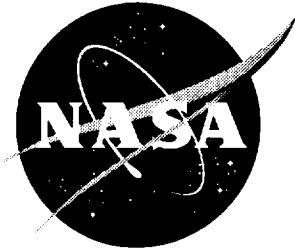


NASA/CR-1998-208446



# Integrated Display System For Low Visibility Landing and Surface Operations

*Sharon Otero Beskenis, David F. Green, Jr., Paul V. Hyer and Edward J. Johnson, Jr.  
Lockheed Martin Engineering & Sciences, Hampton, Virginia*

National Aeronautics and  
Space Administration

Langley Research Center  
Hampton, Virginia 23681-2199

Prepared for Langley Research Center  
under Contract NAS1-96014

---

July 1998

---

Available from the following:

NASA Center for Aerospace Information (CASI)  
800 Elkridge Landing Road  
Linthicum Heights, MD 21090-2934  
(301) 621-0390

National Technical Information Service (NTIS)  
5285 Port Royal Road  
Springfield, VA 22161-2171  
(703) 487-4650

# CONTENTS

CONTENTS .....	III
<b>1.0 INTRODUCTION .....</b>	<b>1</b>
1.1 OVERVIEW .....	1
1.2 GENERAL REQUIREMENTS .....	3
<b>2.0 SYSTEM ARCHITECTURE AND COMPONENTS.....</b>	<b>5</b>
2.1 HARDWARE ARCHITECTURE AND COMPONENTS.....	5
2.1.1 <i>B-757 Flight Hardware and Components</i> .....	5
2.1.2 <i>VMS Simulation Hardware and Components</i> .....	7
2.2 SOFTWARE ARCHITECTURE AND COMPONENTS .....	8
<b>3.0 TECHNICAL APPROACH .....</b>	<b>10</b>
3.1 DISPLAY PROGRAM.....	10
3.1.1 <i>Taxi Displays</i> .....	12
3.1.2 <i>Roll Out and Turn Off Displays (ROTO)</i> .....	14
3.1.2.1 <i>Default ROTO Display</i> .....	16
3.1.2.2 <i>Alternate ROTO Display</i> .....	17
3.1.3 <i>Flight Displays</i> .....	17
3.1.3.1 <i>Approach HUD Display</i> .....	19
3.1.3.2 <i>Flight HUD Display</i> .....	20
3.1.3.3 <i>Go-Around Display</i> .....	20
3.1.4 <i>Takeoff Display</i> .....	21
3.1.5 <i>Runway Exit Plan-View Display</i> .....	21
3.1.6 <i>Display Calibration</i> .....	23
3.1.7 <i>Databases</i> .....	23
3.1.8 <i>Conversion of Latitude and Longitude Data</i> .....	24
3.1.9 <i>Pilot Input</i> .....	25
3.1.10 <i>Controller-Pilot Message Communications</i> .....	26
3.1.10.1 <i>Message Manager</i> .....	26
3.1.11 <i>Taxi Routing and Monitoring</i> .....	28
3.1.11.1 <i>Dynamic Taxi Routing</i> .....	28
3.1.11.2 <i>ATC Hold Short Instructions (Hold Bars)</i> .....	29
3.1.11.3 <i>Taxi Monitor</i> .....	29
3.2 REAL-TIME DATA COMMUNICATIONS MODULE .....	30
3.2.1 <i>Datablock Design</i> .....	30
3.2.2 <i>Device Drivers and Communications Programs</i> .....	31
3.2.3 <i>Playback Programs</i> .....	32
3.2.4 <i>View Programs</i> .....	32
3.3 CONTROL PROGRAM .....	32
<b>4.0 CODE VALIDATION .....</b>	<b>33</b>
<b>5.0 LESSONS LEARNED .....</b>	<b>34</b>
<b>ACKNOWLEDGMENTS .....</b>	<b>39</b>
<b>REFERENCES.....</b>	<b>39</b>
<b>APPENDIX.....</b>	<b>40</b>
<b>FIGURES .....</b>	<b>44</b>



# **INTEGRATED DISPLAY SYSTEM FOR LOW VISIBILITY LANDING AND SURFACE OPERATIONS**

## **SUMMARY REPORT**

### **1.0 INTRODUCTION**

This report summarizes the software products, system architectures, and operational procedures developed by Lockheed Martin in support of the Low Visibility Landing and Surface Operations (LVLASO) program at the NASA Langley Research Center. The work was performed in support of program milestones to develop and flight test a set of prototype flight deck displays that provide guidance and enhanced situational awareness to a flight crew during all operational phases of landing, roll-out, turn-off, inbound taxi, outbound taxi, and takeoff [1] [2]. This report provides an overview of the technical aspects, capabilities, and system integration issues associated with the LVLASO Integrated Display System (IDS), which collects, processes and presents information to the flight crew. A series of technical publications that provide more complete information on individual components of the IDS will be published as separate contractor reports.

#### **1.1 Overview**

LVLASO is the sub-element within NASA's Terminal Area Productivity program that is responsible for identifying, developing, and demonstrating technologies that can safely maintain clear weather capacities on the airport surface in low visibility conditions. As part of this charter, the LVLASO program established a research and flight test program aimed at providing critical information to the flight crew during aircraft taxi and landing operations in low visibility. The approach was to integrate data from both ground-based and airborne technologies using digital datalinks, and to develop symbologies for displaying this information on aircraft flight deck displays. The system fused information on aircraft location from a Differential Global Positioning System (DGPS), air traffic control messages, airport traffic data from Airport Surface Detection Equipment (ASDE-3) radar, aircraft identities from the Airport Traffic Identification System (ATIDS), runway status information from the Airport Movement Area Safety System (AMASS), and real-time information from aircraft flight management computers with a database containing the physical layout of the airport. Information was presented to the flight crew using a liquid crystal head-down display (HDD) and/or a head-up display (HUD) mounted at the captain's station. NASA's Boeing 757 research aircraft housed the airborne components of the system.

For taxi operations, both displays were used. An electronic moving map of the airport was implemented on the HDD, which included the airport layout, runway status information, the locations and identities of all known surface traffic, the taxi route assigned by air traffic control

(ATC), and textual transcriptions of ATC communications. The flight crew could select any one of five zoom levels to focus on the area immediately around the test aircraft or view the entire airport. An electronic taxi route, displayed on the HUD during taxi operations, presented a symbolic path that conformed to the actual locations of taxiways. Text information describing the current aircraft location within the assigned taxi route was also displayed.

During the landing and roll-out and turn-off (ROTO) phases of flight, the LVLASO system provided the captain with deceleration guidance to an exit taxiway selected by the flight crew. At the discretion of the flight crew, the exit could be manually selected using a rotary switch in the cockpit or determined automatically by the flight software using current aircraft data from the flight management computer. In addition to a full suite of "typical" HUD symbology for navigation and control of the aircraft, the IDS included additional symbology for in-air and on-ground operations. A symbolic runway and exit taxiway, which was conformal with the actual runway and taxiway, was displayed on the HUD during approach and landing. After touchdown, symbology was shown that provided aircraft deceleration guidance, additional information about the exit taxiway, and the predicted speed of the aircraft at the exit. HUD symbology was programmed to automatically de-clutter and seamlessly transition from flight through landing, ROTO, and into the display format for taxi. This sequence was reversed for takeoff. During level flight, the flight crew could remove the symbolic runway and exit information by de-selecting the instrument landing system (ILS) frequency.

Information processed by the IDS for the flight crew was received through a shared memory interface and local area network. Onboard NASA's B757 research aircraft, the data were transmitted over a SCRAMNet<sup>+</sup> (Shared Common Random Access Memory Network) communications ring and distributed to specific memory locations, in real time, by software developed for the IDS. For testing purposes, synthetic data from ground-based facilities and aircraft systems could be generated and placed in shared memory. The IDS code operated on this data just as though it had been sent from a ground station over a datalink or retrieved from the onboard systems. Additional software was developed to permit a real-time operator interaction with memory contents at selected locations. This control program was used during flight operations to change optional features of the IDS code without stopping the displays. Software to view the contents of the shared memory locations, in real time, was also developed and used extensively to verify data on the communications ring during flight and testing operations. Code debugging was aided by software which captured the shared memory contents to a data twenty times per second. Real-time performance could be simulated by "playing back" this file at the appropriate rate.

The IDS was successfully flight tested on board NASA's 757 research aircraft and demonstrated to guests during two deployments to the Atlanta Hartsfield International Airport in August of 1997. The components which comprised this successful research program were developed through close collaboration between NASA, the FAA, avionics manufacturers, academic researchers, and Lockheed Martin Engineering and Sciences. This report addresses work done by Lockheed Martin to develop, test, and verify the IDS in support of this research. Work done by other members of the research team will be published separately.

