active).

Definitions of Values to be Encapsulated

STATE VARIABLE

LAPPR

Parent: $\text{Modes}_{\text{v-34}}$.On

Type: $\{\text{Cleared, Selected}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_LAPPR}_{\text{v-138}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED ⟶ Cleared **if**

AND	$\text{Select\_LAPPR}_{\text{m-50}}()$	F
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

UNDEFINED ⟶ Selected **if**

AND	$\text{Select\_LAPPR}_{\text{m-50}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Cleared ⟶ Selected **if**

AND	$\text{Select\_LAPPR}_{\text{m-50}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Selected ⟶ Cleared **if**

			OR	
AND	$\text{Deselect\_LAPPR}_{\text{m-51}}()$	T	·	·
	$\text{Dearm\_LAPPR}_{\text{m-51}}()$	·	T	·
	PREVIOUS STEP($\text{LAPPR\_Selected}_{\text{v-55}}$) =Armed	·	T	·
	$\text{Deactivate\_LAPPR}_{\text{m-51}}()$	·	·	T
	PREVIOUS STEP($\text{LAPPR\_Selected}_{\text{v-55}}$) =Active	·	·	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T	T	T

Purpose: This variable maintains the current base state of Lateral Approach mode, i.e., whether it is cleared or selected. Note that if this side of the FGS is not active, it obtains its value from the off side. The normal transitions occur only when this side is active.

STATE VARIABLE

LAPPR_Selected

Parent: $\text{Modes}_{\text{v-34}}.\text{On} \triangleright \text{LAPPR}_{\text{v-54}}.\text{Selected}$

Type: $\{\text{Armed, Active}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_LAPPR\_Selected}_{\text{v-138}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED $\longrightarrow$ Armed **if** TRUE

Armed $\longrightarrow$ Active **if**

AND	$\text{Activate\_LAPPR}_{\text{m-50}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Purpose: This variable maintains the current state of Lateral Approach mode when it is selected, i.e., whether it is armed or active. Note that if this side of the FGS is not active, it obtains its value from the off side. The normal transitions occur only when this side is active.

2.7.1.5 Lateral Go Around (LGA) Mode

In Go Around mode the FGS generates guidance commands for the pilot after aborting an approach. The AP is never engaged during Go Around mode. Lateral Go Around mode provides guidance to maintain the reference heading.

Definitions of Values to be Imported

MACRO

Select_LGA

Condition:

AND	When_GA_Switch_Pressed_Seen$_{\text{m-116}}$()	T
	Overspeed_Condition$_{\text{m-128}}$()	F

Purpose: This event defines when Lateral Go Around mode is to be selected.

MACRO

Deselect_LGA

Condition:

		OR					
AND	When(AP$_{\text{v-132}}$ =Engaged)	T	·	·	·	·	·
	When_Nonbasic_Lateral_Mode_Activated$_{\text{m-35}}$()	·	T	·	·	·	·
	When_Nonbasic_Vertical_Mode_Activated$_{\text{m-60}}$()	·	·	T	·	·	·
	When_SYNC_Switch_Pressed_Seen$_{\text{m-117}}$()	·	·	·	T	·	·
	When_Pilot_Flying_Transfer$_{\text{m-29}}$()	·	·	·	·	T	·
	When(Modes$_{\text{v-34}}$ =Off)	·	·	·	·	·	T

Purpose: This event defines when Lateral Go Around mode is to be deselected.

Definitions of Values to be Exported

STATE VARIABLE

Is_LGA_Selected

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($LGA_{V\text{-}59}$ =Selected) **if** TRUE

Purpose: Indicates if LGA mode is selected.

STATE VARIABLE

Is_LGA_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($LGA_{V\text{-}59}$ =Selected) **if** TRUE

Purpose: Indicates if LGA mode is active.

Comment: Even though LGA Selected and LGA Active are the same value, this is defined to provide a common set of controlled variables across all modes.

MACRO

When_LGA_Activated

Condition:

AND	Select_LGA$_{m\text{-}56}$()	T
	PREVIOUS STEP(LGA$_{v\text{-}59}$) =Selected	F
	Is_This_Side_Active$_{v\text{-}30}$	T

Purpose: This signal occurs when Lateral Go Around mode is activated.

Comment: This event is defined this way to avoid circular dependencies. It would be preferable to define it as When(LGA = Selected).

Definitions of Values to be Encapsulated

STATE VARIABLE

LGA

Parent: $\text{Modes}_{\text{v-34}}$.On

Type: $\{\text{Cleared, Selected}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_LGA}_{\text{v-139}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED ⟶ Cleared **if**

AND	$\text{Select\_LGA}_{\text{m-56}}()$	F
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

UNDEFINED ⟶ Selected **if**

AND	$\text{Select\_LGA}_{\text{m-56}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Cleared ⟶ Selected **if**

AND	$\text{Select\_LGA}_{\text{m-56}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Selected ⟶ Cleared **if**

AND	$\text{Deselect\_LGA}_{\text{m-56}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Purpose: This variable maintains the current state of Lateral Go Around mode. Note that if this side is not active, it obtains its value from the off side. The normal transitions for Lateral Go Around mode occur only when this side is active.

2.7.2 Vertical Modes

The vertical modes select the control laws generating commands directing the aircraft along the vertical, or pitch, axis. This component describes the specific vertical modes present in this aircraft and defines how they are synchronized.

Definitions of Values to be Encapsulated

MACRO

When_Nonbasic_Vertical_Mode_Activated

Condition:

		OR					
	When_VS_Activated$_{\text{m-67}}$()	T	·	·	·	·	·
	When_FLC_Activated$_{\text{m-71}}$()	·	T	·	·	·	·
AND	When_ALT_Activated$_{\text{m-75}}$()	·	·	T	·	·	·
	When_ALTSEL_Activated$_{\text{m-81}}$()	·	·	·	T	·	·
	When_VAPPR_Activated$_{\text{m-88}}$()	·	·	·	·	T	·
	When_VGA_Activated$_{\text{m-93}}$()	·	·	·	·	·	T

Purpose: This event indicates when a new vertical mode other than the basic mode becomes active. It is used to deselect active or armed modes.

Comment: Basic mode is excluded to avoid a cyclic dependency in the definition of this macro.

MACRO

Is_No_Nonbasic_Vertical_Mode_Active

Condition:

	$Is\_VS\_Active_{v\text{-}66}$	F
	$Is\_FLC\_Active_{v\text{-}71}$	F
AND	$Is\_ALT\_Active_{v\text{-}74}$	F
	$Is\_ALTSEL\_Active_{v\text{-}80}$	F
	$Is\_VAPPR\_Active_{v\text{-}87}$	F
	$Is\_VGA\_Active_{v\text{-}92}$	F

Purpose: This condition indicates if no vertical mode except basic mode is active. It is used to trigger the activation of the basic lateral mode.

Comment: Basic mode is excluded to avoid a cyclic dependency in the definition of this macro.

2.7.2.1 Pitch Hold (PITCH) Mode

In Pitch Hold mode the FGS generates guidance commands to hold the aircraft at a fixed pitch angle. Pitch Hold mode is the basic vertical mode and is always active when the modes are displayed and no other vertical mode is active.

Definitions of Values to be Imported

MACRO

Select_PITCH

Condition:

A	Is_No_Nonbasic_Vertical_Mode_Active$_{\text{m-61}}()$	T
N		
D	$\text{Modes}_{\text{v-34}}$ =On	T

Purpose: Pitch Hold mode is the basic, or default, mode and is selected whenever the mode annunciations are on and no other vertical mode is active.

Comment: To avoid cyclic dependencies, the only way to select Pitch Hold mode is to deselect the active vertical mode, which will automatically activate Pitch Hold mode.

MACRO

Deselect_PITCH

Condition:

			OR
A	When_Nonbasic_Vertical_Mode_Activated$_{\text{m-60}}()$	T	·
N			
D	When($\text{Modes}_{\text{v-34}}$ =Off)	·	T

Purpose: Pitch Hold mode is deselected when: a new vertical mode is activated or the modes are turned off.

Definitions of Values to be Exported

STATE VARIABLE

Is_PITCH_Selected

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($PITCH_{v\text{-}64}$ =Selected) **if** TRUE

Purpose: Indicates if PITCH mode is selected.

STATE VARIABLE

Is_PITCH_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($PITCH_{v\text{-}64}$ =Selected) **if** TRUE

Purpose: Indicates if PITCH mode is active.

Comment: Even though PITCH Selected and PITCH Active are the same value, this is defined to provide a common set of controlled variables across all modes.

Definitions of Values to be Encapsulated

STATE VARIABLE

PITCH

Parent: $\text{Modes}_{\text{v-34}}$.On

Type: $\{\text{Cleared, Selected}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_PITCH}_{\text{v-139}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED ⟶ Cleared **if**

AND	$\text{Select\_PITCH}_{\text{m-62}}()$	F
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

UNDEFINED ⟶ Selected **if**

AND	$\text{Select\_PITCH}_{\text{m-62}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Cleared ⟶ Selected **if**

AND	$\text{Select\_PITCH}_{\text{m-62}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Selected ⟶ Cleared **if**

AND	$\text{Deselect\_PITCH}_{\text{m-62}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Purpose: This variable maintains the current base state of Pitch Hold mode, i.e., whether it is cleared or selected. Note that if this side of the FGS is not active, it obtains its value from the off side. The normal transitions occur only when this side is active.

2.7.2.2 Vertical Speed (VS) Mode

In Vertical Speed mode, the FGS provides pitch guidance commands to to hold the aircraft to the Vertical Speed (VS) reference.

Definitions of Values to be Imported

MACRO

Select_VS

Condition:

AND	When_VS_Switch_Pressed_Seen$_{m\text{-}111}$()	T
	Overspeed_Condition$_{m\text{-}128}$()	F
	PREVIOUS STEP(Is_VAPPR_Active$_{v\text{-}87}$)	F

Purpose: This event defines when Vertical Speed mode is to be selected.

MACRO

Deselect_VS

Condition:

				OR	
AND	When_VS_Switch_Pressed_Seen$_{m\text{-}111}$()	T	·	·	·
	When_Nonbasic_Vertical_Mode_Activated$_{m\text{-}60}$()	·	T	·	·
	When_Pilot_Flying_Transfer$_{m\text{-}29}$()	·	·	T	·
	When(Modes$_{v\text{-}34}$ =Off)	·	·	·	T

Purpose: This event defines when Vertical Speed mode is to be deselected.

Definitions of Values to be Exported

STATE VARIABLE

Is_VS_Selected

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($VS_{v\text{-}68}$ =Selected) **if** TRUE

Purpose: Indicates if VS mode is selected.

STATE VARIABLE

Is_VS_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($VS_{v\text{-}68}$ =Selected) **if** TRUE

Purpose: Indicates if VS mode is active.

Comment: Even though VS Selected and VS Active are the same value, this is defined to provide a common set of controlled variables across all modes.

MACRO

When_VS_Activated

Condition:

AND	Select_VS$_{m\text{-}65}$()	T
	PREVIOUS STEP(VS$_{v\text{-}68}$) =Selected	F
	Is_This_Side_Active$_{v\text{-}30}$	T

Purpose: This signal occurs when Vertical Speed mode is activated.

Comment: This event is defined this way to avoid circular dependencies. It would be preferable to define it as When(VS = Selected).

Definitions of Values to be Encapsulated

STATE VARIABLE

VS

Parent: $\text{Modes}_{\text{v-34}}$.On

Type: $\{\text{Cleared, Selected}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_VS}_{\text{v-140}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED ⟶ Cleared **if**

AND	$\text{Select\_VS}_{\text{m-65}}()$	F
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

UNDEFINED ⟶ Selected **if**

AND	$\text{Select\_VS}_{\text{m-65}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Cleared ⟶ Selected **if**

AND	$\text{Select\_VS}_{\text{m-65}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Selected ⟶ Cleared **if**

AND	$\text{Deselect\_VS}_{\text{m-65}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Purpose: This variable maintains the current base state of Vertical Speed mode, i.e., whether it is cleared or selected. Note that if this side of the FGS is not active, it obtains its value from the off side. The normal transitions occur only when this side is active.

2.7.2.3 Flight Level Change (FLC) Mode

In Flight Level Change mode, the FGS provides guidance commands to acquire and track an Indicated Airspeed (IAS) or Mach Reference Airspeed, taking into account the need to climb or descend to bring the aircraft to the PreSelector Altitude (PSA).

Definitions of Values to be Imported

MACRO

Select_FLC

Condition:

		OR	
	When_FLC_Switch_Pressed_Seen$_{m\text{-}112}$()	T	·
	PREVIOUS STEP(Is_VAPPR_Active$_{v\text{-}87}$)	F	·
AND	Overspeed_Condition$_{m\text{-}128}$()	·	T
AND	PREVIOUS STEP(Is_ALT_Active$_{v\text{-}74}$)	·	F
AND	When_ALT_Activated$_{m\text{-}75}$()	·	F
AND	PREVIOUS STEP(Is_ALTSEL_Active$_{v\text{-}80}$)	·	F
AND	When_ALTSEL_Activated$_{m\text{-}81}$()	·	F

Purpose: This event defines when Flight Level Change mode is to be selected.

MACRO

Deselect_FLC

Condition:

		OR				
	When_FLC_Switch_Pressed_Seen$_{\text{m-112}}$()	T	·	·	·	·
	When_VS_Pitch_Wheel_Rotated_Seen$_{\text{m-110}}$()	·	T	·	·	·
AND	Overspeed_Condition$_{\text{m-128}}$()	F	F	·	·	·
	When_Nonbasic_Vertical_Mode_Activated$_{\text{m-60}}$()	·	·	T	·	·
	When_Pilot_Flying_Transfer$_{\text{m-29}}$()	·	·	·	T	·
	When(Modes$_{\text{v-34}}$ =Off)	·	·	·	·	T

Purpose: This event defines when Flight Level Change mode is to be deselected.

Definitions of Values to be Exported

STATE VARIABLE

Is_FLC_Selected

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= (FLC$_{\text{v-72}}$ =Selected) **if** TRUE

Purpose: Indicates if FLC mode is selected.

STATE VARIABLE

Is_FLC_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($FLC_{v\text{-}72}$ =Selected) **if** TRUE

Purpose: Indicates if FLC mode is active.

Comment: Even though FLC Selected and FLC Active are the same value, this is defined to provide a common set of controlled variables across all modes.

MACRO

When_FLC_Activated

Condition:

AND	$Select\_FLC_{m\text{-}69}()$	T
	PREVIOUS STEP($FLC_{v\text{-}72}$) =Selected	F
	$Is\_This\_Side\_Active_{v\text{-}30}$	T

Purpose: This signal occurs when Flight Level Change mode is activated.

Comment: This event is defined this way to avoid circular dependencies. It would be preferable to define it as When(FLC = Selected).