## 1.2 General Requirements

Development of the IDS began from a set of general requirements developed by NASA engineers for the overall experimental system. Commercial head-up and head-down cockpit display technology would be used concurrently to provide the flight crew with improved situational awareness during approach, landing, deceleration, and all surface taxi operations. The HUD would show airborne navigational guidance, aircraft deceleration cues to a selected exit following landing, and taxi route guidance during ground operations. The HDD would provide situational awareness information specifically for ground taxi operations, including information on airport surface traffic and a text messaging system for ATC transmissions. Components of the experimental system would include an interface with the aircraft flight management computers, an interface with the flight crew, a real-time data system, digital datalinks for transmission of positional and traffic information, databases containing airport information, a taxi routing system, text message system, and a means to control and monitor the displays without stopping them.

Lockheed Martin was tasked with developing the hardware and software architecture, communication interfaces, and computer graphics necessary to concurrently drive both displays with information received from an array of external sources. Final specifications for the IDS evolved as the project matured and responses to pilot evaluations were implemented. The Taxiway Navigation and Situation Awareness (T-NASA) format developed by the NASA Ames Research Center was specified as the baseline taxi display format for both the HDD and HUD. This format defined symbology for the taxi route HUD display, moving map HDD display, ground traffic symbols on the HDD, and specified a database containing the locations of runway and taxiway center lines. The baseline T-NASA format made no provision for the text messaging system, traffic tags, runway status indicators, or ROTO operations required for the LVLASO experimental system. Those features were added to the IDS by Lockheed Martin as part of the final specifications. The IDS supported a fully integrated display concept in which display elements automatically and seamlessly transitioned between operational phases.

The IDS text messaging system uses a ground-based voice recognition system to translate ATC instructions into digitized text messages for transmission to the NASA 757 research aircraft. The IDS was required to receive, display, and operate on the content of these messages. Since taxi routes were sent and/or modified through the text messaging system in real time, an algorithm to dynamically determine the appropriate taxi route was also required. The dynamic routing algorithm was required to recognize an ATC taxi route instruction and determine, based on the current aircraft location, if the route was appropriate and respond accordingly. The IDS was required to interpret messages containing hold short instructions, and draw a "hold bar" on the airport moving map at the appropriate location. Runway status was treated in a similar fashion. The IDS received information on active runways from ground facilities and automatically placed hold bars at the appropriate runway and taxiway intersections depicted on the map. A capability to monitor the progress of the research aircraft along the assigned taxi route, and notify controllers of deviations was also required. Whenever possible, identification tags were assigned to the symbols depicting the location of all known surface traffic on the electronic moving map. Air carrier flight numbers, aircraft registration numbers or vehicle numbers were attached to traffic symbols associated with transponder equipped aircraft or vehicles.

Initial designs for the LVLASO experiment system specified that the HUD would be used to support two basic test modes. During roll-out and turn-off (ROTO), symbology on the HUD would provide guidance to assist the captain in decelerating the aircraft and exiting the runway on a pre-selected taxiway at a pre-selected speed. Following ROTO operations or for ground taxi operations, the HUD would show taxi guidance in the T-NASA format. The IDS was required to be configurable for automatic transitions between these modes of operation or for taxi only operations. Both modes of operation involved HUD symbology depicting the runway or taxiway, which was required to be conformal with actual locations on the airport surface.

The LVLASO experimental design specified a HUD-based guidance system, for ROTO operations, which blended data from the aircraft flight management computers with data from a differential global positioning system to locate the aircraft in a database of the airport layout and determine aircraft performance parameters. The system was required to display the runway and exit taxiway selected by the pilot, provide guidance to the exit relative to a pre-selected profile, continuously monitor aircraft deceleration for exceeded parameters, and possess both manual and automatic modes of operation. The system was also required to support multiple deceleration profiles, multiple ROTO guidance formats and to seamlessly transition to the T-NASA taxi HUD format for on-ground operations. ROTO functions of the LVLASO experimental program and the ROTO symbology were to be developed at LaRC based on specifications jointly determined by NASA engineers and Lockheed Martin personnel. Specific requirements for the ROTO system evolved, as capabilities matured, into a group of HUD displays and algorithms which provided guidance through approach, landing, roll-out, takeoff, cruise, and go-around flight phases.

In-air symbology was required on the HUD to demonstrate a natural transition from the landing approach into ROTO deceleration guidance. In-air symbology used on the HUD was typical of that found on commercial units. The in-air symbology was overlaid with T-NASA ground symbology during the landing approach. While landing, the IDS was required to automatically de-clutter the HUD and transition to a ROTO display for deceleration. Symbol sets required for this transition and those for the takeoff and flight displays were derivatives of the ROTO on-ground and in-air symbology. Displays appropriate for go-around flight operations and a HUD display showing a plan view of the selected runway with all possible ROTO exits depicted were also required.

The LVLASO experiment design specified that a single rotary switch on the Pilot Input Device (PID) would be used by the pilots to control the exit selection process, and place the ROTO system in manual or automatic exit selection mode. In manual mode, the ROTO system was required to calculate and display guidance to the pilots' chosen exit based on the current deceleration algorithm. If the pilot changed the selected exit, new deceleration guidance was required using the current aircraft location and speed. In automatic mode, deceleration limits established for passenger comfort were used by the ROTO system to automatically select an exit taxiway. During the landing and roll-out, the ROTO system was required to continuously monitor the aircraft deceleration. If a deceleration exceeding allowable limits was necessary to reach the selected exit speed, the ROTO system was required to automatically sequence forward to the next ROTO exit taxiway and generate a new deceleration profile.



## 2.0 SYSTEM ARCHITECTURE AND COMPONENTS

In the following section, the hardware architecture utilized by LVLASO-IDS is discussed. Particular emphasis is placed on the aircraft hardware architecture. A number of configurations were utilized in the simulators. The architecture used once the simulation migrated from the Transport Systems Research Vehicle (TSRV) simulator to the Visual Motion Simulator (VMS) is discussed.

### 2.1 Hardware Architecture and Components

For operations aboard NASA's B-757 aircraft, the LVLASO-IDS hardware architecture used dedicated computers to drive the HUD and HDD. This configuration was selected after reviewing the computational and graphics capabilities of the available flight hardened computers, the video outputs required to drive the two displays, and the onboard communications requirements. Code development and pilot training operations in LaRC flight simulation facilities utilized a different hardware configuration. Both configurations are briefly discussed below.

#### 2.1.1 B-757 Flight Hardware and Components

The following hardware was installed on the NASA B-757 research aircraft as part of the LVLASO experimental systems. All of the hardware, which was contained in the Indigo2 and Iris pallets, was tested in the Experimental Aircraft Systems Integration Laboratory (EASILY) prior to installation on the B-757. Figure 1 depicts the flight hardware architecture.

**Silicon Graphics, Inc. (SGI) Computers:** Two flight hardened SGI computers, an Indigo2 and a Personal Iris (PI), both running the IRIX 5.3 operating system were selected as the flight computers. The Indigo2 computer hosted IDS display software configured to drive the HDD and communications software to support a SCRAMNet+ interface. The Personal Iris (PI) computer hosted display software configured to drive the HUD and SCRAMNet+ communications software.

**Pilot Input Device (PID)** - The PID panel (figure 2) containing switches, buttons, knobs, and lights was installed in the flight deck center aisle stand. Its purpose was to facilitate pilot control of the displays. Lights on the device alerted pilots when ATC messages were received. The PID was designed jointly between LaRC and ARC, and was constructed at LaRC.

**SCRAMNet+ communications and networking hardware:** A SCRAMNet+ (Shared Common Random Access Memory Network) communications network developed by SYSTRAN Corp. was used for B-757 onboard communications among the two SGIs, the Data Acquisition System (DAS), and the I/O concentrator, which acquired data from several sources and placed it on the communications ring. This hardware was based on a replicated shared memory concept. Fiber optic connections were used between nodes on the ring.

**DAT drive:** A 4mm Digital Audio Tape (DAT) drive connected via a Small Computer System Interface (SCSI) to the PI provided the capability to load new software and store recorded flight data for playback, debugging, and post analysis in the LVLASO development lab.

**Ethernet communications and networking hardware:** A local Ethernet communications network utilizing TCP/IP (Transmission Control Protocol/Internet Protocol) communications protocols was employed for onboard communications among the two SGIs. Recorded data were transferred to the PI for archival, and new software was loaded onto the Indigo2 via the DAT drive on the PI. Data transfers occurred at a maximum data rate of 10Mbits/sec via a twisted pair connection. Transceivers were used to adapt between the thicknet connector on the two machines to the thinnet local area network.

**Video distribution amplifiers:** The video outputs of the PI and the Indigo2 were fed into video distribution amplifiers. Those units then distributed the RGB video outputs to the HDD or HUD, to the computer monitor, and to the aircraft video system. The outputs were converted to the correct video formats for transmission throughout the aircraft for selectable viewing of the video sources, recording of mixed video and audio sources, signal monitoring and conditioning, and video telemetry to the ground station for viewing by guest observers.

**Collins interface unit:** The Collins unit converted the composite video outputs of the Indigo2 to an analog optical signal to drive the HDD. The signal output by the Indigo2 was a 60 Hz. non-interlaced, 813 line format using the Green output as the source of sync. The resolution of this format, 1024x768 pixels, necessitated the use of the Maximum Impact graphics hardware option.

**Terrabit interface units:** A Terrabit raster input unit and a Terrabit interface unit, Evans and Sutherland Model No. SR101-285, were used to take the composite raster video outputs of the Personal Iris and convert them to an X, Y, Z stroke video format. The video format was sent to the overhead HUD unit, which returned a flyback trigger signal to the Terrabit for synchronization. The overhead unit converted the X, Y, Z signals into video images and projected them onto a combining glass for the pilot's viewing. The signal output by the PI was RS343, a 30 Hz. interlaced, 1023 line format with the Green output as the source of sync with resolution of 1280x960 pixels.

**HDD -** The HDD (figure 3), a high resolution, sunlight readable liquid crystal display (LCD), was developed by Rockwell-Collins. The LCD, which was used to display the moving map, was suspended under the B-757 glareshield, left of center, in portrait orientation.

**HUD -** A Flight Dynamics HUD projector and combiner glass (figure 3) was mounted at the captain's station in the B-757. It employed holographic optics to create symbology effectively focused at infinity.

**Flight Hardened PCs:** Flight hardened PCs connected to the serial ports of the Indigo2 and the PI served as terminals for operator input. The terminals were used to run view software discussed in section 3.2, or control software discussed in section 3.3.

The hardware components described above were individually flight hardened, installed, and tested in EASILY. The Indigo2 and PI computers, upon installation in their respective pallets, were checked for correct operation with the Aydin monitors, specialized cabling and connectors, and flight certified power connections. Local and Wide Area Net Ethernet capabilities, terminal interfaces, the DAT drive, the SCRAMNet+ cards and fiber optic connections, and the PID were confirmed for full functionality.

Component video testing was performed in EASILY. NTSC and RS343 video output signals from the PI and a 1024x768 video output signal from the Indigo2 were measured using 100% white field for horizontal frequency, horizontal front porch duration, horizontal sink, and horizontal back porch duration. Those measurements confirmed the signal clarity both from the

SGI source computers and as an output of the video distribution amplifiers. Proving that the SGI hosts met video specs was an important procedure since it enabled isolation of poor video signal quality on the HDD or the HUD to the interface units or the display devices themselves. Based on good results with RS343, the higher resolution, higher line rate video signal was chosen over NTSC for use by the HUD. Extensive coordination with Collins and Flight Dynamics, the suppliers of the HDD and HUD units, enabled proper diagnosis and adjustments to the interface units or display units to correct problems. Testing with the video pallets for selectable viewing of the video sources and video telemetry was performed. Both display systems functioned very well during the flight tests and demonstrations in Atlanta.

Upon installation of the hardware on the aircraft, power testing was performed to ensure that all equipment operated within specs with no deviations or spikes utilizing ground, APU (Auxiliary Power Unit), and engine power. EMI (Electromagnetic Interference) testing was performed to ensure that the hardware components did not interfere with other aircraft systems such as the Autoland, the ILS, or the communications signals. Correct operation of SCRAMNet+, Ethernet, the terminals, the DAT drive, the PID, and the video display systems was confirmed under all power sources on the aircraft. Spare components were assembled and tested on the aircraft prior to deployment to Atlanta. No spares were utilized during the flight tests and demonstration. "Preflight Procedures", "Power Up Procedures", and "Shutdown Procedures" were written and used to confirm correct hardware power sequencing and operation on the B-757 research aircraft.

### ***2.1.2 VMS Simulation Hardware and Components***

The following hardware was utilized in the Visual Motion Simulator (VMS) as part of the LVLASO simulation systems.

**Silicon Graphics, Inc. (SGI) Computer:** An Onyx computer running the IRIX 5.3 operating system hosted IDS display software configured to drive the HUD and communications software to support a DR11W interface.

**DR11W communications:** A DR11W-A communications card developed by VME Microsystems International Corp. (VMIC) provided a high performance, 32-bit Direct Memory Access (DMA) interface that formed a 16-bit parallel interface between the VMEbus and DEC DR11-W compatible devices. The device driver was modified accordingly to communicate with the Computer Automated Measurement and Control (CAMAC) serial highway in simulation.

**Sun Microsystems, Inc. Computer:** A Sun computer utilizing a network interface to the Onyx machine was used for operator input. Sun windows on the Onyx were used to run the IDS display software and control software discussed in section 3.3.

**ROTO Knob** - A modified preferred exit knob with a subset of the positions available on the B-757 research aircraft PID was installed in the flight deck center aisle stand. Its purpose was to facilitate pilot selection of the desired runway exit for ROTO.

**RGB Spectrums video mixer:** The video mixer mixed the composite video outputs of the Onyx with the Evans and Sutherland generated out-the-window scene of the Atlanta airport to simulate

the look of a HUD. The signal output by the Onyx was a 30 Hz. interlaced, 1023 line format using the Green output as the source of sync. The video output had a resolution of 1280x960 pixels, which matched the resolution of the out-the window displays in the VMS.

## 2.2 Software Architecture and Components

The high level software architecture for the LVLASO IDS is depicted in figure 4. This architecture has been developed to maximize performance and to incorporate all LVLASO requirements for integration of experimental systems on board the NASA B-757 research aircraft. The primary software design concept of the architecture made use of multiple parallel processes (i.e., programs) and a shared memory interface to pass data and communicate among processes. This design offered several advantages over a single process architecture. The principal advantage was that all processes in the IDS executed independently and shared the same data without the need for synchronization. As a result, the communications program handled all I/O operations, supplying continuous real-time data through the shared memory interface to the display program at varying rates and from many different sources. Graphics performance was greatly enhanced since the display program was freed from I/O bottlenecks, and could run at its maximum frame rate. The display program did not have to include low level functions that dealt with the source of the data or the type of communications interface. Other processes such as the control and view programs shared the same data and provided additional IDS capabilities without affecting I/O and graphics performance. The flexibility of the architecture allowed processes to be run alone or together, and the number of concurrent processes could be incremented as needed to add functionality for different requirements.

The architecture proved to be highly adaptable for implementation in LaRC's flight simulation facilities as well as on the NASA B-757 research aircraft. In simulation, the architecture was implemented on a single computer that executed the display and communications programs on separate processors of the computer. The display program was configured to run the ROTO HUD, which was superimposed on the out-the-window scene in the simulator. The communications program received simulated real-time data through a DR11W interface, and made the data available to the display program through the shared memory interface.

On board the NASA B-757, the architecture was implemented on two separate computers, which executed the display programs for the HUD and HDD in the flight deck. The communications program ran on both machines and received real-time data through interfaces to a SCRAMNet+ communications ring (see section 2.1). The dual implementation of the architecture allowed data to be received simultaneously for both the HUD and HDD through the shared memory interfaces on both computers.

The IDS software architecture included the following major software components:

**Data Communications Programs:** Several communications programs were developed to interface with different hardware devices on the research aircraft and in the flight simulator, and provide varied capabilities for data recording and playback (see section 3.2). The real-time communications program provided the interfaces to the communications hardware via the device drivers (SCRAMNet+ or DR11W), read real-time data from the device, and wrote the data into the shared memory for use by the display process. For LVLASO on board the B-757, the program also read from the shared memory and wrote to SCRAMNet+ memory in specific cases such as sending downlink messages. A parallel process sharing the virtual address space of the

parent communications program provided the recording of data at the appropriate data rate. Another communications program, called the View program, ran concurrently and provided the capability to monitor all data in real time. The playback program enabled replay of the real-time data at the appropriate data rate. The replay could pause and be made interactive via the PID. The real-time communications program and device drivers were similar but not the same due to the hardware constraints of the Indigo2 and PI.

**Display Program:** The display program read real-time data from shared memory and used the data to drive the graphics display for either the HDD or HUD, depending on software configuration. The program also wrote to shared memory when sending downlink messages to the LVLASO test controller (see section 3.1.10). The HDD moving map display software was executed on the Indigo2 and the HUD software was executed on the PI (see section 2.1 above). The display program was identical on both machines; however, the software configuration utility was used to configure each program to drive a specific display (see section 3.1).