Definitions of Values to be Encapsulated

STATE VARIABLE

FLC

Parent: $\text{Modes}_{\text{v-34}}$.On

Type: $\{\text{Cleared, Selected}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_FLC}_{\text{v-140}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED ⟶ Cleared **if**

AND	$\text{Select\_FLC}_{\text{m-69}}()$	F
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

UNDEFINED ⟶ Selected **if**

AND	$\text{Select\_FLC}_{\text{m-69}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Cleared ⟶ Selected **if**

AND	$\text{Select\_FLC}_{\text{m-69}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Selected ⟶ Cleared **if**

AND	$\text{Deselect\_FLC}_{\text{m-70}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Purpose: This variable maintains the current state of Flight Level Change mode. Note that if this side is not active, it obtains its value from the off side. The normal transitions for Heading Select mode occur only when this side is active.

2.7.2.4 Altitude Hold (ALT) Mode

In Altitude Hold mode, the FGS provides pitch guidance commands to to acquire and track the Altitude reference, which is set to the current altitude when the mode is activated or upon a synchronization request by the flight crew.

Definitions of Values to be Imported

MACRO

Select_ALT

Condition:

			OR
AND	When_ALT_Switch_Pressed_Seen$_{m\text{-}113}$()	T	·
	PREVIOUS STEP(Is_VAPPR_Active$_{v\text{-}87}$)	F	F
	When_ALTSEL_Target_Altitude_Changed_Seen$_{m\text{-}114}$()	·	T
	PREVIOUS STEP(Is_ALTSEL_Track$_{v\text{-}80}$)	·	T

Purpose: This event defines when Altitude Hold mode is to be selected.

MACRO

Deselect_ALT

Condition:

						OR
AND	When_ALT_Switch_Pressed_Seen$_{m\text{-}113}$()	T	·	·	·	·
	When_VS_Pitch_Wheel_Rotated_Seen$_{m\text{-}110}$()	·	T	·	·	·
	When_Nonbasic_Vertical_Mode_Activated$_{m\text{-}60}$()	·	·	T	·	·
	When_Pilot_Flying_Transfer$_{m\text{-}29}$()	·	·	·	T	·
	When(Modes$_{v\text{-}34}$ =Off)	·	·	·	·	T

Purpose: This event defines when Altitude Hold mode is to be deselected.

Definitions of Values to be Exported

STATE VARIABLE

Is_ALT_Selected

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($ALT_{v\text{-}76}$ =Selected) **if** TRUE

Purpose: Indicates if ALT mode is selected.

STATE VARIABLE

Is_ALT_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($ALT_{v\text{-}76}$ =Selected) **if** TRUE

Purpose: Indicates if ALT mode is active.

Comment: Even though ALT Selected and ALT Active are the same value, this is defined to provide a common set of controlled variables across all modes.

MACRO

When_ALT_Activated

Condition:

AND	Select_ALT$_{m\text{-}73}$()	T
	PREVIOUS STEP(ALT$_{v\text{-}76}$) =Selected	F
	Is_This_Side_Active$_{v\text{-}30}$	T

Purpose: This signal occurs when Altitude Hold mode is activated.

Comment: This event is defined this way to avoid circular dependencies. It would be preferable to define it as When(ALT = Selected).

Definitions of Values to be Encapsulated

STATE VARIABLE

ALT

Parent: $\text{Modes}_{\text{v-34}}$.On

Type: $\{\text{Cleared, Selected}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_ALT}_{\text{v-141}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED $\longrightarrow$ Cleared **if**

AND	$\text{Select\_ALT}_{\text{m-73}}()$	F
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

UNDEFINED $\longrightarrow$ Selected **if**

AND	$\text{Select\_ALT}_{\text{m-73}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Cleared $\longrightarrow$ Selected **if**

AND	$\text{Select\_ALT}_{\text{m-73}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Selected $\longrightarrow$ Cleared **if**

AND	$\text{Deselect\_ALT}_{\text{m-73}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Purpose: This variable maintains the current base state of Altitude Hold mode, i.e., whether it is cleared or selected. Note that if this side of the FGS is not active, it obtains its value from the off side. The normal transitions occur only when this side is active.

2.7.2.5 Altitude Select (ALTSEL) Mode

In Altitude Select mode, the FGS provides guidance commands to capture and track the Preselected Altitude. Altitude Select is an arming mode, i.e., after being selected, it must remained armed for a period of time before it can become active. Altitude Select mode is normally armed, although it is automically deselected by the activatation of Vertical Approach, Go Around, or Altitude Hold mode. While in the armed state, the FGS monitors the aircraft closure rate towards the target altitude and determines the optimum capture point to transition to the capture state. In the capture state, the FGS generates vertical guidance commands to perform a smooth capture of the target altitude. Once it acquires the target altitude, it enters the track state, during which it generates vertical guidance commands to maintain the aircraft at the target altitude.

Definitions of Values to be Imported

MACRO

Select_ALTSEL

Condition:

AND	$Is\_VAPPR\_Active_{v\text{-}87}$	F
AND	$Is\_VGA\_Active_{v\text{-}92}$	F
AND	$Is\_ALT\_Active_{v\text{-}74}$	F
	$Modes_{v\text{-}34}$ =On	T

Purpose: This event defines when Altitude Select mode is to be selected, i.e., to become armed. Altitude Select mode is armed or active except when Vertical Approach, Go Around, or Altitude Hold are active.

MACRO

Capture_ALTSEL

Condition: When_ALTSEL_Capture_Cond_Met_Seen$_{\text{m-97}}$()

Purpose: This event defines when Altitude Select mode is to transition from the armed to the capture state.

MACRO

Track_ALTSEL

Condition: When_ALTSEL_Track_Cond_Met_Seen$_{\text{m-96}}$()

Purpose: This event defines when Altitude Select mode is to transition from the capture to the track state.

MACRO

Deselect_ALTSEL

Condition:

		OR			
AND	Is_VAPPR_Active$_{\text{v-87}}$	T	·	·	·
	Is_VGA_Active$_{\text{v-92}}$	·	T	·	·
	Is_ALT_Active$_{\text{v-74}}$	·	·	T	·
	When(Modes$_{\text{v-34}}$ =Off)	·	·	·	T

Purpose: This event defines when Altitude Select mode is to be deselected, i.e., to be cleared regardless of whether it is armed or active. Altitude Select mode is armed or active except when Vertical Approach, Go Around, or Altitude Hold are active.

MACRO

Deactivate_ALTSEL

Condition:

				OR	
A	When_ALTSEL_Target_Altitude_Changed_Seen$_{\text{m-114}}$()	T	·	·	·
N	When_VS_Pitch_Wheel_Rotated_Seen$_{\text{m-110}}$()	·	T	·	·
D	When_Pilot_Flying_Transfer$_{\text{m-29}}$()	·	·	T	·
	When_Nonbasic_Vertical_Mode_Activated$_{\text{m-60}}$()	·	·	·	T

Purpose: This event defines when Altitude Select mode is to be cleared if it is active, but to remain armed if it is armed.

Definitions of Values to be Exported

STATE VARIABLE

Is_ALTSEL_Selected

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= (ALTSEL$_{\text{v-82}}$ =Selected) **if** TRUE

Purpose: Indicates if ALTSEL mode is selected.

STATE VARIABLE

Is_ALTSEL_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= (ALTSEL_Selected$_{v\text{-}83}$ =Active) **if** TRUE

Purpose: Indicates if ALTSEL mode is active.

STATE VARIABLE

Is_ALTSEL_Track

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= (ALTSEL_Active$_{v\text{-}84}$ =Track) **if** TRUE

Purpose: Indicates if ALTSEL mode is tracking.

MACRO

When_ALTSEL_Activated

Condition:

AND	Capture_ALTSEL$_{m\text{-}78}$()	T
	PREVIOUS STEP(ALTSEL_Selected$_{v\text{-}83}$) =Armed	T
	Is_This_Side_Active$_{v\text{-}30}$	T

Purpose: This signal occurs when Altitude Select mode is activated, i.e., it enters the capture state.

Comment: This event is defined this way to avoid circular dependencies. It would be preferable to define it as When(ALTSEL_Selected = Active).

Definitions of Values to be Encapsulated

STATE VARIABLE

ALTSEL

Parent: $\text{Modes}_{\text{v-34}}$.On

Type: $\{\text{Cleared, Selected}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_ALTSEL}_{\text{v-141}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED ⟶ Cleared **if**

AND	$\text{Select\_ALTSEL}_{\text{m-77}}()$	F
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

UNDEFINED ⟶ Selected **if**

AND	$\text{Select\_ALTSEL}_{\text{m-77}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Cleared ⟶ Selected **if**

AND	$\text{Select\_ALTSEL}_{\text{m-77}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Selected ⟶ Cleared **if**

AND	$\text{Deselect\_ALTSEL}_{\text{m-78}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Purpose: This variable maintains the current base state of Altitude Select mode, i.e., whether it is cleared or selected. Note that if this side of the FGS is not active, it obtains its value from the off side. The normal transitions occur only when this side is active.

STATE VARIABLE

ALTSEL_Selected

Parent: $\text{Modes}_{\text{v-34}}$.On ▷ $\text{ALTSEL}_{\text{v-82}}$.Selected

Type: $\{\text{Armed, Active}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_ALTSEL\_Selected}_{\text{v-142}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

[UNDEFINED] ⟶ [Armed] **if** TRUE

[Armed] ⟶ [Active] **if**

AND	$\text{Capture\_ALTSEL}_{\text{m-78}}()$	T
AND	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

[Active] ⟶ [Armed] **if**

AND	$\text{Deactivate\_ALTSEL}_{\text{m-79}}()$	T
AND	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Purpose: This variable maintains the current state of Altitude Select mode when it is selected, i.e., whether it is armed or active. Note that if this side of the FGS is not active, it obtains its value from the off side. The normal transitions occur only when this side is active.

STATE VARIABLE

ALTSEL_Active

Parent: $\text{Modes}_{\text{v-34}}$.On ▷ $\text{ALTSEL}_{\text{v-82}}$.Selected ▷ $\text{ALTSEL\_Selected}_{\text{v-83}}$.Active

Type: $\{\text{Capture, Track}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_ALTSEL\_Active}_{\text{v-142}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED ⟶ Capture **if** TRUE

Capture ⟶ Track **if** $\text{Track\_ALTSEL}_{\text{m-78}}()$

Purpose: This variable maintains the current state of Altitude Select mode when it is active, i.e., whether it is captured or tracking. Note that if this side of the FGS is not active, it obtains its value from the off side. The normal transitions occur only when this side is active.

2.7.2.6 Vertical Approach (VAPPR) Mode

In Vertical Approach mode, the FGS provides guidance commands to track vertical guidance for ILS precision Glideslope approaches. Vertical Approach is an arming mode, i.e., after being selected, it must remained armed for a period of time before it can become active.

Definitions of Values to be Imported

MACRO

Select_VAPPR

Condition: When_APPR_Switch_Pressed_Seen$_{\text{m-115}}$()

Purpose: This event defines when Vertical Approach mode is to be selected.

MACRO

Activate_VAPPR

Condition:

AND	When_VAPPR_Track_Cond_Met_Seen$_{\text{m-99}}$()	T
	Is_LAPPR_Active$_{\text{v-52}}$	T
	Overspeed_Condition$_{\text{m-128}}$()	F

Purpose: This event defines when Vertical Approach mode is to transition from the armed to the active state.

MACRO

Deselect_VAPPR

Condition:

		OR					
	When_APPR_Switch_Pressed_Seen$_{m\text{-}115}$()	T	·	·	·	·	·
	When(Is_LAPPR_Selected$_{v\text{-}52}$ =FALSE)	·	T	·	·	·	·
AND	When_Selected_Nav_Source_Changed$_{v\text{-}127}$	·	·	T	·	·	·
	When_Selected_Nav_Frequency_Changed$_{v\text{-}127}$	·	·	·	T	·	·
	When_Pilot_Flying_Transfer$_{m\text{-}29}$()	·	·	·	·	T	·
	When(Modes$_{v\text{-}34}$ =Off)	·	·	·	·	·	T

Purpose: This event defines when Vertical Approach mode is to be deselected, i.e., to be cleared regardless of whether it is armed or active.

MACRO

Deactivate_VAPPR

Condition:

When_Nonbasic_Vertical_Mode_Activated$_{m\text{-}60}$()	T

Purpose: This event defines when Vertical Approach mode is to be cleared if it is active, but to remain armed if it is armed.

Definitions of Values to be Exported

STATE VARIABLE

Is_VAPPR_Selected

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= (VAPPR$_{v\text{-}89}$ =Selected) **if** TRUE

Purpose: Indicates if VAPPR mode is selected.

STATE VARIABLE

Is_VAPPR_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= (VAPPR_Selected$_{v\text{-}90}$ =Active) **if** TRUE

Purpose: Indicates if VAPPR mode is active.

MACRO

When_VAPPR_Activated

Condition:

AND	Activate_VAPPR$_{m\text{-}85}$()	T
	PREVIOUS STEP(VAPPR_Selected$_{v\text{-}90}$) =Armed	T
	Is_This_Side_Active$_{v\text{-}30}$	T

Purpose: This signal occurs when Vertical Approach mode is activated.

Comment: This event is defined this way to avoid circular dependencies. It would be preferable to define it as When(VAPPR_Selected = Active).

Definitions of Values to be Encapsulated

STATE VARIABLE

VAPPR

Parent: $\text{Modes}_{\text{v-34}}$.On

Type: $\{\text{Cleared, Selected}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_VAPPR}_{\text{v-143}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED ⟶ Cleared **if**

AND	$\text{Select\_VAPPR}_{\text{m-85}}()$	F
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

UNDEFINED ⟶ Selected **if**

AND	$\text{Select\_VAPPR}_{\text{m-85}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Cleared ⟶ Selected **if**

AND	$\text{Select\_VAPPR}_{\text{m-85}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Selected ⟶ Cleared **if**

			OR
AND	$\text{Deselect\_VAPPR}_{\text{m-86}}()$	T	·
	$\text{Deactivate\_VAPPR}_{\text{m-86}}()$	·	T
	PREVIOUS STEP($\text{VAPPR\_Selected}_{\text{v-90}}$) =Active	·	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T	T

Purpose: This variable maintains the current base state of Vertical Approach mode, i.e., whether it is cleared or selected. Note that if this side of the FGS is not active, it obtains its value from the off side. The normal transitions occur only when this side is active.

STATE VARIABLE

VAPPR_Selected

Parent: $\text{Modes}_{\text{v-34}}.\text{On} \triangleright \text{VAPPR}_{\text{v-89}}.\text{Selected}$

Type: $\{\text{Armed, Active}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_VAPPR\_Selected}_{\text{v-143}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED ⟶ Armed **if** TRUE

Armed ⟶ Active **if**

AND	$\text{Activate\_VAPPR}_{\text{m-85}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Purpose: This variable maintains the current state of Vertical Approach mode when it is selected, i.e., whether it is armed or active. Note that if this side of the FGS is not active, it obtains its value from the off side. The normal transitions occur only when this side is active.