**Databases:** Several types of databases were developed for use by the display program (see section 3.1.7). Visual databases were developed for the Atlanta airport and Langley Air Force Base using a MultiGen modeling tool. The Atlanta airport database was based on data from Jeppesen Sanderson, Inc., which provided center and edge lines for all taxiways and runways to a accuracy of plus or minus one foot. The Langley database was based on local surveys. Other databases were developed to provide information on taxiway/runway segments and intersections, location of hold bars, taxi routing and other data requirements including runway and exit parameters for the ROTO display.

**Control Program:** The control program used shared memory to communicate with the display and communications processes. The purpose of this program during the LVLASO experiment was to provide a means for operator input (as opposed to pilot input) in order to control parameters of the experiment including ROTO algorithms, criteria for taxi only operations, canned taxi route input, and data recording. The program also provided a mechanism to simulate inputs from different sources during development testing.

### **3.0 TECHNICAL APPROACH**

This section describes the technical approach for specific elements of the IDS developed by Lockheed Martin to meet LVLASO requirements. More detailed technical information for selected elements is provided in other referenced NASA publications or documents.

Software for the IDS was developed on Silicon Graphics computers using the C programming language, OpenGL graphics language, and X-Windows to facilitate future portability to other platforms. The design used an object-based approach allowing the individual software modules to function independently. Code was written separately for four individual software modules - the data communications module, taxi display module, ROTO display module, and the control program module. Codes for the taxi and ROTO display modules were integrated and compiled as a single program. The communications module consisted of a number of separate programs, including various hardware dependent data communications and monitoring programs and several versions of data recording and playback programs used for specialized purposes. The control program module was written as a single program that was generic for any hardware or software configuration. All programs ran concurrently and communicated through the shared memory interface as discussed above. General procedures, capabilities, and technical descriptions of the software for displays, data communications, and control will be discussed in the following subsections.

#### **3.1 Display Program**

The display software was designed to provide flight deck displays for all phases of ground and flight operations, and smooth transitions of the displays between all phases. During roll-out and runway exit, the Roll-Out Turn-Off (ROTO) displays were activated on the HUD. During taxi operations, the taxi displays were activated on the HUD and HDD. During takeoff, flight, approach and landing phases, standard flight symbologies were presented on the HUD for transitioning to or from the ROTO HUD displays.

**Software Configuration:** The display program in the IDS was designed as a single program with options to run any type of display (HDD or HUD), any combination of data type (manual keyboard and mouse or real time), different display monitors, and several other options. To set up the display program for these options prior to execution, a software configuration utility was developed using a menu driven windows graphical user interface (GUI). The windows GUI made use of the Silicon Graphics buttonfly program, which provided a set of software push button menus to select program options. Software configuration allowed the display program software to be the same regardless of the computer it was running on, the display it was executing, or the monitor it was using. Using the configuration utility was a convenient method of setting up multiple input and display modes without having to enter complicated command lines. The following options were available:

- Type of display - HDD map or HUD.

- Type of HUD to start - Taxi or ROTO.

- Type of display device - SGI monitor or Collins LCD.

- Input run mode - real-time input or manual/mouse input.

Airport initialization parameters.

Network mode - client or server (not used).

Name of project (not used).

Review options selected.

Program structure: The top level program structure and main loop for the displays program was provided by the function main(), which performed the following sequential tasks:

#### Initialize startup

- Read command line input

- Attach to shared memory

- Read airport databases into program memory

- Read/initialize the software startup configuration

  - (Set type of display, monitor, real time or mouse, etc.)

- Initialize graphics and miscellaneous variables

- Open and start log file

#### Main loop - run continuously until terminated

- Get real-time data and user input

- Draw graphics for specific display

- Get frame time

#### Program termination

- Detach from shared memory

- Close log file

- Exit

Each task in main() was performed by an appropriate function or task manager. For example, the input tasks in the main loop were performed by various functions within the input manager depending on the software configuration (i.e., HDD or HUD, taxi or ROTO, manual or real-time). The graphics functions were executed by the displays manager depending on which type of display the software was configured to draw. Data processing and computations common to all displays were accomplished by the input manager, and computations for a specific display were done by the graphics functions.

The graphics functions were broken down into two main program areas: displays used for taxi operations and displays used for ROTO, takeoff, and flight operations. Taxi displays represented a combination of work done at ARC and LaRC, and are discussed in section 3.1.1. The ROTO, takeoff and flight displays were developed solely at LaRC and are discussed in section 3.1.2 through section 3.1.6.

### **3.1.1 Taxi Displays**

**T-NASA display format:** The T-NASA display format and visual database specifications developed at ARC were used as the baseline for the taxi displays. The original T-NASA software utilized separate programs for the HDD map and the HUD. Those programs were redesigned and consolidated into a single display program to allow sharing of all common functions. Modifications to the T-NASA code and additional software were required for integration with the IDS architecture and LVLASO experimental system to accept a high volume of real-time data with different data rates and formats, and to add other features and capabilities not available in T-NASA. However, the basic T-NASA display format and database capabilities were not changed. The T-NASA system has been well documented in other publications [3] [4]. T-NASA display features used in the IDS are briefly summarized as follows:

**Taxi Map HDD** A moving map of the airport surface (figures 5, 6) included: the airport runways, taxiways, ramps, grass areas and buildings; a perspective view of the airport at various zoom levels and a plan view of the entire airport; runway/taxiway labels; icon (triangle) for current position of the research aircraft; arc or wedge in front of the triangle icon to highlight the critical area in front of the aircraft; digital heading at top of map; positions of other traffic (circles); and compass heading around the border of the map.

**Taxi HUD** A three dimensional scene-linked symbology display (figure 7) overlaid the out-the-window scene and included: centerline markings; edge cones to indicate a defined distance of 43 feet from either side of the centerline; signs to indicate the location and angle of turns; ground speed; and current, previous and next taxiway indicators.

**Additional taxi displays:** T-NASA was modified or supplemented at LaRC to provide the following additional display features (see figures 5 and 6):

**Display Device (LCD) Interface:** To support portrait orientation of the Rockwell-Collins LCD in the flight deck, modifications to the graphics code and new fonts were required to allow a rotation of 90 degrees for all HDD graphics and text. The pixel dimensions of the HDD map and the relative sizes of text displays were also changed to fit the exact size of the 1024x768 LCD.

**AMASS Runway Status Warnings:** Airport Movement Area Safety System (AMASS) information was used to warn of oncoming traffic on occupied (hot) runways. Red hold lines (called AMASS hold bars) were drawn at the intersections of taxiways and runways to warn when aircraft were landing or taking off, making it unsafe to enter or cross the runway. The AMASS hold bars were removed after the oncoming aircraft passed the intersection and no longer posed a threat. The AMASS runway status data were a part of the LVLASO experimental system used during the Atlanta flight tests. The visual format on the HDD seen by the pilot was the same as the display seen by Atlanta controllers in the control tower. The AMASS runway status warnings were very effective in alerting pilots of current activity on all runways.

The AMASS software in the IDS included a bit-mapped data format for input, a database for AMASS hold bar coordinates, data structures for storing data in memory, and a function to draw the graphics on the HDD. The bit-mapped data format was defined as one 32 bit word for each runway on the airport, where each bit of a word represented one or more intersections of taxiways with the runway. The AMASS database developed for Atlanta conformed with this format, and included four runways and the coordinates for



all intersections on each runway. The formatted bit pattern and corresponding database were unique for Atlanta. However, the software, which included the AMASS data structures and drawing function, were generic for any airport. The function used the bit pattern and intersection coordinates defined in the AMASS database to draw one or more hold bars for each bit that was set by the real-time input data.

**Traffic Tags:** Traffic tags were uplinked along with the positional traffic data and were shown in a 3-dimensional format. The tags consisted of alpha-numeric text for the flight number or tail number of the aircraft, whichever was available. The text was drawn at the end of vertical lines (or tag poles) that extended upward from the center of the traffic symbols (circles) in the Z-axis. The tags rotated to remain horizontal as the traffic symbols moved to preserve readability. If a tag was not available, a dummy string was sent and an empty tag pole (i.e., no text at the end of the vertical pole) was displayed. The ownship traffic data, identified by the tag "NASA557", were filtered to prevent drawing the ownship symbol twice, since ownship was already drawn from the DGPS data.

**Traffic Not Available Warnings:** If traffic data were not available for five seconds or longer, all traffic symbols were removed and a warning message, "TRAFFIC NOT AVAILABLE", was drawn in large red letters at the bottom of the map display. This warning message was necessary to distinguish between a drop out of traffic data and a simple lack of traffic. The method used to determine non-availability was based on the target scan count (i.e., count of radar scans). The count was updated (normally once a second) for each new radar scan of traffic data. If the scan count was not updated for five seconds, all traffic was removed from the HDD because their positions were considered unreliable.

**ATC Messages - Display, Scrolling and Acknowledgment:** ATC messages were displayed in a scrollable message area at the bottom of the HDD in the order in which they were received. The scroll area contained up to three messages at any one time; and the last six messages could be scrolled backward or forward by the pilot using the PID. The most recent message was printed in blue and all previous messages were yellow. The pilot was notified of a new message when the LED light on the PID was turned on, and also when the text of the new message flashed in the message scroll area on the HDD. The pilot acknowledged receipt of a new message by pressing the CONFIRM button on the PID. Later messages received were not displayed until all earlier ones had been acknowledged. The LED light was turned off when all flashing messages had been acknowledged via the CONFIRM button. This method of pilot acknowledgment of messages was a LVLASO requirement and was used during the first phase of LVLASO flight testing at Atlanta. During the second phase of testing, the procedure was changed so that all messages were acknowledged automatically by the software. The LED lights on the PID were not used and the pilot did not have to press the CONFIRM button to acknowledge messages.

All messaging capabilities mentioned above were handled in software by message management functions of the IDS. Individual functions managed the tasks of receiving and parsing messages by type, controlling the PID lights, and placing messages in queues used by drawing routines for display, scrolling and acknowledgment (see section 3.1.10).

**Taxi Route Clearance Display:** Current ATC taxi route clearance instructions and current location were displayed in text format on the HDD in an area just above the

message scroll area. Each leg of the taxi route was shown in sequence in a separate block of the HDD clearance area. Angle brackets (“>> <<”) on either side of the leg name highlighted the leg on which the aircraft was currently located. All text information remained in the clearance area until taxi operations were completed, and were removed when the aircraft reached the end of the taxi route. The data for the clearance area were provided by the IDS taxi monitoring software (see section 3.1.11.3).

**Ownship Position Invalid Warnings:** Ownship position accuracy was dependent on availability of Differential Global Positioning System (DGPS) data and a blending of the aircraft’s Inertial Navigation System (INS) data with the DGPS data. In the event DGPS was not available, the ownship position accuracy would be degraded over time. A warning message, “DGPS INVALID”, was printed on the HDD in large flashing red letters when DGPS corrections were not received or the smoothed INS blended position was no longer accurate. That event occurred approximately 10 seconds after the loss of DGPS corrections. On the HUD, all taxi guidance symbology was removed and the message, “DGPS POSITION INVALID”, was printed. The event states for receipt/non-receipt of DGPS and accuracy of blended position were determined from error status words in the aircraft data system. The warning messages were significant since all taxi displays became unusable when the ownship position was unknown.

### ***3.1.2 Roll Out and Turn Off Displays (ROTO)***

**Display format:** After touchdown, the ROTO software module automatically transitioned the HUD symbology to display deceleration guidance to the selected exit. The guidance could be based on either of two primary concepts for presentation to the pilot. Both displays utilized a system of redundant cues that provided information in symbolic and digital formats for easy assimilation by the pilot. For both displays, the guidance information shown was updated in real time using DGPS positional information blended with data from one of three onboard Inertial Reference Units (IRU). The current Universal Time Code (UTC) provided timing data to time-filter some display elements and calculate time rates of change when necessary. Software developed for the IDS ROTO displays was constrained by the design of the experiment to produce on-ground symbology that had the “look and feel” of the T-NASA taxi HUD format and use the same airport database. To meet those objectives T-NASA concepts were re-engineered to support ROTO operations and a separate ROTO data structure containing information about the selected airport was developed. IDS displays for ROTO operations used a multiple route concept, which was based on the single route T-NASA concept, to display ground symbology for the runway and the selected exit. The display elements common to the two primary symbol sets are presented immediately below followed by discussions of the symbology unique to each display. More detailed information about this module will be published in a separate NASA Contractor Report.

**ROTO Box:** Information about the currently selected exit taxiway was displayed in an information box located on the upper right side of the HUD display. The ROTO MAN line indicated that the pilot was to manually decelerate the aircraft using guidance cues from the system. The EX and VE lines indicated the currently selected exit taxiway and the target exit speed for the exit. The DIST line indicated the available braking distance from

a nominal touchdown 1500 feet from the runway threshold. That information was read from the ROTO database each time the exit selection was changed.

**Runway Edge Markers:** A symbolic runway was drawn on the HUD in a manner such that it was conformal with the actual runway. The edges of the symbolic runway were identified with synthetic cones or “plops” in the T-NASA taxi format. The plops were located fifty feet apart along the edges of the runway. Runway width data was contained in the ROTO database. The runway was treated in a manner analogous to a single T-NASA taxi route. The ROTO module permitted up to nine exit taxiways to be associated with a runway. An indexing scheme was used to identify the route associated with the selected exit taxiway. The route was drawn on the HUD and the remainder of the runway was drawn using the runway only route. Logic was developed to eliminate ground symbology that obstructed the runway and to prevent overlapping of symbology associated with the two routes.

**Runway Centerline:** The centerline of the symbolic runway was depicted on the HUD in the T-NASA format as a series of rectangles spaced fifty feet apart. The symbolic centerline began at the runway threshold, continued down the runway until the exit taxiway, then depicted the turn from the runway onto the exit taxiway centerline. The symbolic centerline associated with the runway beyond the turn point was drawn using the runway only route.

**Taxiway Edge Markers:** During ROTO operations, exit taxiway edges were depicted on the HUD to be conformal with the actual exit taxiway edges rather than spacing them 43 feet from either side of the centerline as was done for the taxi display. The ROTO database contained the width of each ROTO exit on the airport surface. The width of the symbolic exit taxiway was appropriately reset each time the exit taxiway selection was changed.

**Flight Path Symbol:** During roll-out, the flight path object provided the reference position for the speed error tape and the acceleration deviation carrot (discussed below). While the aircraft touched down, the flight path object indicated the inertial path of the aircraft. Vertical speed, ground speed, and aircraft track angle data placed on the SCRAMNet<sup>+</sup> ring directly from the aircraft flight management computers were used to position the symbol. At a ground speed of 80 knots, the symbol was repositioned upward to indicate the local horizon and provide increased visibility of the runway environment.

**Ground Speed:** The current aircraft ground speed was displayed near the top of the HUD display, slightly left of center. The ground speed line was identified with a “G” at the start of the line. That data was available on the SCRAMNet<sup>+</sup> ring from the flight management system.

**Projected Exit Speed:** A projected speed at the selected exit was displayed directly below the ground speed. The projection was based on the current aircraft track acceleration, ground speed, and position. The data were read from the SCRAMNet<sup>+</sup> ring and the projected speed was updated continuously during the roll-out. The projected exit speed was time filtered to prevent the value from changing too rapidly for the pilot to read.

**Runway Exit Location:** The start of the turn onto the exit taxiway was marked on the HUD display with a line across the symbolic runway. The line became known as the “goal line.” Information from both the ROTO database and the airport layout database was used to determine the location of the line based on the currently selected exit taxiway.

**Football Symbol:** An oval shaped symbol was placed on the surface of the symbolic runway at a location where the aircraft would reach the targeted exit speed if the current deceleration was maintained. This symbol became known as the “football.” Near the exit, pilots could ensure a safe exit speed by holding the football at the goal line. Deceleration data were read from the SCRAMNet<sup>+</sup> ring, and the location of the football was continuously updated during the roll-out. A time filter was applied to the position calculation to prevent the football from moving too rapidly.

**Trend Vector:** A segmented trend vector was drawn on the HUD to be conformal with the outside scene and provide a visual indication of the aircraft ground track. The ends of the segments indicated the projected location of the aircraft at two and four seconds into the future.

**Runway Remaining Signs:** Symbolic runway remaining signs were drawn on either side of the symbolic runway to indicate the remaining runway in thousands of feet.