2.7.2.7 Vertical Go Around (VGA) Mode

Go Around mode provides guidance to the pilot after aborting an approach. The AP is never engaged during Go Around mode. Vertical Go Around mode provides guidance to maintain a fixed pitch angle.

Definitions of Values to be Imported

MACRO

Select_VGA

Condition:

AND	When_GA_Switch_Pressed_Seen$_{\text{m-116}}$()	T
	Overspeed_Condition$_{\text{m-128}}$()	F

Purpose: This event defines when Vertical Go Around mode is to be selected.

MACRO

Deselect_VGA

Condition:

					OR				
AND	When(AP$_{\text{v-132}}$ =Engaged)	T	·	·	·	·	·	·
	When_Nonbasic_Lateral_Mode_Activated$_{\text{m-35}}$()	·	T	·	·	·	·	·
	When_Nonbasic_Vertical_Mode_Activated$_{\text{m-60}}$()	·	·	T	·	·	·	·
	When_SYNC_Switch_Pressed_Seen$_{\text{m-117}}$()	·	·	·	T	·	·	·
	When_VS_Pitch_Wheel_Rotated_Seen$_{\text{m-110}}$()	·	·	·	·	T	·	·
	When_Pilot_Flying_Transfer$_{\text{m-29}}$()	·	·	·	·	·	T	·
	When(Modes$_{\text{v-34}}$ =Off)	·	·	·	·	·	·	T

Purpose: This event defines when Vertical Go Around mode is to be deselected.

Definitions of Values to be Exported

STATE VARIABLE

Is_VGA_Selected

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($VGA_{v\text{-}94}$ =Selected) **if** TRUE

Purpose: Indicates if VGA mode is selected.

STATE VARIABLE

Is_VGA_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($VGA_{v\text{-}94}$ =Selected) **if** TRUE

Purpose: Indicates if VGA mode is active.

Comment: Even though VGA Selected and VGA Active are the same value, this is defined to provide a common set of controlled variables across all modes.

MACRO

When_VGA_Activated

Condition:

AND	Select_VGA$_{\text{m-91}}$()	T
	PREVIOUS STEP(VGA$_{\text{v-94}}$) =Selected	F
	Is_This_Side_Active$_{\text{v-30}}$	T

Purpose: This signal occurs when Vertical Go Around mode is activated.

Comment: This event is defined this way to avoid circular dependencies. It would be preferable to define it as When(LGA = Selected).

Definitions of Values to be Encapsulated

STATE VARIABLE

VGA

Parent: $\text{Modes}_{\text{v-34}}$.On

Type: $\{\text{Cleared, Selected}\}_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: State
:= $\text{Offside\_VGA}_{\text{v-144}}$ **if** **not** $\text{Is\_This\_Side\_Active}_{\text{v-30}}$

UNDEFINED ⟶ Cleared **if**

AND	$\text{Select\_VGA}_{\text{m-91}}()$	F
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

UNDEFINED ⟶ Selected **if**

AND	$\text{Select\_VGA}_{\text{m-91}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Cleared ⟶ Selected **if**

AND	$\text{Select\_VGA}_{\text{m-91}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Selected ⟶ Cleared **if**

AND	$\text{Deselect\_VGA}_{\text{m-91}}()$	T
	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$	T

Purpose: This variable maintains the current state of Vertical Go Around mode. Note that if this side is not active, it obtains its value from the off side. The normal transitions for Vertical Go Around mode occur only when this side is active.

2.8 Flight Control Laws

The Flight Control Laws generate the pitch and roll guidance commands provided to the Flight Director and the Autopilot. Different Flight Control Laws are armed or activated based on the current modes of the FGS. They also provide inputs to the mode logic that determine when transitions are made from the armed to the active state. While the Flight Control Laws are not specified in this model, this component defines the interface between them and the mode logic and defines the events provided by the Flight Control Laws that the mode logic depends upon. As with events provided by the Flight Control Panel, some of these are prioritized so that if multiple events occur at the same time, only the most significant events are seen by the other parts of the specification.

Definitions of Values to be Exported

MACRO

When_NAV_Track_Cond_Met

Condition: Is_NAV_Track_Cond_Met$_{\text{iv-100}}$

Purpose: This event indicates when the NAV Track Condition is met.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_NAV_Track_Cond_Met_Seen

Condition:

AND	When_NAV_Track_Cond_Met$_{\text{m-95}}$()	T
	No_Higher_Event_Than_NAV_Track_Cond_Met$_{\text{m-96}}$()	T

Purpose: This event indicates when the NAV Track condition is met and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_NAV_Track_Cond_Met

Condition:

A	When_FD_Switch_Pressed$_{\text{m-103}}$()	F
N		
D	No_Higher_Event_Than_FD_Switch_Pressed$_{\text{m-104}}$()	T

Purpose: This event occurs when no event with a priority higher than pressing the NAV switch has occurred.

MACRO

When_ALTSEL_Track_Cond_Met

Condition: Is_ALTSEL_Track_Cond_Met$_{\text{iv-101}}$

Purpose: This event indicates when the ALTSEL Track Condition is met.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_ALTSEL_Track_Cond_Met_Seen

Condition:

A	When_ALTSEL_Track_Cond_Met$_{\text{m-96}}$()	T
N		
D	No_Higher_Event_Than_ALTSEL_Track_Cond_Met$_{\text{m-97}}$()	T

Purpose: This event indicates when the ALTSEL Track condition is met and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_ALTSEL_Track_Cond_Met

Condition:

AND	When_FD_Switch_Pressed$_{\text{m-103}}$()	F
	No_Higher_Event_Than_FD_Switch_Pressed$_{\text{m-104}}$()	T

Purpose: This event occurs when no event with a priority higher than pressing the FD switch has occurred.

MACRO

When_ALTSEL_Capture_Cond_Met

Condition: Is_ALTSEL_Capture_Cond_Met$_{\text{iv-101}}$

Purpose: This event indicates when the ALTSEL Capture Condition is met.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_ALTSEL_Capture_Cond_Met_Seen

Condition:

AND	When_ALTSEL_Capture_Cond_Met$_{\text{m-97}}$()	T
	No_Higher_Event_Than_ALTSEL_Capture_Cond_Met$_{\text{m-98}}$()	T

Purpose: This event indicates when the ALTSEL Capture condition is met and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_ALTSEL_Capture_Cond_Met

Condition:

AND	When_FD_Switch_Pressed$_{\text{m-103}}$()	F
	No_Higher_Event_Than_FD_Switch_Pressed$_{\text{m-104}}$()	T

Purpose: This event occurs when no event with a priority higher than pressing the FD switch has occurred.

MACRO

When_LAPPR_Track_Cond_Met

Condition: Is_LAPPR_Track_Cond_Met$_{\text{iv-101}}$

Purpose: This event indicates when the LAPPR Track Condition is met.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_LAPPR_Track_Cond_Met_Seen

Condition:

AND	When_LAPPR_Track_Cond_Met$_{\text{m-98}}$()	T
	No_Higher_Event_Than_LAPPR_Track_Cond_Met$_{\text{m-99}}$()	T

Purpose: This event indicates when the LAPPR Track condition is met and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_LAPPR_Track_Cond_Met

Condition:

AND	When_FD_Switch_Pressed$_{\text{m-103}}$()	F
	No_Higher_Event_Than_FD_Switch_Pressed$_{\text{m-104}}$()	T

Purpose: This event occurs when no event with a priority higher than pressing the FD switch has occurred.

MACRO

When_VAPPR_Track_Cond_Met

Condition: Is_VAPPR_Track_Cond_Met$_{\text{iv-102}}$

Purpose: This event indicates when the VAPPR Track Condition is met.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_VAPPR_Track_Cond_Met_Seen

Condition:

AND	When_VAPPR_Track_Cond_Met$_{\text{m-99}}$()	T
	No_Higher_Event_Than_VAPPR_Track_Cond_Met$_{\text{m-100}}$()	T

Purpose: This event indicates when the VAPPR Track condition is met and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_VAPPR_Track_Cond_Met

Condition:

A	When_FD_Switch_Pressed$_{\text{m-103}}()$	F
N		
D	No_Higher_Event_Than_FD_Switch_Pressed$_{\text{m-104}}()$	T

Purpose: This event occurs when no event with a priority higher than pressing the FD switch has occurred.

Definitions of Values to be Encapsulated

INPUT VARIABLE

Is_NAV_Track_Cond_Met

Type: BOOLEAN
Initial Value: FALSE
Classified as: MONITORED
Purpose: This input variable becomes true when the lateral navigation (NAV) flight control law transitions from the armed to the track state.

INPUT VARIABLE

Is_ALTSEL_Capture_Cond_Met

Type: BOOLEAN
Initial Value: FALSE
Classified as: MONITORED
Purpose: This input variable becomes true when the vertical altitude select flight control law transitions from the armed to the capture state.

INPUT VARIABLE

Is_ALTSEL_Track_Cond_Met

Type: BOOLEAN
Initial Value: FALSE
Classified as: MONITORED
Purpose: This input variable becomes true when the vertical altitude select flight control law transitions from the capture to the track state.

INPUT VARIABLE

Is_LAPPR_Track_Cond_Met

Type: BOOLEAN
Initial Value: FALSE
Classified as: MONITORED
Purpose: This input variable becomes true when the lateral approach (LAPPR) flight control law transitions from the armed to the track state.

INPUT VARIABLE

Is_VAPPR_Track_Cond_Met

Type: BOOLEAN
Initial Value: FALSE
Classified as: MONITORED
Purpose: This input variable becomes true when the vertical approach (VAPPR) flight control law transitions from the armed to the track state.

2.9 Flight Control Panel (FCP)

The Flight Control Panel (FCP) is the primary device with which the flight crew interacts with the FGS. This component defines the FGS's view of the FCP. This consists of the definition of the monitored variables representing the switches and dials of the FCP and the controlled variables representing the lamps displayed on the FCP. A few additional inputs, such as the SYNC and AP Disengage switches on the control yokes and the GA switches on the throttles, are routed through the FCP aand are defined here. This component also defines FCP specific events, such as the pressing of a switch, used in the rest of the specification. These events are prioritized so that if multiple events occur at the same time, only the most significant events are seen by the other parts of the specification.

Definitions of Values to be Exported

MACRO

When_FD_Switch_Pressed

Condition: When($\text{FD\_Switch}_{\text{iv-118}}$ =ON)

Purpose: This event indicates when the FD switch associated with this FGS is pressed.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_FD_Switch_Pressed_Seen

Condition:

AND	$\text{When\_FD\_Switch\_Pressed}_{\text{m-103}}()$	T
	$\text{No\_Higher\_Event\_Than\_FD\_Switch\_Pressed}_{\text{m-104}}()$	T

Purpose: This event occurs when the FD Switch is pressed and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_FD_Switch_Pressed

Condition:

AND	When_Transfer_Switch_Pressed$_{\text{m-104}}$()	F
	No_Higher_Event_Than_Transfer_Switch_Pressed$_{\text{m-105}}$()	T

Purpose: This event occurs when no event with a priority higher than pressing the FD switch has occurred.

MACRO

When_Transfer_Switch_Pressed

Condition: When(Transfer_Switch$_{\text{iv-125}}$ =ON)

Purpose: This event indicates when the TRANSFER switch ais pressed.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_Transfer_Switch_Pressed_Seen

Condition:

AND	When_Transfer_Switch_Pressed$_{\text{m-104}}$()	T
	No_Higher_Event_Than_Transfer_Switch_Pressed$_{\text{m-105}}$()	T

Purpose: This event indicates when the Transfer switch is pressed and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_Transfer_Switch_Pressed

Condition:

	When_AP_Engage_Switch_Pressed$_{\text{m-105}}$()	F
	No_Higher_Event_Than_AP_Engage_Switch_Pressed$_{\text{m-106}}$()	T
AND	When_VS_Pitch_Wheel_Rotated$_{\text{m-109}}$()	F
	No_Higher_Event_Than_VS_Pitch_Wheel_Rotated$_{\text{m-110}}$()	T
	When_NAV_Switch_Pressed$_{\text{m-107}}$()	F
	No_Higher_Event_Than_NAV_Switch_Pressed$_{\text{m-108}}$()	T

Purpose: This event occurs when no event with a priority higher than pressing the Transfer switch has occurred.

MACRO

When_AP_Engage_Switch_Pressed

Condition: When(AP_Engage_Switch$_{\text{iv-126}}$ =ON)

Purpose: This event indicates when the AP Engaged switch is pressed.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_AP_Engage_Switch_Pressed_Seen

Condition:

AND	When_AP_Engage_Switch_Pressed$_{\text{m-105}}$()	T
	No_Higher_Event_Than_AP_Engage_Switch_Pressed$_{\text{m-106}}$()	T

Purpose: This event indicates when the AP Engage switch is pressed and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_AP_Engage_Switch_Pressed

Condition:

AND	When_AP_Disconnect_Switch_Pressed$_{\text{m-106}}$()	F
	No_Higher_Event_Than_AP_Disconnect_Switch_Pressed$_{\text{m-107}}$()	T

Purpose: This event occurs when no event with a priority higher than pressing the AP Engage switch has occurred.

MACRO

When_AP_Disconnect_Switch_Pressed

Condition: When(AP_Disconnect_Switch$_{\text{iv-126}}$ =ON)

Purpose: This event indicates when the AP Disconnect switch is pressed.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_AP_Disconnect_Switch_Pressed_Seen

Condition:

AND	When_AP_Disconnect_Switch_Pressed$_{\text{m-106}}$()	T
	No_Higher_Event_Than_AP_Disconnect_Switch_Pressed$_{\text{m-107}}$()	T

Purpose: This event indicates when the AP Disconnect switch is pressed and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_AP_Disconnect_Switch_Pressed

Condition: TRUE

Purpose: This event occurs when no event with a priority higher than pressing the AP Disconnect switch has occurred.

Lateral Switches — MACRO

When_NAV_Switch_Pressed

Condition: When(NAV_Switch$_{\text{iv-119}}$ =ON)

Purpose: This event indicates when the NAV switch is pressed.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_NAV_Switch_Pressed_Seen

Condition:

AND	When_NAV_Switch_Pressed$_{\text{m-107}}$()	T
	No_Higher_Event_Than_NAV_Switch_Pressed$_{\text{m-108}}$()	T

Purpose: This event indicates when the NAV Switch is pressed and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_NAV_Switch_Pressed

Condition:

AND	When_HDG_Switch_Pressed$_{\text{m-108}}$()	F
	No_Higher_Event_Than_HDG_Switch_Pressed$_{\text{m-109}}$()	T

Purpose: This event occurs when no event with a priority higher than pressing the NAV switch has occurred.