### ***3.1.2.1 Default ROTO Display***

**Display format:** Default deceleration guidance was based on computing a target deceleration speed profile from the touchdown point to the currently selected exit and providing the pilot with a visual depiction of the current speed error relative to the desired profile. Figure 8 presents an example of this display format. The target speed profile was computed using one of four numeric algorithms. NASA engineers designed the algorithms to investigate specific braking characteristics. The linear algorithm produced a target profile that reduced the aircraft speed in proportion to the distance traveled. A delayed linear algorithm permitted a short period of coasting to minimize runway occupancy time before linearly reducing the speed with distance. The nonlinear algorithm resulted in a profile that increased the required braking action towards the middle of the deceleration to mimic the “natural” deceleration profile pilots typically follow. A constant deceleration algorithm produced a target deceleration profile based on a constant rate of deceleration from main gear touchdown to the exit. The control program provided a means for operators to change the active algorithm prior to a test, if desired. The following items were added to the basic ROTO display elements:

**Speed Error:** A speed tape projecting from the wing of the flight path symbol provided a visual comparison between the current aircraft ground speed and the target ground speed of the deceleration profile. A speed error tape above the wing indicated the current speed was too high and more reverse thrust and/or braking was required to return to the profile. A speed error tape below the flight path symbol wing indicated the current ground speed was slow relative to the desired profile.

**Acceleration Deviation:** An acceleration deviation symbol, “>”, provided a visual depiction of the track acceleration relative to the selected deceleration profile. An

acceleration carrot above the wing of the flight path indicated that the current deceleration was less than the planned level and that the speed error was becoming more positive.

### ***3.1.2.2 Alternate ROTO Display***

**Display format:** The philosophy of the alternative ROTO display was to provide the pilot with a guidance error relative to the projected exit ground speed at the selected exit, rather than relative to the current target ground speed of the deceleration profile for the default ROTO display. No deceleration profile was provided as pilots slowed the aircraft according to their own discretion. The current deceleration, ground speed, and distance to the exit were used to continuously update the projected exit speed during the roll-out. Figure 9 presents an example of this display format. The following items were added to the basic ROTO display elements.

**Maximum Exit Speed Indicator:** An arrow with the word “MAX” above it was positioned on the HUD display to indicate the maximum acceptable exit speed. It was positioned above the wing of the flight path symbol by an amount proportional to the difference between the nominal exit ground speed and the maximum allowable exit speed. Any speed error tape below the MAX arrow indicated a safe exit speed would be attained if current conditions were maintained. Runway occupancy times were minimized if the pilot maintained a speed tape length between the flight path symbol wing and the MAX arrow. Both the nominal and maximum exit speeds for an exit were individually tailored for the exit and read from the ROTO database.


**Speed Error:** The length of the speed error tape indicated the difference between the projected exit ground speed and the nominal ground speed, if current deceleration were maintained. The projected exit speed, and therefore the length of the speed tape, was continuously updated during the roll-out. Deceleration and positional data were read from the SCRAMNet<sup>+</sup> communications ring.

**Acceleration Deviation:** The acceleration symbol “>” provided an additional cue which could be used to control the aircraft during the deceleration. Its location, relative to the wing of the flight path symbol, indicated the rate at which the speed error magnitude was changing. A carrot located below the wing indicated a deceleration that would drive the speed error more negative.

### ***3.1.3 Flight Displays***

**Display format:** HUD symbology originally developed at Langley Research Center [5] was selected as the baseline symbology for IDS in-air displays. This symbol set provided much of the symbology commonly found on commercial HUDs. A demonstration code was modified for use in the LVLASO experimental architecture and supplemented with additional symbols and textual information. That symbology was overlaid with the ROTO ground symbology during approach and landing. All of the aircraft status information presented on those displays was derived from the aircraft flight management system and transmitted to the IDS through the SCRAMNet<sup>+</sup> communications ring. All aircraft status information was continuously updated by the IDS in real time at a frame rate of 25 Hz. The display elements common to all of the symbol sets for flight are presented below, followed by discussions of the symbology unique to each display.

**Flight Path Symbol:** A symbol suggesting the fuselage of the aircraft with symbolic wings on either side was used to depict the actual path of the aircraft through the air. The horizontal position of that symbol relative to the nose object symbol indicated the difference between the actual track angle and the aircraft heading. A vertical displacement from the artificial horizon line indicated that the aircraft was climbing or descending. The magnitudes of those displacements were proportional to the heading difference and vertical speed.

**Nose Object:** A “” symbol was drawn at the center of the HUD display to depict the orientation of the fuselage centerline. This symbol indicated the relative pitch and roll angles of the aircraft when compared to the pitch ladder and the artificial horizon line.

**Flight Director Bars:** Freely moving horizontal and vertical bars of fixed length were drawn on the HUD display to depict pitch and roll commands from the B757 flight management computers. For the IDS, the center of the nose object was used as the reference position for the flight director bars. A horizontal bar located above the nose object indicated an increase in pitch was necessary to maintain the intended flight path. A vertical bar located left of the nose object indicated a left roll was required. The movement of the bars was driven by the flight director commands from the flight management system.

**Artificial Horizon:** An artificial horizon line was drawn on the HUD. It extended the full horizontal width of the HUD display at all times and was conformal with the position of the true local horizon when the aircraft was on the ground or near the Earth's surface. In cruise flight, that line provided a level flight reference line for the flight path symbol and/or the nose object. Due to the curvature of the Earth, the artificial horizon did not match the true horizon at flight altitudes.

**Heading Indicator:** Aircraft magnetic heading information was incorporated into the artificial horizon line. Tick marks on the upper side of the artificial horizon indicated five degree heading increments. A two digit numeric display was used to indicate ten degree heading increments. The standard practice of omitting trailing zeros was followed. The headings and tick marks scrolled into view as the aircraft yawed right or left. They also remained attached to the horizon line as it moved up or down during pitch changes.

**Roll Scale and Roll Indicator:** The roll attitude of the aircraft was indicated on a circular arc scale at the top of the HUD display. A triangular shaped pointer rotated along the arc to indicate roll angles between  $\pm 30$  degrees. Tick marks were drawn at five degree increments. Ten degree tick marks were labeled with a numeric value drawn at the periphery of the circular arc.

**Pitch Ladder:** The pitch attitude of the aircraft was indicated by the position of the nose object relative to vertical pitch ladders located on either side of the display. Pitch reference lines were drawn at five degree increments. Ten degree tick marks on the left pitch ladder were labeled with numeric values located towards the middle of the display.

Pitch ladder reference lines scrolled into view as the aircraft pitched up or down as necessary.

**Airspeed:** The calibrated airspeed was displayed in the lower left portion of the HUD display immediately above the ground speed.

**Ground Speed:** The aircraft ground speed was displayed in the lower portion of the HUD display on a line that began with the letter “G”, to delineate it from the airspeed information.

**Barometric Altitude:** The aircraft barometric altitude was displayed in the lower right portion of the HUD display. The letters “BA” followed the numeric information.

**Vertical Speed:** The aircraft vertical speed was displayed in the lower right portion of the HUD display. The letters “VS” followed the numeric information.

### ***3.1.3.1 Approach HUD Display***

**Display format:** A ROTO approach and landing was initiated when the flight crew tuned the navigation receiver to an Atlanta approach frequency. Since all runways at Atlanta had ILS front course approaches, the frequency uniquely identified the intended runway. A combination of airborne guidance, ground-based symbology, and exit information was presented during the approach. Figure 10 presents an example of the approach HUD format. The ROTO system could operate in either of two modes. In the automatic mode (Preferred Exit knob on the PID in the AUTO position), the system used an exit selection algorithm to automatically select the first suitable ROTO exit and draw the appropriate guidance information. During AUTO operations, the system automatically sequenced forward to the next ROTO exit if the exit selection criteria were violated due to a fast approach or a long landing. In the manual mode (Preferred Exit knob on the PID in any of the MAN positions), the pilot selected exit was used and automatic sequencing was disabled. The following items were added to the basic flight display elements:

**ILS Localizer Deviation Scale and Indicator:** The raw localizer error was displayed on a scale located at the bottom center of the HUD display. A diamond-shaped pointer slid along a standard five dot scale to indicate the location of the course relative to the current aircraft position. Localizer error information was determined by the flight management system. The scale and pointer were only visible if ILS capture was attained.

**ILS Glide Slope Deviation Scale and Indicator:** The raw glide slope error was displayed on a scale located at the right edge of the HUD display. A diamond-shaped pointer slid along a standard five dot scale to indicate the location of the glide path relative to the current aircraft position. Glide slope error was determined by the flight management system. The scale and pointer were only visible if ILS capture was attained.

**Radar Altitude:** The aircraft radar altitude was displayed in the lower right portion of the HUD display whenever its value was less than 2500 feet. The letter “R” followed the numeric information.

**ROTO Box:** Information about the currently selected exit taxiway was displayed in an information box located on the upper right side of the HUD display. The box indicated the current ROTO control mode, selected exit, targeted exit speed, and the nominal braking distance. That information was read from the ROTO database and updated each time the exit selection was changed.

**Runway Edge Markers with Selected Exit:** A symbolic runway, including the selected exit taxiway, was drawn on the HUD in a manner to be conformal with the actual runway and exit. Pavement edges of the runway and taxiway were identified with synthetic cones or “plops” in the T-NASA taxi format. The plops were located fifty feet apart. Width information contained in the ROTO database was dynamically accessed to readjust runway and taxiway widths as necessary.

**Runway and Taxiway Centerlines:** The centerlines of the symbolic runway and exit taxiway were also depicted on the HUD in the T-NASA format as a series of rectangles spaced fifty feet apart. The symbolic centerline began at the runway threshold, continued down the runway until the exit taxiway then depicted the turn from the runway onto the exit taxiway centerline. The symbolic centerline associated with the runway beyond the turn point was also drawn.

### ***3.1.3.2 Flight HUD Display:***

**Display format:** A HUD display suitable for cruise flight was created from the basic flight display elements. All exit information and symbology associated with ground objects was suppressed regardless of the Preferred Exit knob position. Cruise flight was assumed whenever the navigation receiver was tuned to a non-Atlanta ILS frequency. Figure 11 presents an example of this display format.

### ***3.1.3.3 Go-Around Display***

**Display format:** A limited HUD display for the go-around situation was provided by the ROTO module of the IDS. This flight phase was included in the IDS programming since go-around operations were possible during the Atlanta flight tests. The display was initiated if the flight crew pressed go-around switches on the throttles, and it remained active until the mode was terminated in the flight management system. Input from the Preferred Exit knob on the PID was ignored while the go-around mode was active. That prevented an accidental restart of the ROTO system during a critical phase of flight. Figure 12 presents an example of this display format. The format consisted of the basic flight symbology superimposed on runway edge markers, which identified the location of the runway for the pilot. The ROTO box and exit taxiway were not drawn. The following items were added to the basic flight display elements:

**Runway Edge Markers:** A symbolic runway was drawn on the HUD in a manner to be conformal with the actual runway. Pavement edges of the runway were identified with synthetic cones or “plops” in the T-NASA taxi format. The plops were located fifty feet



apart. Width information contained in the ROTO database was dynamically accessed to read the runway width.

**Runway Centerline:** The centerline of the symbolic runway was also depicted on the HUD in the T-NASA format as a series of rectangles spaced fifty feet apart. The symbolic centerline began at the threshold and continued the full length of the runway.

**Radar Altitude:** The aircraft radar altitude was displayed in the lower right portion of the HUD display whenever its value was less than 2500 feet. The letter “R” followed the numeric information.

#### ***3.1.4 Takeoff Display:***

**Display format:** To achieve the most realistic cockpit environment possible, the HUD and HDD operated continuously throughout a data run. Since most data acquisitions involved a short flight in the airport traffic pattern, a HUD display for takeoff was required. Figure 13 presents an example of the takeoff symbol set. The symbol set utilized the full set of basic flight symbology (described in section 3.1.3) superimposed on a subset of the ROTO on-ground symbology (described in section 3.1.2). The ROTO information box, football symbol, goal line, projected exit speed, and exit taxiway were suppressed but the trend vector and runway remaining signs were drawn. A symbolic runway and centerline were drawn on the HUD in a manner to be conformal with the actual runway. Following the T-NASA taxi format, pavement edges of the runway were identified with synthetic cones or “plops” and the centerline was depicted with rectangles. Both types of symbols were located fifty feet apart. Width information contained in the ROTO database was dynamically accessed when the ROTO module was started on the ground. The basic flight display elements were supplemented with the radar altitude in the lower right portion of the display whenever its value was less than 2500 feet.

#### ***3.1.5 Runway Exit Plan-View Display***

**Display Format:** The ROTO module could also produce a special HUD display depicting all of the defined ROTO exits for the current runway. That display was intended to provide the pilot with a graphical representation of the available choices and the selected exit taxiway. An example of the display is shown in Figure 14. It was invoked by placing the Preferred Exit knob on the PID momentarily in the RWY position during normal flight operations. For five seconds, the special HUD display superseded the normal in-air symbology. After the display was initiated, the mode of operation and the selected exit were determined from the current position of the Preferred Exit knob. While the display was active, each movement of the knob selected a new exit and reset an event timer that limited the duration of the special display to five seconds. The pilot could maintain the special display as long as desired and review any possible ROTO exit by moving the knob to a new exit selection before the five second period expired. If the knob was stationary for five seconds, the special display was terminated and a ROTO display corresponding to the current selection was drawn. If the knob was placed in the “RWY” position and not moved to an exit selection or the “AUTO” setting, the ROTO module defaulted to the

automatic exit selection mode and guidance to the system selected exit was displayed. The following display elements comprised the ROTO taxiway plan-view display:

**Runway and Exit Taxiway Edge Lines:** T-NASA taxi route structures were used to simultaneously draw the runway and up to nine ROTO exit taxiways on the HUD display. However, solid lines were used to mark the pavement edges of those elements instead of the T-NASA “plops.” The routes associated with the exits were indexed in the same order as the ROTO exits would be encountered, beginning at the runway threshold. To prevent overlaps, route segments associated with each exit route did not commence drawing until the previous exit taxiway route deviated from the runway centerline. Any runway length, beyond the last ROTO exit, was drawn using the runway only route. The eyepoint was fixed at the runway threshold but displaced upward and tilted downward to make a nearly plan view display of the available ROTO exits. This approach foreshortened the apparent length of the runway and permitted screen elements to be drawn larger than they could have been in a true plan view. A fixed width equal to the runway width was used for the runway and all taxiway routes.

**Taxiway Labels:** Each ROTO exit taxiway was labeled with its three letter alphanumeric designation. The ROTO data structured was accessed to determine the appropriate designation for each exit as it was drawn. The labels were positioned near the end of the ROTO portion of the exit taxiway route where the transition to T-NASA ground guidance occurred.

**Selected Exit Centerline:** A solid line was used to depict the centerline of the selected exit path. All other centerlines, including the runway centerline, were suppressed. This provided a visual depiction of the selected exit, which correlated with the information in the ROTO box.

**ROTO Box:** Information about the currently selected exit taxiway was displayed in an information box located on the upper right side of the HUD display. The box indicated the current ROTO control mode, selected exit, targeted exit speed, and the nominal braking distance. This information was read from the ROTO database and updated each time the exit selection was changed.

**Airport Information:** The name of the approach runway, the airport and a four letter airport designation were displayed in the upper left portion of the HUD display. This information was read from the ROTO database every time the ROTO system was restarted. A three letter airport alphanumeric designation, entered on the command line which started the IDS, was used to determine the current airport and the proper ROTO database.

**Runway Information:** The length, width, elevation and magnetic heading of the currently selected runway were displayed just below the airport information lines. The data were determined from the ROTO database using the ILS frequency as a search key.

### **3.1.6 Display Calibration**

The requirement that edge markers and center lines depicted on the ROTO HUD be conformal with the actual locations of these objects on the airport surface meant that the IDS software and the hardware device drivers for the HUD had to be calibrated as a system. Initial calibration was achieved using an alignment target mounted on a moveable stand that could be located at known distances from the pilot eye point of the B757. Field-of-view parameters in the IDS software and trim potentiometers in the HUD interface unit were adjusted. The alignment was refined by placing the aircraft at a carefully surveyed point on a surveyed taxiway at LaRC. Surveyed points close to the aircraft and more than one thousand feet from it were used in conjunction with the local horizon for the final alignment. This procedure produced a “flat Earth” alignment that was very satisfactory on the ground. It was found in flight operations that the HUD artificial horizon appeared too high at operating altitudes due to the Earth’s curvature.

A different calibration procedure was required for IDS operations in LaRC simulation facilities. No interface hardware was involved since the HUD symbology was merged electronically with the out-the-window scene by a video mixer. Software parameters such as the field of view were adjusted to achieve alignment of HUD symbols with features of the scene such as the runway location, and the horizon during flight and taxi maneuvers. Alignment was verified by checking the invariance of the compass direction toward fixed objects on the ground under changes in aircraft heading and attitude. Due to an electronic artifact of the video mixing process, an empirical offset in yaw was necessary. That negatively impacted the conformality of the HUD symbology, particularly during roll maneuvers, but caused little problem when the aircraft was in a “wings level” attitude or on the ground.