MACRO

When_HDG_Switch_Pressed

Condition: When(HDG_Switch$_{\text{iv-118}}$ =ON)

Purpose: This event indicates when the HDG switch is pressed.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_HDG_Switch_Pressed_Seen

Condition:

AND	When_HDG_Switch_Pressed$_{\text{m-108}}$()	T
	No_Higher_Event_Than_HDG_Switch_Pressed$_{\text{m-109}}$()	T

Purpose: This event indicates when the NAV Switch is pressed and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_HDG_Switch_Pressed

Condition:

AND	When_APPR_Switch_Pressed$_{\text{m-114}}$()	F
	No_Higher_Event_Than_APPR_Switch_Pressed$_{\text{m-115}}$()	T

Purpose: This event occurs when no event with a priority higher than pressing the HDG switch has occurred.

Vertical Switches

MACRO

When_VS_Pitch_Wheel_Rotated

Condition: When(VS_Pitch_Wheel_In_Motion$_{\text{iv-125}}$)

Purpose: This event indicates when the VS Pitch Wheel is rotated.

MACRO

When_VS_Pitch_Wheel_Rotated_Seen

Condition:

A	When_VS_Pitch_Wheel_Rotated$_{\text{m-109}}$()	T
N		
D	No_Higher_Event_Than_VS_Pitch_Wheel_Rotated$_{\text{m-110}}$()	T

Purpose: This event indicates when the VS Pitch Wheel is rotated and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_VS_Pitch_Wheel_Rotated

Condition:

A	When_VS_Switch_Pressed$_{\text{m-110}}$()	F
N		
D	No_Higher_Event_Than_VS_Switch_Pressed$_{\text{m-111}}$()	T

Purpose: This event occurs when no event with a priority higher than rotating the VS Pitch Wheel has occurred.

MACRO

When_VS_Switch_Pressed

Condition: When(VS_Switch$_{\text{iv-120}}$ =ON)

Purpose: This event indicates when the VS switch is pressed.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_VS_Switch_Pressed_Seen

Condition:

AND	When_VS_Switch_Pressed$_{\text{m-110}}$()	T
	No_Higher_Event_Than_VS_Switch_Pressed$_{\text{m-111}}$()	T

Purpose: This event indicates when the VS switch pressed and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_VS_Switch_Pressed

Condition:

AND	When_FLC_Switch_Pressed$_{\text{m-111}}$()	F
	No_Higher_Event_Than_FLC_Switch_Pressed$_{\text{m-112}}$()	T

Purpose: This event occurs when no event with a priority higher than pressing the VS switch has occurred.

MACRO

When_FLC_Switch_Pressed

Condition: When(FLC_Switch$_{\text{iv-122}}$ =ON)

Purpose: This event indicates when the FLC switch is pressed.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_FLC_Switch_Pressed_Seen

Condition:

AND	When_FLC_Switch_Pressed$_{\text{m-111}}$()	T
	No_Higher_Event_Than_FLC_Switch_Pressed$_{\text{m-112}}$()	T

Purpose: This event indicates when the FLC switch pressed and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_FLC_Switch_Pressed

Condition:

AND	When_ALT_Switch_Pressed$_{\text{m-112}}$()	F
	No_Higher_Event_Than_ALT_Switch_Pressed$_{\text{m-113}}$()	T

Purpose: This event occurs when no event with a priority higher than pressing the FLC switch has occurred.

MACRO

When_ALT_Switch_Pressed

Condition: When(ALT_Switch$_{\text{iv-121}}$ =ON)

Purpose: This event indicates when the ALT switch is pressed.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_ALT_Switch_Pressed_Seen

Condition:

AND	When_ALT_Switch_Pressed$_{\text{m-112}}$()	T
	No_Higher_Event_Than_ALT_Switch_Pressed$_{\text{m-113}}$()	T

Purpose: This event indicates when the ALT switch pressed and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_ALT_Switch_Pressed

Condition:

AND	When_ALTSEL_Target_Altitude_Changed$_{\text{m-113}}$()	F
	No_Higher_Event_Than_ALTSEL_Target_Altitude_Changed$_{\text{m-114}}$()	T

Purpose: This event occurs when no event with a priority higher than pressing the ALT switch has occurred.

MACRO

When_ALTSEL_Target_Altitude_Changed

Condition:

AND	ALTSEL_Target_Altitude_Changed$_{\text{iv-129}}$ =UNDEFINED	F
	ALTSEL_Target_Altitude_Changed$_{\text{iv-129}}$	T

Purpose: This event indicates when the ALTSEL target altitude is changed.

MACRO

When_ALTSEL_Target_Altitude_Changed_Seen

Condition:

AND	When_ALTSEL_Target_Altitude_Changed$_{\text{m-113}}$()	T
	No_Higher_Event_Than_ALTSEL_Target_Altitude_Changed$_{\text{m-114}}$()	T

Purpose: This event indicates when the ALTSEL target altitude is changed and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_ALTSEL_Target_Altitude_Changed

Condition:

AND	When_APPR_Switch_Pressed$_{\text{m-114}}$()	F
	No_Higher_Event_Than_APPR_Switch_Pressed$_{\text{m-115}}$()	T

Purpose: This event occurs when no event with a priority higher than rotating the changing of the ALTSEL target altitude has occurred.

Lateral and Vertical Events — MACRO

When_APPR_Switch_Pressed

Condition: When(APPR_Switch$_{\text{iv-123}}$ =ON)

Purpose: This event indicates when the APPR switch is pressed.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_APPR_Switch_Pressed_Seen

Condition:

AND	When_APPR_Switch_Pressed$_{\text{m-114}}$()	T
	No_Higher_Event_Than_APPR_Switch_Pressed$_{\text{m-115}}$()	T

Purpose: This event indicates when the APPR switch pressed and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_APPR_Switch_Pressed

Condition:

AND	When_GA_Switch_Pressed$_{\text{m-115}}$()	F
	No_Higher_Event_Than_GA_Switch_Pressed$_{\text{m-116}}$()	T

Purpose: This event occurs when no event with a priority higher than pressing the APPR switch has occurred.

MACRO

When_GA_Switch_Pressed

Condition: When(GA_Switch$_{\text{iv-124}}$ =ON)

Purpose: This event indicates when the GA switch is pressed.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_GA_Switch_Pressed_Seen

Condition:

AND	When_GA_Switch_Pressed$_{\text{m-115}}$()	T
	No_Higher_Event_Than_GA_Switch_Pressed$_{\text{m-116}}$()	T

Purpose: This event indicates when the GA switch pressed and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_GA_Switch_Pressed

Condition:

AND	When_SYNC_Switch_Pressed$_{\text{m-116}}$()	F
	No_Higher_Event_Than_SYNC_Switch_Pressed$_{\text{m-117}}$()	T

Purpose: This event occurs when no event with a priority higher than pressing the GA switch has occurred.

MACRO

When_SYNC_Switch_Pressed

Condition: When(SYNC_Switch$_{\text{iv-125}}$ =ON)

Purpose: This event indicates when the SYNC switch is pressed.

Comment: This is redefined as a macro to simplify verification.

MACRO

When_SYNC_Switch_Pressed_Seen

Condition:

AND	When_SYNC_Switch_Pressed$_{\text{m-116}}$()	T
	No_Higher_Event_Than_SYNC_Switch_Pressed$_{\text{m-117}}$()	T

Purpose: This event indicates when the SYNC switch pressed and no higher priority event has occurred.

MACRO

No_Higher_Event_Than_SYNC_Switch_Pressed

Condition: TRUE

Purpose: This event occurs when no event with a priority higher than pressing the GA switch has occurred.

Definitions of Values to be Encapsulated

TYPE

Switch

Possible Values: OFF, ON

TYPE

Lamp

Possible Values: OFF, ON

INPUT VARIABLE

FD_Switch

Type: $\{\text{OFF, ON}\}_{\text{ty-117}}$
Initial Value: UNDEFINED
Classified as: MONITORED
Purpose: Holds the last sensed position of the FD switch associated with this FGS.

INPUT VARIABLE

HDG_Switch

Type: $\{\text{OFF, ON}\}_{\text{ty-117}}$
Initial Value: UNDEFINED
Classified as: MONITORED
Purpose: Holds the last sensed position of the HDG switch.

STATE VARIABLE

HDG_Lamp

Parent: NONE

Type: $\{OFF, ON\}_{ty\text{-}118}$
Initial Value: OFF
Classified as: CONTROLLED
:= ON **if** $Is\_HDG\_Selected_{v\text{-}41}$

:= OFF **if** **not** $Is\_HDG\_Selected_{v\text{-}41}$

Purpose: Indicates if the HDG switch lamp on the FCP should be on or off.

INPUT VARIABLE

NAV_Switch

Type: $\{OFF, ON\}_{ty\text{-}117}$
Initial Value: UNDEFINED
Classified as: MONITORED
Purpose: Holds the last sensed position of the NAV switch.

STATE VARIABLE

NAV_Lamp

Parent: NONE

Type: $\{\text{OFF, ON}\}_{\text{ty-118}}$
Initial Value: OFF
Classified as: CONTROLLED
:= ON **if** Is_NAV_Selected$_{\text{v-46}}$

:= OFF **if** **not** Is_NAV_Selected$_{\text{v-46}}$

Purpose: Indicates if the NAV switch lamp should be on or off.

INPUT VARIABLE

VS_Switch

Type: $\{\text{OFF, ON}\}_{\text{ty-117}}$
Initial Value: UNDEFINED
Classified as: MONITORED
Purpose: Holds the last sensed position of the VS switch.

STATE VARIABLE

VS_Lamp

Parent: NONE

Type: $\{\text{OFF, ON}\}_{\text{ty-118}}$
Initial Value: OFF
Classified as: CONTROLLED
:= ON **if** $\text{Is\_VS\_Selected}_{\text{v-66}}$

:= OFF **if** **not** $\text{Is\_VS\_Selected}_{\text{v-66}}$

Purpose: Indicates if the VS switch lamp should be on or off.

INPUT VARIABLE

ALT_Switch

Type: $\{\text{OFF, ON}\}_{\text{ty-117}}$
Initial Value: UNDEFINED
Classified as: MONITORED
Purpose: Holds the last sensed position of the ALT switch.

STATE VARIABLE

ALT_Lamp

Parent: NONE

Type: $\{\text{OFF, ON}\}_{\text{ty-118}}$
Initial Value: OFF
Classified as: CONTROLLED
:= ON **if** $\text{Is\_ALT\_Selected}_{\text{v-74}}$

:= OFF **if** **not** $\text{Is\_ALT\_Selected}_{\text{v-74}}$

Purpose: Indicates if the ALT switch lamp should be on or off.

INPUT VARIABLE

FLC_Switch

Type: $\{\text{OFF, ON}\}_{\text{ty-117}}$
Initial Value: UNDEFINED
Classified as: MONITORED
Purpose: Holds the last sensed position of the FLC switch.

STATE VARIABLE

FLC_Lamp

Parent: NONE

Type: $\{\text{OFF, ON}\}_{\text{ty-118}}$
Initial Value: OFF
Classified as: CONTROLLED
:= ON **if** $\text{Is\_FLC\_Selected}_{\text{v-70}}$

:= OFF **if** **not** $\text{Is\_FLC\_Selected}_{\text{v-70}}$

Purpose: Indicates if the FLC switch lamp should be on or off.

INPUT VARIABLE

APPR_Switch

Type: $\{\text{OFF, ON}\}_{\text{ty-117}}$
Initial Value: UNDEFINED
Classified as: MONITORED
Purpose: Holds the last sensed position of the APPR switch.

STATE VARIABLE

APPR_Lamp

Parent: NONE

Type: $\{\text{OFF, ON}\}_{\text{ty-118}}$
Initial Value: OFF
Classified as: CONTROLLED
:= ON **if**

		OR	
AND	$\text{Is\_LAPPR\_Selected}_{\text{v-52}}$	T	·
	$\text{Is\_VAPPR\_Selected}_{\text{v-87}}$	·	T

:= OFF **if**

AND	$\text{Is\_LAPPR\_Selected}_{\text{v-52}}$	F
	$\text{Is\_VAPPR\_Selected}_{\text{v-87}}$	F

Purpose: Indicates if the APPR switch lamp should be on or off.

INPUT VARIABLE

GA_Switch

Type: $\{\text{OFF, ON}\}_{\text{ty-117}}$
Initial Value: UNDEFINED
Classified as: MONITORED
Purpose: Holds the last sensed position of the GA switch.

INPUT VARIABLE

SYNC_Switch

Type: $\{\text{OFF, ON}\}_{\text{ty-117}}$
Initial Value: UNDEFINED
Classified as: MONITORED
Purpose: Holds the last sensed position of the SYNC switch.

INPUT VARIABLE

Transfer_Switch

Type: $\{\text{OFF, ON}\}_{\text{ty-117}}$
Initial Value: UNDEFINED
Classified as: MONITORED
Purpose: Holds the last sensed position of the TRANSFER switch.

INPUT VARIABLE

VS_Pitch_Wheel_In_Motion

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: MONITORED
Purpose: Holds the last sensed position of the VS/Pitch wheel.

INPUT VARIABLE

AP_Engage_Switch

Type: $\{\text{OFF, ON}\}_{\text{ty-117}}$
Initial Value: UNDEFINED
Classified as: MONITORED
Purpose: Holds the last sensed position of the AP Engage switch.

STATE VARIABLE

AP_Lamp

Parent: NONE

Type: $\{\text{OFF, ON}\}_{\text{ty-118}}$
Initial Value: OFF
Classified as: CONTROLLED
:= ON **if** $\text{Is\_AP\_Engaged}_{\text{v-131}}$

:= OFF **if** **not** $\text{Is\_AP\_Engaged}_{\text{v-131}}$

Purpose: Indicates if the AP switch lamp should be on or off.

INPUT VARIABLE

AP_Disconnect_Switch

Type: $\{\text{OFF, ON}\}_{\text{ty-117}}$
Initial Value: UNDEFINED
Classified as: MONITORED
Purpose: Holds the last sensed position of the AP Disconnect switch.

2.10 Navigation Sources

The FGS receives navigation information from several sources, including VHF Omni-Directional Range (VOR) for lateral navigation, Localizer (LOC) for precision lateral approach, and Glideslope (GS) for precision vertical approach. This component defines the FGS's view of its navigation sources. This includes definition of the types of sources available, the currently selected source, and the information available from that source.

Definitions of Values to be Exported

STATE VARIABLE

When_Selected_Nav_Source_Changed

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: MONITORED
Purpose: This event occurs when a new navigation source is selected.

STATE VARIABLE

When_Selected_Nav_Frequency_Changed

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: MONITORED
Purpose: This event occurs when the frequency of the selected navigation source is changed.

2.11 Air Data System

The Air Data System provides information about the aircraft state sensed from the surrounding air mass such as whether an overspeed condition exists. This component defines the FGS's view of the Air Data System.

Definitions of Values to be Exported

MACRO

Overspeed_Condition

Condition:

AND	$\text{Overspeed}_{\text{iv-128}} \neq$ UNDEFINED	T
	$\text{Overspeed}_{\text{iv-128}} =$ TRUE	T

Definitions of Values to be Encapsulated

INPUT VARIABLE

Overspeed

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: MONITORED
Purpose: This state variable indicates if an overspeed condition exists.

2.12 References

References refer to target values used by several of the modes, such as the Preselected Altitude. These may be set by the flight crew through the Flight Control Panel, or set to the current aircraft value on entry to a mode or a synchronization request. The references may be physically maintained by a variety of systems. This component defines the FGS's view of the references. Details of how these values are obtained from other systems are encapsulated within this component.

Definitions of Values to be Exported

INPUT VARIABLE

ALTSEL_Target_Altitude_Changed

Type: BOOLEAN
Initial Value: TRUE
Classified as: MONITORED
Purpose: This input variable holds the most recent information on whether the ALTSEL target altitude (either the PSA or FPTA) has been changed.

2.13 Autopilot (AP)

The Autopilot (AP) commands movement of the control surfaces based on the pitch and roll commands generated by the FGS. This component defines the FGS's view of the AP.

Definitions of Values to be Imported

MACRO

When_Engage_AP

Condition: When_AP_Engage_Switch_Pressed_Seen$_{\text{m-106}}$()

Purpose: This event defines when the AP is to be engaged.

MACRO

When_Disengage_AP

Condition:

		OR		
AND	When_AP_Engage_Switch_Pressed_Seen$_{\text{m-106}}$()	T	·	·
	When_AP_Disconnect_Switch_Pressed_Seen$_{\text{m-107}}$()	·	T	·
	When_GA_Switch_Pressed_Seen$_{\text{m-116}}$()	·	·	T

Purpose: This event defines when the AP is to be disengaged.

Definitions of Values to be Exported

STATE VARIABLE

Is_AP_Engaged

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: CONTROLLED
:= ($AP_{v\text{-}132}$ =Engaged) **if** TRUE

Purpose: Indicates if the AP is engaged.

Definitions of Values to be Encapsulated

TYPE

AP_State

Possible Values: Disengaged, Engaged

STATE VARIABLE

AP

Parent: NONE

Type: $\{\text{Disengaged, Engaged}\}_{\text{ty-131}}$
Initial Value: Disengaged
Classified as: State
Disengaged $\longrightarrow$ Engaged **if** When_Engage_AP$_{\text{m-130}}()$

Engaged $\longrightarrow$ Disengaged **if** When_Disengage_AP$_{\text{m-130}}()$

Purpose: This variable maintains the current state of the autopilot.

2.14 Offside FGS

The FGS has two physical sides, or channels, on the left and right sides of the aircraft. These provide redundant implementations that communicate with each other over a cross-channel bus. This component defines this side of the FGS's view of the other side.

Definitions of Values to be Exported

STATE VARIABLE

Offside_FD

Parent: NONE

Type: $\{\text{Off, On}\}_{\text{ty-23}}$
Initial Value: Off
Classified as: MONITORED
:= Off **if** **not** $\text{Offside\_FD\_On}_{\text{iv-145}}$

:= On **if** $\text{Offside\_FD\_On}_{\text{iv-145}}$

Purpose: This variable holds the most recent information on the state of the FD on the offside FD.

STATE VARIABLE

Is_Offside_LAPPR_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: MONITORED
:= $\text{Offside\_LAPPR\_Selected}_{\text{v-138}}$ =Active **if** TRUE

Purpose: Indicates if offside LAPPR mode is active.

STATE VARIABLE

Is_Offside_VAPPR_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: MONITORED
:= $\text{Offside\_VAPPR\_Selected}_{\text{v-143}}$ =Active **if** TRUE

Purpose: Indicates if offside VAPPR mode is active.

STATE VARIABLE

Is_Offside_VGA_Active

Parent: NONE

Type: BOOLEAN
Initial Value: FALSE
Classified as: MONITORED
:= Offside_VGA$_{v\text{-}144}$ =Selected **if** TRUE

Purpose: Indicates if offside VGA mode is active.

STATE VARIABLE

Offside_Modes

Parent: NONE

Type: $\{\text{Off, On}\}_{ty\text{-}23}$
Initial Value: Off
Classified as: MONITORED
:= Off **if** **not** Offside_Modes_On$_{iv\text{-}145}$

:= On **if** Offside_Modes_On$_{iv\text{-}145}$

Purpose: This variable holds the most recent information on the state of the mode annunciations on the offside FGS.

STATE VARIABLE

Offside_ROLL

Parent: Offside_Modes$_{\text{v-135}}$.On

Type: {Cleared, Selected}$_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Cleared **if** **not** Offside_Roll_Selected$_{\text{iv-145}}$

:= Selected **if** Offside_Roll_Selected$_{\text{iv-145}}$

Purpose: This variable holds the most recent information on the base state of the lateral ROLL mode of the offside FGS.

STATE VARIABLE

Offside_HDG

Parent: Offside_Modes$_{\text{v-135}}$.On

Type: {Cleared, Selected}$_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Cleared **if** **not** Offside_Hdg_Selected$_{\text{iv-146}}$

:= Selected **if** Offside_Hdg_Selected$_{\text{iv-146}}$

Purpose: This variable holds the most recent information on the base state of the lateral HDG mode of the offside FGS.

STATE VARIABLE

Offside_NAV

Parent: Offside_Modes$_{\text{v-135}}$.On

Type: {Cleared, Selected}$_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Cleared **if** **not** Offside_Nav_Selected$_{\text{iv-146}}$

:= Selected **if** Offside_Nav_Selected$_{\text{iv-146}}$

Purpose: This variable holds the most recent information on the base state of the lateral NAV mode of the offside FGS.

STATE VARIABLE

Offside_NAV_Selected

Parent: Offside_Modes$_{\text{v-135}}$.On ▷ Offside_NAV$_{\text{v-137}}$.Selected

Type: {Armed, Active}$_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Armed **if** **not** Offside_Nav_Active$_{\text{iv-146}}$

:= Active **if** Offside_Nav_Active$_{\text{iv-146}}$

Purpose: This variable holds the most recent information on the selected state of the lateral NAV mode of the offside FGS.

STATE VARIABLE

Offside_LAPPR

Parent: Offside_Modes$_{v\text{-}135}$.On

Type: {Cleared, Selected}$_{ty\text{-}23}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Cleared **if** **not** Offside_Lappr_Selected$_{iv\text{-}147}$

:= Selected **if** Offside_Lappr_Selected$_{iv\text{-}147}$

Purpose: This variable holds the most recent information on the base state of the lateral APPR mode of the offside FGS.

STATE VARIABLE

Offside_LAPPR_Selected

Parent: Offside_Modes$_{v\text{-}135}$.On ▷ Offside_LAPPR$_{v\text{-}138}$.Selected

Type: {Armed, Active}$_{ty\text{-}23}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Armed **if** **not** Offside_Lappr_Active$_{iv\text{-}147}$

:= Active **if** Offside_Lappr_Active$_{iv\text{-}147}$

Purpose: This variable holds the most recent information on the selected state of the lateral APPR mode of the offside FGS.

STATE VARIABLE

Offside_LGA

Parent: Offside_Modes$_{\text{v-135}}$.On

Type: {Cleared, Selected}$_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Cleared **if** **not** Offside_Lga_Selected$_{\text{iv-147}}$

:= Selected **if** Offside_Lga_Selected$_{\text{iv-147}}$

Purpose: This variable holds the most recent information on the base state of the lateral GA mode of the offside FGS.

STATE VARIABLE

Offside_PITCH

Parent: Offside_Modes$_{\text{v-135}}$.On

Type: {Cleared, Selected}$_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Cleared **if** **not** Offside_Pitch_Selected$_{\text{iv-148}}$

:= Selected **if** Offside_Pitch_Selected$_{\text{iv-148}}$

Purpose: This variable holds the most recent information on the base state of the vertical PITCH mode of the offside FGS.

STATE VARIABLE

Offside_VS

Parent: Offside_Modes$_{v\text{-}135}$.On

Type: {Cleared, Selected}$_{ty\text{-}23}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Cleared **if** **not** Offside_Vs_Selected$_{iv\text{-}148}$

:= Selected **if** Offside_Vs_Selected$_{iv\text{-}148}$

Purpose: This variable holds the most recent information on the base state of the vertical VS mode of the offside FGS.

STATE VARIABLE

Offside_FLC

Parent: Offside_Modes$_{v\text{-}135}$.On

Type: {Cleared, Selected}$_{ty\text{-}23}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Cleared **if** **not** Offside_Flc_Selected$_{iv\text{-}148}$

:= Selected **if** Offside_Flc_Selected$_{iv\text{-}148}$

Purpose: This variable holds the most recent information on the base state of the vertical FLC mode of the offside FGS.

STATE VARIABLE

Offside_ALT

Parent: Offside_Modes$_{\text{v-135}}$.On

Type: {Cleared, Selected}$_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Cleared **if** **not** Offside_Alt_Selected$_{\text{iv-149}}$

:= Selected **if** Offside_Alt_Selected$_{\text{iv-149}}$

Purpose: This variable holds the most recent information on the base state of the vertical ALT mode of the offside FGS.

STATE VARIABLE

Offside_ALTSEL

Parent: Offside_Modes$_{\text{v-135}}$.On

Type: {Cleared, Selected}$_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Cleared **if** **not** Offside_Altsel_Selected$_{\text{iv-149}}$

:= Selected **if** Offside_Altsel_Selected$_{\text{iv-149}}$

Purpose: This variable holds the most recent information on the base state of the vertical ALTSEL mode of the offside FGS.

STATE VARIABLE

Offside_ALTSEL_Selected

Parent: Offside_Modes$_{\text{v-135}}$.On ▷ Offside_ALTSEL$_{\text{v-141}}$.Selected

Type: {Armed, Active}$_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Armed **if** **not** Offside_Altsel_Active$_{\text{iv-149}}$

:= Active **if** Offside_Altsel_Active$_{\text{iv-149}}$

Purpose: This variable holds the most recent information on the selected state of the vertical ALTSEL mode of the offside FGS.

STATE VARIABLE

Offside_ALTSEL_Active

Parent:
Offside_Modes$_{\text{v-135}}$.On ▷ Offside_ALTSEL$_{\text{v-141}}$.Selected ▷ Offside_ALTSEL_Selected$_{\text{v-142}}$.Active

Type: {Capture, Track}$_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Capture **if** **not** Offside_Altsel_Track$_{\text{iv-150}}$

:= Track **if** Offside_Altsel_Track$_{\text{iv-150}}$

Purpose: This variable holds the most recent information on the active state of the vertical ALTSEL mode of the offside FGS.

STATE VARIABLE

Offside_VAPPR

Parent: Offside_Modes$_{\text{v-135}}$.On

Type: {Cleared, Selected}$_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Cleared **if** **not** Offside_Vappr_Selected$_{\text{iv-150}}$

:= Selected **if** Offside_Vappr_Selected$_{\text{iv-150}}$

Purpose: This variable holds the most recent information on the base state of the vertical APPR mode of the offside FGS.

STATE VARIABLE

Offside_VAPPR_Selected

Parent: Offside_Modes$_{\text{v-135}}$.On ▷ Offside_VAPPR$_{\text{v-143}}$.Selected

Type: {Armed, Active}$_{\text{ty-23}}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Armed **if** **not** Offside_Vappr_Active$_{\text{iv-150}}$

:= Active **if** Offside_Vappr_Active$_{\text{iv-150}}$

Purpose: This variable holds the most recent information on the selected state of the vertical APPR mode of the offside FGS.

STATE VARIABLE

Offside_VGA

Parent: Offside_Modes$_{v\text{-}135}$.On

Type: {Cleared, Selected}$_{ty\text{-}23}$
Initial Value: UNDEFINED
Classified as: MONITORED
:= Cleared **if** **not** Offside_Vga_Selected$_{iv\text{-}151}$

:= Selected **if** Offside_Vga_Selected$_{iv\text{-}151}$

Purpose: This variable holds the most recent information on the base state of the vertical GA mode of the offside FGS.

Definitions of Values to be Encapsulated

INPUT VARIABLE

Offside_FGS_Active

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside FGS believes it is active.

INPUT VARIABLE

Offside_FD_On

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside FD is turned on.

INPUT VARIABLE

Offside_Modes_On

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside modes are on.

INPUT VARIABLE

Offside_Roll_Selected

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside ROLL mode is selected.

INPUT VARIABLE

Offside_Hdg_Selected

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside Heading mode is selected.

INPUT VARIABLE

Offside_Nav_Selected

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside Nav mode is selected.

INPUT VARIABLE

Offside_Nav_Active

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside Nav mode is active.

INPUT VARIABLE

Offside_Lappr_Selected

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside Lappr mode is selected.

INPUT VARIABLE

Offside_Lappr_Active

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside Lappr mode is active.

INPUT VARIABLE

Offside_Lga_Selected

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside Lga mode is selected.

INPUT VARIABLE

Offside_Pitch_Selected

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside vertical Pitch mode is selected.

INPUT VARIABLE

Offside_Vs_Selected

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside vertical Speed mode is selected.

INPUT VARIABLE

Offside_Flc_Selected

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside vertical Flight Level Change mode is selected.

INPUT VARIABLE

Offside_Alt_Selected

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside vertical Altitude Hold mode is selected.

INPUT VARIABLE

Offside_Altsel_Selected

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside vertical Altitude Select mode is selected.

INPUT VARIABLE

Offside_Altsel_Active

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside vertical Altitude Select mode is active.

INPUT VARIABLE

Offside_Altsel_Track

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside vertical Altitude Select mode is tracking.

INPUT VARIABLE

Offside_Vappr_Selected

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside Vappr mode is selected.

INPUT VARIABLE

Offside_Vappr_Active

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside Vappr mode is active.

INPUT VARIABLE

Offside_Vga_Selected

Type: BOOLEAN
Initial Value: UNDEFINED
Classified as: Input
Purpose: This input variable holds the most recent information on whether the offside Vga mode is selected.

2.15 FGS Inputs

This section defines the physical interface for all inputs to the FGS. The input variables associated with these fields are defined in the part of the specification to which they are logically related.

MESSAGE

Other_Input_Msg

Fields:	AltSel	**is**	BOOLEAN
	, AltselAct	**is**	BOOLEAN
	, AltselSel	**is**	BOOLEAN
	, AltselTrk	**is**	BOOLEAN
	, FdOn	**is**	BOOLEAN
	, FGSActive	**is**	BOOLEAN
	, FlcSel	**is**	BOOLEAN
	, HdgSel	**is**	BOOLEAN
	, LapprAct	**is**	BOOLEAN
	, LapprSel	**is**	BOOLEAN
	, LgaSel	**is**	BOOLEAN
	, ModesOn	**is**	BOOLEAN
	, NavAct	**is**	BOOLEAN
	, NavSel	**is**	BOOLEAN
	, PthSel	**is**	BOOLEAN
	, RollSel	**is**	BOOLEAN
	, VapprAct	**is**	BOOLEAN
	, VapprSel	**is**	BOOLEAN
	, VgaSel	**is**	BOOLEAN
	, VsSel	**is**	BOOLEAN

MESSAGE

This_Input_Msg

Fields:	AltPreRefChanged	**is**	BOOLEAN
	, AltselCaptureCondMet	**is**	BOOLEAN
	, AltselTargetAltChanged	**is**	BOOLEAN
	, AltselTrackCondMet	**is**	BOOLEAN
	, AltSwi	**is**	$\{\text{OFF, ON}\}_{\text{ty-117}}$
	, ApDiscSwi	**is**	$\{\text{OFF, ON}\}_{\text{ty-117}}$
	, ApEngSwi	**is**	$\{\text{OFF, ON}\}_{\text{ty-117}}$
	, ApprSwi	**is**	$\{\text{OFF, ON}\}_{\text{ty-117}}$
	, FdSwi	**is**	$\{\text{OFF, ON}\}_{\text{ty-117}}$
	, FlcSwi	**is**	$\{\text{OFF, ON}\}_{\text{ty-117}}$
	, GaSwi	**is**	$\{\text{OFF, ON}\}_{\text{ty-117}}$
	, HdgSwi	**is**	$\{\text{OFF, ON}\}_{\text{ty-117}}$
	, LapprTrackCondMet	**is**	BOOLEAN
	, NavSwi	**is**	$\{\text{OFF, ON}\}_{\text{ty-117}}$
	, NavTrackCondMet	**is**	BOOLEAN
	, Overspeed	**is**	BOOLEAN
	, SyncSwi	**is**	$\{\text{OFF, ON}\}_{\text{ty-117}}$
	, TransSwi	**is**	$\{\text{OFF, ON}\}_{\text{ty-117}}$
	, VapprTrackCondMet	**is**	BOOLEAN
	, VsPthWhlMot	**is**	BOOLEAN
	, VsSwi	**is**	$\{\text{OFF, ON}\}_{\text{ty-117}}$

INPUT INTERFACE

Other_Input

Min. Separation: UNDEFINED
Max. Separation: UNDEFINED
Input Action: READ(Other_Input_$\text{Msg}_{\text{m-152}}$)

Handler

Guard Condition: TRUE

Assignment(s)	$\text{Offside\_Alt\_Selected}_{\text{iv-149}}$	**is assigned**	AltSel
	$\text{Offside\_Altsel\_Active}_{\text{iv-149}}$	**is assigned**	AltselAct
	$\text{Offside\_Altsel\_Selected}_{\text{iv-149}}$	**is assigned**	AltselSel
	$\text{Offside\_Altsel\_Track}_{\text{iv-150}}$	**is assigned**	AltselTrk
	$\text{Offside\_FD\_On}_{\text{iv-145}}$	**is assigned**	FdOn
	$\text{Offside\_FGS\_Active}_{\text{iv-144}}$	**is assigned**	FGSActive
	$\text{Offside\_Flc\_Selected}_{\text{iv-148}}$	**is assigned**	FlcSel
	$\text{Offside\_Hdg\_Selected}_{\text{iv-146}}$	**is assigned**	HdgSel
	$\text{Offside\_Lappr\_Selected}_{\text{iv-147}}$	**is assigned**	LapprSel
	$\text{Offside\_Lappr\_Active}_{\text{iv-147}}$	**is assigned**	LapprAct
	$\text{Offside\_Lga\_Selected}_{\text{iv-147}}$	**is assigned**	LgaSel
	$\text{Offside\_Modes\_On}_{\text{iv-145}}$	**is assigned**	ModesOn
	$\text{Offside\_Nav\_Active}_{\text{iv-146}}$	**is assigned**	NavAct
	$\text{Offside\_Nav\_Selected}_{\text{iv-146}}$	**is assigned**	NavSel
	$\text{Offside\_Pitch\_Selected}_{\text{iv-148}}$	**is assigned**	PthSel
	$\text{Offside\_Roll\_Selected}_{\text{iv-145}}$	**is assigned**	RollSel
	$\text{Offside\_Vappr\_Selected}_{\text{iv-150}}$	**is assigned**	VapprSel
	$\text{Offside\_Vappr\_Active}_{\text{iv-150}}$	**is assigned**	VapprAct
	$\text{Offside\_Vga\_Selected}_{\text{iv-151}}$	**is assigned**	VgaSel
	$\text{Offside\_Vs\_Selected}_{\text{iv-148}}$	**is assigned**	VsSel

INPUT INTERFACE

This_Input

Min. Separation: UNDEFINED
Max. Separation: UNDEFINED
Input Action: READ(This_Input_Msg$_{m\text{-}152}$)

Handler

Guard Condition: TRUE

Assignment(s)

Is_ALTSEL_Capture_Cond_Met$_{iv\text{-}101}$	**is assigned**	AltselCaptureCondMet
ALTSEL_Target_Altitude_Changed$_{iv\text{-}129}$	**is assigned**	AltselTargetAltChanged
Is_ALTSEL_Track_Cond_Met$_{iv\text{-}101}$	**is assigned**	AltselTrackCondMet
ALT_Switch$_{iv\text{-}121}$	**is assigned**	AltSwi
AP_Disconnect_Switch$_{iv\text{-}126}$	**is assigned**	ApDiscSwi
AP_Engage_Switch$_{iv\text{-}126}$	**is assigned**	ApEngSwi
APPR_Switch$_{iv\text{-}123}$	**is assigned**	ApprSwi
FD_Switch$_{iv\text{-}118}$	**is assigned**	FdSwi
FLC_Switch$_{iv\text{-}122}$	**is assigned**	FlcSwi
GA_Switch$_{iv\text{-}124}$	**is assigned**	GaSwi
HDG_Switch$_{iv\text{-}118}$	**is assigned**	HdgSwi
Is_LAPPR_Track_Cond_Met$_{iv\text{-}101}$	**is assigned**	LapprTrackCondMet
NAV_Switch$_{iv\text{-}119}$	**is assigned**	NavSwi
Is_NAV_Track_Cond_Met$_{iv\text{-}100}$	**is assigned**	NavTrackCondMet
Overspeed$_{iv\text{-}128}$	**is assigned**	Overspeed$_{iv\text{-}128}$
SYNC_Switch$_{iv\text{-}125}$	**is assigned**	SyncSwi
Transfer_Switch$_{iv\text{-}125}$	**is assigned**	TransSwi
Is_VAPPR_Track_Cond_Met$_{iv\text{-}102}$	**is assigned**	VapprTrackCondMet
VS_Pitch_Wheel_In_Motion$_{iv\text{-}125}$	**is assigned**	VsPthWhlMot
VS_Switch$_{iv\text{-}120}$	**is assigned**	VsSwi

2.16 FGS Outputs

This section defines the physical interface for all outputs from the FGS. The output variables associated with these fields are defined in the part of the specification to which they are logically related.

MESSAGE

This_Output_Msg

Fields:	AltLamp	**is**	$\{\text{OFF, ON}\}_{\text{ty-118}}$
	, AltSel	**is**	BOOLEAN
	, AltselAct	**is**	BOOLEAN
	, AltselSel	**is**	BOOLEAN
	, AltselTrk	**is**	BOOLEAN
	, ApEng	**is**	BOOLEAN
	, ApLamp	**is**	$\{\text{OFF, ON}\}_{\text{ty-118}}$
	, ApprLamp	**is**	$\{\text{OFF, ON}\}_{\text{ty-118}}$
	, FdOn	**is**	BOOLEAN
	, FGSActive	**is**	BOOLEAN
	, FlcLamp	**is**	$\{\text{OFF, ON}\}_{\text{ty-118}}$
	, FlcSel	**is**	BOOLEAN
	, HdgLamp	**is**	$\{\text{OFF, ON}\}_{\text{ty-118}}$
	, HdgSel	**is**	BOOLEAN
	, LapprAct	**is**	BOOLEAN
	, LapprSel	**is**	BOOLEAN
	, LgaSel	**is**	BOOLEAN
	, ModesOn	**is**	BOOLEAN
	, NavAct	**is**	BOOLEAN
	, NavLamp	**is**	$\{\text{OFF, ON}\}_{\text{ty-118}}$
	, NavSel	**is**	BOOLEAN
	, PilotFlying	**is**	$\{\text{LEFT, RIGHT}\}_{\text{ty-23}}$
	, PthSel	**is**	BOOLEAN
	, RollSel	**is**	BOOLEAN
	, VapprAct	**is**	BOOLEAN
	, VapprSel	**is**	BOOLEAN
	, VgaSel	**is**	BOOLEAN
	, VsLamp	**is**	$\{\text{OFF, ON}\}_{\text{ty-118}}$
	, VsSel	**is**	BOOLEAN

MACRO

When_Lateral_Mode_Changed

Condition:

		OR						
	Changed(Is_ROLL_Selected$_{v\text{-}38}$)	T	·	·	·	·	·	·
	Changed(Is_HDG_Selected$_{v\text{-}41}$)	·	T	·	·	·	·	·
AND	Changed(Is_NAV_Selected$_{v\text{-}46}$)	·	·	T	·	·	·	·
AND	Changed(Is_NAV_Active$_{v\text{-}46}$)	·	·	·	T	·	·	·
AND	Changed(Is_LAPPR_Selected$_{v\text{-}52}$)	·	·	·	·	T	·	·
AND	Changed(Is_LAPPR_Active$_{v\text{-}52}$)	·	·	·	·	·	T	·
AND	Changed(Is_LGA_Selected$_{v\text{-}57}$)	·	·	·	·	·	·	T

MACRO

When_Vertical_Mode_Changed

Condition:

		OR									
	Changed(Is_PITCH_Selected$_{v\text{-}63}$)	T	·	·	·	·	·	·	·	·	·
	Changed(Is_VS_Selected$_{v\text{-}66}$)	·	T	·	·	·	·	·	·	·	·
	Changed(Is_ALT_Selected$_{v\text{-}74}$)	·	·	T	·	·	·	·	·	·	·
	Changed(Is_ALTSEL_Selected$_{v\text{-}79}$)	·	·	·	T	·	·	·	·	·	·
AND	Changed(Is_ALTSEL_Active$_{v\text{-}80}$)	·	·	·	·	T	·	·	·	·	·
AND	Changed(Is_ALTSEL_Track$_{v\text{-}80}$)	·	·	·	·	·	T	·	·	·	·
	Changed(Is_FLC_Selected$_{v\text{-}70}$)	·	·	·	·	·	·	T	·	·	·
	Changed(Is_VAPPR_Selected$_{v\text{-}87}$)	·	·	·	·	·	·	·	T	·	·
	Changed(Is_VAPPR_Active$_{v\text{-}87}$)	·	·	·	·	·	·	·	·	T	·
	Changed(Is_VGA_Selected$_{v\text{-}92}$)	·	·	·	·	·	·	·	·	·	T

MACRO

When_Lamp_Changed

Condition:

		OR							
	Changed(AP_Lamp$_{\text{v-126}}$)	T	·	·	·	·	·	·	·
	Changed(APPR_Lamp$_{\text{v-124}}$)	·	T	·	·	·	·	·	·
	Changed(HDG_Lamp$_{\text{v-119}}$)	·	·	T	·	·	·	·	·
AND	Changed(NAV_Lamp$_{\text{v-120}}$)	·	·	·	T	·	·	·	·
	Changed(VS_Lamp$_{\text{v-121}}$)	·	·	·	·	T	·	·	·
	Changed(ALT_Lamp$_{\text{v-122}}$)	·	·	·	·	·	T	·	·
	Changed(FLC_Lamp$_{\text{v-123}}$)	·	·	·	·	·	·	T	·
	Changed(APPR_Lamp$_{\text{v-124}}$)	·	·	·	·	·	·	·	T

OUTPUT INTERFACE

This_Output

Min. Separation: UNDEFINED
Max. Separation: UNDEFINED
Output Action: PUBLISH(This_Output_Msg$_{m\text{-}156}$)

Handler
Guard Condition:

		OR							
	Changed(Mode_Annunciations_On$_{v\text{-}33}$)	T	·	·	·	·	·	·	·
	Changed(Pilot_Flying$_{v\text{-}29}$)	·	T	·	·	·	·	·	·
	Changed(Onside_FD_On$_{v\text{-}28}$)	·	·	T	·	·	·	·	·
AND	Changed(Is_This_Side_Active$_{v\text{-}30}$)	·	·	·	T	·	·	·	·
	Changed(Is_AP_Engaged$_{v\text{-}131}$)	·	·	·	·	T	·	·	·
	When_Lateral_Mode_Changed$_{m\text{-}157}$()	·	·	·	·	·	T	·	·
	When_Vertical_Mode_Changed$_{m\text{-}157}$()	·	·	·	·	·	·	T	·
	When_Lamp_Changed$_{m\text{-}158}$()	·	·	·	·	·	·	·	T

Assignment(s)	AltLamp	**is assigned**	$\text{ALT\_Lamp}_{\text{v-122}}$
	AltSel	**is assigned**	$\text{Is\_ALT\_Selected}_{\text{v-74}}$
	AltselAct	**is assigned**	$\text{Is\_ALTSEL\_Active}_{\text{v-80}}$
	AltselSel	**is assigned**	$\text{Is\_ALTSEL\_Selected}_{\text{v-79}}$
	AltselTrk	**is assigned**	$\text{Is\_ALTSEL\_Track}_{\text{v-80}}$
	ApEng	**is assigned**	$\text{Is\_AP\_Engaged}_{\text{v-131}}$
	ApLamp	**is assigned**	$\text{AP\_Lamp}_{\text{v-126}}$
	ApprLamp	**is assigned**	$\text{APPR\_Lamp}_{\text{v-124}}$
	FdOn	**is assigned**	$\text{Onside\_FD\_On}_{\text{v-28}}$
	FGSActive	**is assigned**	$\text{Is\_This\_Side\_Active}_{\text{v-30}}$
	FlcLamp	**is assigned**	$\text{FLC\_Lamp}_{\text{v-123}}$
	FlcSel	**is assigned**	$\text{Is\_FLC\_Selected}_{\text{v-70}}$
	HdgLamp	**is assigned**	$\text{HDG\_Lamp}_{\text{v-119}}$
	HdgSel	**is assigned**	$\text{Is\_HDG\_Selected}_{\text{v-41}}$
	LapprSel	**is assigned**	$\text{Is\_LAPPR\_Selected}_{\text{v-52}}$
	LapprAct	**is assigned**	$\text{Is\_LAPPR\_Active}_{\text{v-52}}$
	LgaSel	**is assigned**	$\text{Is\_LGA\_Selected}_{\text{v-57}}$
	ModesOn	**is assigned**	$\text{Mode\_Annunciations\_On}_{\text{v-33}}$
	NavAct	**is assigned**	$\text{Is\_NAV\_Active}_{\text{v-46}}$
	NavLamp	**is assigned**	$\text{NAV\_Lamp}_{\text{v-120}}$
	NavSel	**is assigned**	$\text{Is\_NAV\_Selected}_{\text{v-46}}$
	PilotFlying	**is assigned**	$\text{Pilot\_Flying}_{\text{v-29}}$
	PthSel	**is assigned**	$\text{Is\_PITCH\_Selected}_{\text{v-63}}$
	RollSel	**is assigned**	$\text{Is\_ROLL\_Selected}_{\text{v-38}}$
	VapprSel	**is assigned**	$\text{Is\_VAPPR\_Selected}_{\text{v-87}}$
	VapprAct	**is assigned**	$\text{Is\_VAPPR\_Active}_{\text{v-87}}$
	VgaSel	**is assigned**	$\text{Is\_VGA\_Selected}_{\text{v-92}}$
	VsLamp	**is assigned**	$\text{VS\_Lamp}_{\text{v-121}}$
	VsSel	**is assigned**	$\text{Is\_VS\_Selected}_{\text{v-66}}$

Action: SEND

Bibliography

[1] Anonymous. Mission critical systems: Defense attempting to address major software challenges. Technical Report GAO/IMTEC-93-13, U.S. General Accounting Office, 1992.

[2] Anonymous. Joint advisory circular: Flight guidance system approval. Technical Report AC/ACJ 25.1329, Federal Aviation Administration, 2001.

[3] Barry Boehm. Prentice-Hall, Englewood Cliffs, NJ, 1981.

[4] Fred Brooks. No silver bullet: : Essence and accidents of software engineering. *IEEE Computer*, pages 10–19, April 1997.

[5] David Harel and Amnon Naamad. The STATEMATE semantics of statecharts. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 5(4):293–333, October 1996.

[6] Matts Heimdahl and Nancy G. Leveson. Completeness and consistency in hierarchical state-based requirements. *IEEE Transactions on Software Engineering*, 22(6):363–377, June 1996.

[7] Constance L. Heitmeyer, James Kirby, , and Bruce G. Labaw. Automated consistency checking of requirements specification. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 5(3):231–261, July 1996.

[8] Nancy Leveson, Matts Heimdahl, Holly Hildreth, and Jon Reese. Requirements specifications for process-control systems. *IEEE Transactions on Software Engineering*, 20(9):684–707, September 1994.

[9] Nancy Leveson, Denise Pinnel, Sean Sandys, Shuichi Koga, and Jon Reese. Analyzing software specifications for mode confusion potential. 1997.

[10] Nancy Leveson, Sean Sandys, Denise Pinnel, Susan Joslyn, Liliana Alfaro, Zelda Zabinsky, and Alan Shaw. Safety analysis of air traffic control upgrades. Technical report, Safeware Engineering, 1997.

[11] Nancy G. Leveson. Addison-Wesley Publishing Company, Reading, Massachusetts, 1995.

[12] Robyn Lutz. Analyzing software requirements errors in safety-critical, embedded systems. In *IEEE Symposium on Requirements Engineering*, San Diego, CA, 1993.

[13] Steven P. Miller. Specifying the mode logic of a flight guidance system in core and scr. In *Second Workshop on Formal Methods in Software Practice (FMSP98)*, Clearwater Beach, Florida, 1998.

[14] E. H. Phillip. Global teamwork called key to improving aviation safety. *Aviation Week & Space Technology*, page 70, July 2001.

Index

ALTSEL (State Variable)
Definition, 82
Reference(s), 79, 83, 84
ALT (State Variable)
Definition, 76
Reference(s), 74, 75
ALT_Lamp (State Variable)
Definition, 122
Reference(s), 158, 160
ALT_Switch (Input Variable)
Definition, 121
Reference(s), 112, 155
ALTSEL_Active (State Variable)
Definition, 84
Reference(s), 80
ALTSEL_Selected (State Variable)
Definition, 83
Reference(s), 80, 81, 84
ALTSEL_Target_Altitude_Changed (Input Variable)
Definition, 129
Reference(s), 113, 155
AP_Disconnect_Switch (Input Variable)
Definition, 126
Reference(s), 106, 155
AP_Engage_Switch (Input Variable)
Definition, 126
Reference(s), 105, 155
AP_Lamp (State Variable)
Definition, 126
Reference(s), 158, 160
AP_State (Type)
Definition, 131
Reference(s), 132
APPR_Lamp (State Variable)
Definition, 124
Reference(s), 158, 160
APPR_Switch (Input Variable)
Definition, 123
Reference(s), 114, 155
Activate_LAPPR (Macro)
Definition, 50
Reference(s), 53, 55
Activate_VAPPR (Macro)
Definition, 85
Reference(s), 88, 90
Activate_NAV (Macro)
Definition, 44
Reference(s), 47, 49
Active_State (Type)
Definition, 23
Reference(s), 84, 142
AP (State Variable)
Definition, 132
Reference(s), 25, 32, 56, 91, 131

Base_State (Type)
Definition, 23
Reference(s), 39, 43, 48, 54, 59, 64, 68, 72, 76, 82, 89, 94, 136–141, 143, 144
Boolean
Reference, 28, 30, 33, 38, 41, 46, 52, 57, 63, 66, 70, 71, 74, 79, 80, 87, 92, 100–102, 125, 127–129, 131, 134, 135, 144–153, 156

Capture_ALTSEL (Macro)
Definition, 78
Reference(s), 81, 83

Deactivate_ALTSEL (Macro)
Definition, 79
Reference(s), 83
Deactivate_LAPPR (Macro)
Definition, 51
Reference(s), 54
Deactivate_VAPPR (Macro)
Definition, 86
Reference(s), 89

Deactivate_NAV (Macro)
Definition, 45
Reference(s), 48
Dearm_LAPPR (Macro)
Definition, 51
Reference(s), 54
Dearm_NAV (Macro)
Definition, 45
Reference(s), 48
Deselect_ALTSEL (Macro)
Definition, 78
Reference(s), 82
Deselect_LAPPR (Macro)
Definition, 51
Reference(s), 54
Deselect_PITCH (Macro)
Definition, 62
Reference(s), 64
Deselect_ROLL (Macro)
Definition, 37
Reference(s), 39
Deselect_VAPPR (Macro)
Definition, 86
Reference(s), 89
Deselect_ALT (Macro)
Definition, 73
Reference(s), 76
Deselect_FLC (Macro)
Definition, 70
Reference(s), 72
Deselect_HDG (Macro)
Definition, 40
Reference(s), 43
Deselect_LGA (Macro)
Definition, 56
Reference(s), 59
Deselect_NAV (Macro)
Definition, 45
Reference(s), 48
Deselect_VGA (Macro)
Definition, 91
Reference(s), 94
Deselect_VS (Macro)
Definition, 65
Reference(s), 68

FD_Switch (Input Variable)
Definition, 118
Reference(s), 103, 155
FLC (State Variable)
Definition, 72
Reference(s), 70, 71
FLC_Lamp (State Variable)
Definition, 123
Reference(s), 158, 160
FLC_Switch (Input Variable)
Definition, 122
Reference(s), 111, 155

GA_Switch (Input Variable)
Definition, 124
Reference(s), 115, 155

HDG (State Variable)
Definition, 43
Reference(s), 41, 42
HDG_Lamp (State Variable)
Definition, 119
Reference(s), 158, 160
HDG_Switch (Input Variable)
Definition, 118
Reference(s), 108, 155

Independent_Mode (State Variable)
Definition, 31
Reference(s), 30
Independent_Mode_Condition (Macro)
Definition, 31
Reference(s), 31
Is_ALT_Active (State Variable)
Definition, 74
Reference(s), 61, 69, 77, 78
Is_ALT_Selected (State Variable)
Definition, 74
Reference(s), 122, 157, 160
Is_ALTSEL_Active (State Variable)
Definition, 80
Reference(s), 27, 61, 69, 157, 160
Is_ALTSEL_Capture_Cond_Met (Input Variable)
Definition, 101
Reference(s), 97, 155
Is_ALTSEL_Selected (State Variable)
Definition, 79
Reference(s), 157, 160
Is_ALTSEL_Track (State Variable)
Definition, 80
Reference(s), 73, 157, 160

Is_ALTSEL_Track_Cond_Met (Input Variable)
Definition, 101
Reference(s), 96, 155
Is_AP_Engaged (State Variable)
Definition, 131
Reference(s), 32, 126, 159, 160
Is_FLC_Active (State Variable)
Definition, 71
Reference(s), 61
Is_FLC_Selected (State Variable)
Definition, 70
Reference(s), 123, 157, 160
Is_HDG_Active (State Variable)
Definition, 41
Reference(s), 36
Is_HDG_Selected (State Variable)
Definition, 41
Reference(s), 119, 157, 160
Is_LAPPR_Active (State Variable)
Definition, 52
Reference(s), 31, 36, 85, 157, 160
Is_LAPPR_Selected (State Variable)
Definition, 52
Reference(s), 86, 124, 157, 160
Is_LAPPR_Track_Cond_Met (Input Variable)
Definition, 101
Reference(s), 98, 155
Is_LGA_Active (State Variable)
Definition, 57
Reference(s), 36
Is_LGA_Selected (State Variable)
Definition, 57
Reference(s), 157, 160
Is_NAV_Active (State Variable)
Definition, 46
Reference(s), 36, 157, 160
Is_NAV_Selected (State Variable)
Definition, 46
Reference(s), 120, 157, 160
Is_NAV_Track_Cond_Met (Input Variable)
Definition, 100
Reference(s), 95, 155
Is_No_Nonbasic_Lateral_Mode_Active (Macro)
Definition, 36
Reference(s), 37
Is_No_Nonbasic_Vertical_Mode_Active (Macro)
Definition, 61
Reference(s), 62
Is_Offside_LAPPR_Active (State Variable)
Definition, 134
Reference(s), 31
Is_Offside_VAPPR_Active (State Variable)
Definition, 134
Reference(s), 31
Is_Offside_VGA_Active (State Variable)
Definition, 135
Reference(s), 31
Is_PITCH_Active (State Variable)
Definition, 63
Is_PITCH_Selected (State Variable)
Definition, 63
Reference(s), 157, 160
Is_ROLL_Active (State Variable)
Definition, 38
Is_ROLL_Selected (State Variable)
Definition, 38
Reference(s), 157, 160
Is_This_Side_Active (State Variable)
Definition, 30
Reference(s), 26, 27, 34, 39, 42, 43, 47–49, 53–55, 58, 59, 64, 67, 68, 71, 72, 75, 76, 81–84, 88–90, 93, 94, 159, 160
Is_VAPPR_Active (State Variable)
Definition, 87
Reference(s), 27, 31, 61, 65, 69, 73, 77, 78, 157, 160
Is_VAPPR_Selected (State Variable)
Definition, 87
Reference(s), 124, 157, 160
Is_VAPPR_Track_Cond_Met (Input Variable)
Definition, 102
Reference(s), 99, 155
Is_VGA_Active (State Variable)
Definition, 92
Reference(s), 31, 61, 77, 78
Is_VGA_Selected (State Variable)
Definition, 92
Reference(s), 157, 160
Is_VS_Active (State Variable)
Definition, 66
Reference(s), 27, 61
Is_VS_Selected (State Variable)
Definition, 66
Reference(s), 121, 157, 160

LAPPR (State Variable)

Definition, 54
Reference(s), 52, 53, 55
LAPPR_Selected (State Variable)
Definition, 55
Reference(s), 52–54
LGA (State Variable)
Definition, 59
Reference(s), 57, 58
Lamp (Type)
Definition, 118
Reference(s), 119–124, 126, 156

Mode_Annunciations_On (State Variable)
Definition, 33
Reference(s), 25–27, 159, 160
Modes (State Variable)
Definition, 34
Reference(s), 33, 37, 39, 40, 43, 45, 48, 49, 51, 54–56, 59, 62, 64, 65, 68, 70, 72, 73, 76–78, 82–84, 86, 89–91, 94

NAV (State Variable)
Definition, 48
Reference(s), 46, 47, 49
NAV_Lamp (State Variable)
Definition, 120
Reference(s), 158, 160
NAV_Selected (State Variable)
Definition, 49
Reference(s), 46–48
NAV_Switch (Input Variable)
Definition, 119
Reference(s), 107, 155
No_Higher_Event_Than_ALT_Switch_Pressed (Macro)
Definition, 113
Reference(s), 112, 113
No_Higher_Event_Than_ALTSEL_Capture_Cond_Met (Macro)
Definition, 98
Reference(s), 97
No_Higher_Event_Than_ALTSEL_Target_Altitude_Changed (Macro)
Definition, 114
Reference(s), 113, 114
No_Higher_Event_Than_ALTSEL_Track_Cond_Met (Macro)
Definition, 97
Reference(s), 96
No_Higher_Event_Than_AP_Disconnect_Switch_Pressed (Macro)
Definition, 107
Reference(s), 106, 107
No_Higher_Event_Than_AP_Engage_Switch_Pressed (Macro)
Definition, 106
Reference(s), 105, 106
No_Higher_Event_Than_APPR_Switch_Pressed (Macro)
Definition, 115
Reference(s), 109, 114, 115
No_Higher_Event_Than_FD_Switch_Pressed (Macro)
Definition, 104
Reference(s), 96–100, 103
No_Higher_Event_Than_FLC_Switch_Pressed (Macro)
Definition, 112
Reference(s), 111, 112
No_Higher_Event_Than_GA_Switch_Pressed (Macro)
Definition, 116
Reference(s), 115, 116
No_Higher_Event_Than_HDG_Switch_Pressed (Macro)
Definition, 109
Reference(s), 108, 109
No_Higher_Event_Than_LAPPR_Track_Cond_Met (Macro)
Definition, 99
Reference(s), 98
No_Higher_Event_Than_NAV_Switch_Pressed (Macro)
Definition, 108
Reference(s), 105, 108
No_Higher_Event_Than_NAV_Track_Cond_Met (Macro)
Definition, 96
Reference(s), 95

No_Higher_Event_Than_SYNC_Switch_Pressed (Macro)
Definition, 117
Reference(s), 116, 117
No_Higher_Event_Than_Transfer_Switch_Pressed (Macro)
Definition, 105
Reference(s), 104
No_Higher_Event_Than_VAPPR_Track_Cond_Met (Macro)
Definition, 100
Reference(s), 99
No_Higher_Event_Than_VS_Pitch_Wheel_Rotated (Macro)
Definition, 110
Reference(s), 105, 110
No_Higher_Event_Than_VS_Switch_Pressed (Macro)
Definition, 111
Reference(s), 110, 111

Offside_ALTSEL (State Variable)
Definition, 141
Reference(s), 82, 142
Offside_LAPPR (State Variable)
Definition, 138
Reference(s), 54, 138
Offside_PITCH (State Variable)
Definition, 139
Reference(s), 64
Offside_ROLL (State Variable)
Definition, 136
Reference(s), 39
Offside_VAPPR (State Variable)
Definition, 143
Reference(s), 89, 143
Offside_ALT (State Variable)
Definition, 141
Reference(s), 76
Offside_Alt_Selected (Input Variable)
Definition, 149
Reference(s), 141, 154
Offside_Altsel_Active (Input Variable)
Definition, 149
Reference(s), 142, 154
Offside_ALTSEL_Active (State Variable)
Definition, 142
Reference(s), 84
Offside_Altsel_Selected (Input Variable)
Definition, 149
Reference(s), 141, 154
Offside_ALTSEL_Selected (State Variable)
Definition, 142
Reference(s), 83, 142
Offside_Altsel_Track (Input Variable)
Definition, 150
Reference(s), 142, 154
Offside_FLC (State Variable)
Definition, 140
Reference(s), 72
Offside_FD (State Variable)
Definition, 133
Reference(s), 32
Offside_FD_On (Input Variable)
Definition, 145
Reference(s), 133, 154
Offside_FGS_Active (Input Variable)
Definition, 144
Reference(s), 154
Offside_Flc_Selected (Input Variable)
Definition, 148
Reference(s), 140, 154
Offside_HDG (State Variable)
Definition, 136
Reference(s), 43
Offside_Hdg_Selected (Input Variable)
Definition, 146
Reference(s), 136, 154
Offside_LGA (State Variable)
Definition, 139
Reference(s), 59
Offside_Lappr_Active (Input Variable)
Definition, 147
Reference(s), 138, 154
Offside_Lappr_Selected (Input Variable)
Definition, 147
Reference(s), 138, 154
Offside_LAPPR_Selected (State Variable)
Definition, 138
Reference(s), 55, 134
Offside_Lga_Selected (Input Variable)
Definition, 147
Reference(s), 139, 154
Offside_Modes (State Variable)

Definition, 135
Reference(s), 30, 34, 136–144
Offside_Modes_On (Input Variable)
Definition, 145
Reference(s), 135, 154
Offside_NAV (State Variable)
Definition, 137
Reference(s), 48, 137
Offside_Nav_Active (Input Variable)
Definition, 146
Reference(s), 137, 154
Offside_Nav_Selected (Input Variable)
Definition, 146
Reference(s), 137, 154
Offside_NAV_Selected (State Variable)
Definition, 137
Reference(s), 49
Offside_Pitch_Selected (Input Variable)
Definition, 148
Reference(s), 139, 154
Offside_Roll_Selected (Input Variable)
Definition, 145
Reference(s), 136, 154
Offside_VGA (State Variable)
Definition, 144
Reference(s), 94, 135
Offside_Vappr_Active (Input Variable)
Definition, 150
Reference(s), 143, 154
Offside_Vappr_Selected (Input Variable)
Definition, 150
Reference(s), 143, 154
Offside_VAPPR_Selected (State Variable)
Definition, 143
Reference(s), 90, 134
Offside_Vga_Selected (Input Variable)
Definition, 151
Reference(s), 144, 154
Offside_VS (State Variable)
Definition, 140
Reference(s), 68
Offside_Vs_Selected (Input Variable)
Definition, 148
Reference(s), 140, 154
On_Off (Type)
Definition, 23
Reference(s), 28, 31, 34, 133, 135
Onside_FD (State Variable)
Definition, 28
Reference(s), 28, 32
Onside_FD_On (State Variable)
Definition, 28
Reference(s), 159, 160
Other_Input (Input Interface)
Definition, 153
Other_Input_Msg (Message)
Definition, 152
Reference(s), 153
Overspeed (Input Variable)
Definition, 128
Reference(s), 27, 128, 155
Overspeed_Condition (Macro)
Definition, 128
Reference(s), 25, 26, 56, 65, 69, 70, 85, 91

PITCH (State Variable)
Definition, 64
Reference(s), 63
Pilot_Flying (State Variable)
Definition, 29
Reference(s), 25, 29, 30, 159, 160

ROLL (State Variable)
Definition, 39
Reference(s), 38

SYNC_Switch (Input Variable)
Definition, 125
Reference(s), 116, 155
Select_ALTSEL (Macro)
Definition, 77
Reference(s), 82
Select_LAPPR (Macro)
Definition, 50
Reference(s), 53, 54
Select_PITCH (Macro)
Definition, 62
Reference(s), 64
Select_ROLL (Macro)
Definition, 37
Reference(s), 39
Select_VAPPR (Macro)
Definition, 85
Reference(s), 89
Select_ALT (Macro)
Definition, 73
Reference(s), 75, 76

Select_FLC (Macro)
Definition, 69
Reference(s), 71, 72
Select_HDG (Macro)
Definition, 40
Reference(s), 42, 43
Select_LGA (Macro)
Definition, 56
Reference(s), 58, 59
Select_NAV (Macro)
Definition, 44
Reference(s), 47, 48
Select_VGA (Macro)
Definition, 91
Reference(s), 93, 94
Select_VS (Macro)
Definition, 65
Reference(s), 67, 68
Selected_State (Type)
Definition, 23
Reference(s), 49, 55, 83, 90, 137, 138, 142, 143
Side (Type)
Definition, 23
Reference(s), 29, 156
Switch (Type)
Definition, 117
Reference(s), 118–126, 153

THIS_SIDE (Constant)
Definition, 24
Reference(s), 25, 30
This_Input (Input Interface)
Definition, 155
This_Input_Msg (Message)
Definition, 152
Reference(s), 155
This_Output (Output Interface)
Definition, 159
This_Output_Msg (Message)
Definition, 156
Reference(s), 159
Track_ALTSEL (Macro)
Definition, 78
Reference(s), 84
Transfer_Switch (Input Variable)
Definition, 125
Reference(s), 104, 155

VAPPR (State Variable)
Definition, 89
Reference(s), 87, 90
VAPPR_Selected (State Variable)
Definition, 90
Reference(s), 87–89
VGA (State Variable)
Definition, 94
Reference(s), 92, 93
VS_Lamp (State Variable)
Definition, 121
Reference(s), 158, 160
VS_Pitch_Wheel_In_Motion (Input Variable)
Definition, 125
Reference(s), 109, 155
VS_Switch (Input Variable)
Definition, 120
Reference(s), 110, 155
VS (State Variable)
Definition, 68
Reference(s), 66, 67

When_ALT_Activated (Macro)
Definition, 75
Reference(s), 60, 69
When_ALT_Switch_Pressed (Macro)
Definition, 112
Reference(s), 112, 113
When_ALT_Switch_Pressed_Seen (Macro)
Definition, 113
Reference(s), 27, 73
When_ALTSEL_Activated (Macro)
Definition, 81
Reference(s), 60, 69
When_ALTSEL_Capture_Cond_Met (Macro)
Definition, 97
Reference(s), 97
When_ALTSEL_Capture_Cond_Met_Seen (Macro)
Definition, 97
Reference(s), 78
When_ALTSEL_Target_Altitude_Changed (Macro)
Definition, 113
Reference(s), 113, 114
When_ALTSEL_Target_Altitude_Changed_Seen (Macro)
Definition, 114

Reference(s), 27, 73, 79
When_ALTSEL_Track_Cond_Met (Macro)
Definition, 96
Reference(s), 96
When_ALTSEL_Track_Cond_Met_Seen (Macro)
Definition, 96
Reference(s), 78
When_AP_Disconnect_Switch_Pressed (Macro)
Definition, 106
Reference(s), 106, 107
When_AP_Disconnect_Switch_Pressed_Seen (Macro)
Definition, 107
Reference(s), 130
When_AP_Engage_Switch_Pressed (Macro)
Definition, 105
Reference(s), 105, 106
When_AP_Engage_Switch_Pressed_Seen (Macro)
Definition, 106
Reference(s), 130
When_APPR_Switch_Pressed (Macro)
Definition, 114
Reference(s), 109, 114, 115
When_APPR_Switch_Pressed_Seen (Macro)
Definition, 115
Reference(s), 26, 27, 50, 51, 85, 86
When_Disengage_AP (Macro)
Definition, 130
Reference(s), 132
When_Engage_AP (Macro)
Definition, 130
Reference(s), 132
When_FD_Switch_Pressed (Macro)
Definition, 103
Reference(s), 96–100, 103
When_FD_Switch_Pressed_Seen (Macro)
Definition, 103
Reference(s), 25, 26
When_FLC_Activated (Macro)
Definition, 71
Reference(s), 60
When_FLC_Switch_Pressed (Macro)
Definition, 111
Reference(s), 111, 112
When_FLC_Switch_Pressed_Seen (Macro)
Definition, 112
Reference(s), 27, 69, 70
When_GA_Switch_Pressed (Macro)
Definition, 115
Reference(s), 115, 116
When_GA_Switch_Pressed_Seen (Macro)
Definition, 116
Reference(s), 26, 27, 56, 91, 130
When_HDG_Activated (Macro)
Definition, 42
Reference(s), 35
When_HDG_Switch_Pressed (Macro)
Definition, 108
Reference(s), 108, 109
When_HDG_Switch_Pressed_Seen (Macro)
Definition, 109
Reference(s), 26, 40
When_Lamp_Changed (Macro)
Definition, 158
Reference(s), 159
When_LAPPR_Activated (Macro)
Definition, 53
Reference(s), 35
When_LAPPR_Armed (Macro)
Definition, 53
Reference(s), 45
When_LAPPR_Track_Cond_Met (Macro)
Definition, 98
Reference(s), 98
When_LAPPR_Track_Cond_Met_Seen (Macro)
Definition, 98
Reference(s), 50
When_Lateral_Mode_Changed (Macro)
Definition, 157
Reference(s), 159
When_Lateral_Mode_Manually_Selected (Macro)
Definition, 26
Reference(s), 25
When_LGA_Activated (Macro)
Definition, 58
Reference(s), 35
When_NAV_Activated (Macro)
Definition, 47
Reference(s), 35
When_NAV_Armed (Macro)
Definition, 47
Reference(s), 51
When_NAV_Switch_Pressed (Macro)
Definition, 107
Reference(s), 105, 108
When_NAV_Switch_Pressed_Seen (Macro)

Definition, 108
Reference(s), 26, 44, 45
When_NAV_Track_Cond_Met (Macro)
Definition, 95
Reference(s), 95
When_NAV_Track_Cond_Met_Seen (Macro)
Definition, 95
Reference(s), 44
When_Nonbasic_Lateral_Mode_Activated (Macro)
Definition, 35
Reference(s), 37, 40, 45, 51, 56, 91
When_Nonbasic_Vertical_Mode_Activated (Macro)
Definition, 60
Reference(s), 56, 62, 65, 70, 73, 79, 86, 91
When_Pilot_Flying_Transfer (Macro)
Definition, 29
Reference(s), 25, 40, 45, 51, 56, 65, 70, 73, 79, 86, 91
When_Selected_Nav_Frequency_Changed (State Variable)
Definition, 127
Reference(s), 45, 51, 86
When_Selected_Nav_Source_Changed (State Variable)
Definition, 127
Reference(s), 45, 51, 86
When_SYNC_Switch_Pressed (Macro)
Definition, 116
Reference(s), 116, 117
When_SYNC_Switch_Pressed_Seen (Macro)
Definition, 117
Reference(s), 56, 91
When_Transfer_Switch_Pressed (Macro)
Definition, 104
Reference(s), 104
When_Transfer_Switch_Pressed_Seen (Macro)
Definition, 104
Reference(s), 29
When_Turn_FD_Off (Macro)
Definition, 26
Reference(s), 28
When_Turn_FD_On (Macro)
Definition, 25
Reference(s), 28
When_Turn_Modes_Off (Macro)
Definition, 32
Reference(s), 34
When_Turn_Modes_On (Macro)
Definition, 32
Reference(s), 34
When_VAPPR_Activated (Macro)
Definition, 88
Reference(s), 60
When_VAPPR_Track_Cond_Met (Macro)
Definition, 99
Reference(s), 99
When_VAPPR_Track_Cond_Met_Seen (Macro)
Definition, 99
Reference(s), 85
When_Vertical_Mode_Changed (Macro)
Definition, 157
Reference(s), 159
When_Vertical_Mode_Manually_Selected (Macro)
Definition, 27
Reference(s), 25
When_VGA_Activated (Macro)
Definition, 93
Reference(s), 60
When_VS_Activated (Macro)
Definition, 67
Reference(s), 60
When_VS_Pitch_Wheel_Rotated (Macro)
Definition, 109
Reference(s), 105, 110
When_VS_Pitch_Wheel_Rotated_Seen (Macro)
Definition, 110
Reference(s), 27, 70, 73, 79, 91
When_VS_Switch_Pressed (Macro)
Definition, 110
Reference(s), 110, 111
When_VS_Switch_Pressed_Seen (Macro)
Definition, 111
Reference(s), 27, 65

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01- 06 - 2003	Contractor Report	12/2000-5/2003

4. TITLE AND SUBTITLE
Flight Guidance System Requirements Specification

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Steven P. Miller
Alan C. Tribble
Timothy M. Carlson
Eric J. Danielson

5d. PROJECT NUMBER
NCC1-01001

5e. TASK NUMBER

5f. WORK UNIT NUMBER
728-30-10-03

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)
NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NASA/CR-2003-212426

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 61
Availability: NASA CASI (301) 621-0390 Distribution: Standard

13. SUPPLEMENTARY NOTES
An electronic version can be found at http://techreports.larc.nasa.gov/ltrs/ or http://ntrs.nasa.gov
Langley Technical Monitor: Ricky W. Butler

14. ABSTRACT

This report describes a requirements specification written in the RSML-e language for the mode logic of a Flight Guidance System of a typical regional jet aircraft. This model was created as one of the first steps in a five-year project sponsored by the NASA Langley Research Center, Rockwell Collins Inc., and the Critical Systems Research Group of the University of Minnesota to develop new methods and tools to improve the safety of avionics designs. This model will be used to demonstrate the application of a variety of methods and techniques, including safety analysis of system and subsystem requirements, verification of key properties using theorem provers and model checkers, identification of potential sources mode confusion in system designs, partitioning of applications based on the criticality of system hazards, and autogeneration of avionics quality code. While this model is representative of the mode logic of a typical regional jet aircraft, it does not describe an actual or planned product. Several aspects of a full Flight Guidance System, such as recovery from failed sensors, have been omitted, and no claims are made regarding the accuracy or completeness of this specification.

15. SUBJECT TERMS
formal methods, requirements, mode confusion, flight guidance system, avionics, formal analysis, flight software

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	176	(301) 621-0390

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18