### **3.1.7 Databases**

Several different types of databases were developed for two airports: Atlanta and Langley Air Force Base. The databases unique to a specific airport were read into program memory during startup initialization enabling the IDS software to be generic for any airport. Pilot confidence in the HUD and/or HDD was critically dependent on the accuracy of these databases. The required accuracy was verified using DGPS data recorded during flight tests as the aircraft taxied on a known centerline and by the conformality of HUD (displayed symbols) with actual ground objects. The types of databases developed are described below:

**Visual database of airport surface:** The visual database was developed jointly by NASA and Lockheed Martin personnel using the MultiGen modeling tool. It included exact coordinates for all taxiway/runway centerlines, ramp and grass areas, buildings, and labels. Positions were accurate to plus or minus one foot based on surveyed coordinates provided by Jeppesen Sanderson, Inc. Although the database itself was highly accurate, in actual practice at Atlanta the centerlines of the database had to be aligned to match the DGPS flight data. The alignment was accomplished empirically by using recorded flight data to make small adjustments to the latitude and longitude of the database origin and determine the angle of deviation (theta angle) of the database from true north. Those values were modified in the airport initialization file with no changes required in software. They were read during startup initialization and used by the latitude/longitude conversion function (See 3.1.8). Before this alignment procedure was performed, the database centerlines were in error by as much as 30 feet on some taxiways and

runways. After alignment, the centerlines were within the range of DGPS accuracy, or approximately one meter on either side of the actual painted centerline.

**Ways database:** The ways database (taxi-ways, run-ways and ramps) provided all the data needed to analyze taxi route legs and determine intersecting ways, segment numbers for each leg, directions of travel, and hold short locations. A detailed breakdown of this database will be published in a separate NASA Contractor Report on the IDS taxi routing and monitoring system.

**Databases for ramps and runways:** Ramp and runway databases contained special data needed for ramps and runways which included names in various formats used for determining airport locations, directions of travel and interpretation of ATC controller messages.

**Hold bars database:** The hold bars database provided the names, descriptions, coordinates, and angles of all marked hold short locations used on the airport surface (see section 3.1.11.2). A detailed description of this database will be published in a separate NASA Contractor Report on the IDS taxi routing and monitoring system.

**AMASS hold bars database:** The AMASS hold bars database stored the sequence and coordinates for the AMASS hold bars associated with each runway and taxiway intersection on the airport (see section 3.1.1).

**Routes database:** The routes database contained the names and route numbering for all the pre-determined (canned) taxi routes (see section 3.1.11.1).

**Airport initialization database:** Data used in airport initialization such as magnetic heading, latitude and longitude of the airport map origin, angular deviation from true north and other information were contained in this database.

**ROTO database:** The ROTO database contained the specific information required for ROTO operations for each runway on the airport. This database listed each runway for the airport, and each exit taxiway for each runway. Tabulated data for each runway included the number of exits, the latitude, longitude, and altitude of the threshold, the ILS approach frequency, and the runway length, width and orientation. Tabulated data for each exit taxiway included distance from the threshold, width, angle with respect to the runway, nominal exit speed, maximum exit speed, and speed for signaling the code to transition to taxi mode. A route identifier was included, keyed to the Routes database. A detailed description of this database will be published in a separate NASA Contractor Report on the IDS ROTO module.

### ***3.1.8 Conversion of Latitude and Longitude Data***

Position data for ownship and traffic were given in degrees of latitude and longitude, which had to be converted into map Cartesian coordinates (x,y in feet) for use by the graphics routines. The software for the conversion function had been used successfully on a previous LVLASO flight test at Atlantic City in June 1995 [6], and was reused during the flight tests and demonstrations at Atlanta. Calculations for the conversion included the latitude and longitude of the map database origin and the angular deviation of the map database from true north, values that were stored in the airport initialization database. The values were modified based on empirical data during the database alignment process as described in section 3.1.7 above.

### **3.1.9 Pilot Input**

Pilot interactive input to control the displays was provided by the pilot input device (PID) mounted in the flight deck (see section 2.1), and software in the IDS communications and display programs that interfaced to the PID. PID inputs were transmitted as bit mapped data where each bit represented a state of one of the knobs, switches, or buttons of the PID. The data were transmitted continuously through the aircraft data communications system. PID interface software in the display program performed necessary tasks and changed the displays based on pilot inputs and corresponding changes in PID states.

The PID is shown in figure 2. The types of pilot inputs available on the PID and the results of those inputs for the HDD moving map and the HUD during ROTO operations are described as follows. There were no pilot inputs available for the HUD during taxi operations.

#### **Pilot inputs for the HDD moving map:**

(1) Change zoom level (map scale) using the FIELD OF VIEW knob. Six positions on the knob were APT for a north up plan view of the entire airport with an eyepoint of 12000 feet above the surface, and X1 through X5 for zoom levels of 10000 feet through 2000 feet respectively.

(2) Turn traffic on or off with the TRAFFIC switch.

(3) Turn traffic tags on or off with the LABELS switch.

(4) Acknowledge receipt of ATC messages using the CONFIRM button.

(5) View LED lights to determine when ATC messages were received and acknowledged.

The message manager software in the display program for the HDD initiated data through the PID interface software to control the LED lights when ATC messages were received and acknowledged (see section 3.1.10.1). The lights were turned on when messages were received and turned off after all messages were acknowledged by the CONFIRM button. The PID interface software sent the lights control data back to the PID through the IDS communications program.

(6) Scroll messages forward using the FWD button or backward with REV button.

#### **Pilot inputs for the ROTO HUD:**

(1) Select preferred ROTO exit. During ROTO operations the pilot used the PREFERRED EXIT knob to control the ROTO module and change the HUD display. Pilots could select a preferred exit by turning the knob to a numbered exit in the manual range or choose the "AUTO" position and let the software select an exit automatically.

(2) Check available exits. Placing the PREFERRED EXIT knob in the "RWY" position initiated a plan view on the HUD of all the available ROTO exits, which the pilot could reference during his/her exit selection decision.

### ***3.1.10 Controller-Pilot Message Communications***

The proper handling and management of non-verbal message communications between the pilot and ATC controller was a major objective of the LVLASO experiment. For the flight tests at Atlanta, Controller-Pilot Data Link Communications (CPDLC) were used. CPDLC protocols were defined for the enroute and terminal area by RTCA DO-219 MOPS for ATC Two-Way Data Link Communications. Selected messages from DO-219 and new messages for the ground controller were implemented in a Controller Interface (CI) program developed by St. Cloud State University, MN. The CI was used to transmit uplinked messages from a LVLASO test controller to the NASA research aircraft, and receive the downlinked messages initiated from the aircraft to the test controller.

The CI utilized a voice recognition system to convert ground controller instructions into digitized text messages for transmission to the aircraft. The LVLASO test controller operated the CI on a PC controller workstation. Atlanta ground control instructions for the aircraft were repeated into a voice recognition system that was trained for the voices of each test controller. The spoken messages were converted to CPDLC protocols and transmitted via the Mode S datalink. Uplinked messages were received on the aircraft and sent via the data communications system to the display program for both the HDD and HUD. Downlinked messages were initiated by the display program and sent via the IDS communications program as described in section 3.1.10.1 below.

The ATC message types defined for LVLASO are listed in Figure 15.

#### ***3.1.10.1 Message Manager***

The message manager was a set of software functions in the display program that managed or performed the tasks associated with uplinked and downlinked messages. When an uplinked message was received via the communications program in the real-time data block, the message was copied to a buffer and parsed by type to determine the correct format. Depending on the type, the message would be passed to another function for further parsing and type specific processing. Messages applicable to taxi routes were forwarded to the dynamic taxi routing function (see section 3.1.11.1). Messages applicable to hold bars were sent to the hold bars function (see section 3.1.11.2). Message types 1 through 153 were not applicable to any other function and were not passed.

When messages were passed to the taxi routing or hold bars functions, the message manager waited for a response from those functions to determine if the message was processed successfully. If an error condition was returned, the message manager sent an "UNABLE" downlinked message (see Sending Downlinked Messages below). The message to which the "UNABLE" applied was not displayed to the pilot. If a successful status was returned from the other functions, the message was displayed on the HDD (see section 3.1.1) by entering the message in two queues, the display queue and acknowledge queue. Exceptions were message types 223 and 224, which were neither displayed to the pilot nor acknowledged during the LVLASO experiment in compliance with ATC requirements at Atlanta. All other messages were entered into the display and acknowledge queues, including messages that were not passed to other functions.

**Message queues:** The message queue used for display was implemented as a string buffer array that contained an arbitrary maximum number of messages (six was used). The buffer was managed as a circular queue by using modular arithmetic on the buffer indexes (locations of messages in the buffer) to wrap messages in sequence from the last location back to the first. When the buffer was filled with the maximum number of messages, a new message was placed in the buffer overwriting the oldest message. The display queue kept track of the buffer locations of messages that were currently being displayed (up to three) and the locations of all messages in order from oldest to newest. The acknowledge queue used a similar approach and kept track of which messages had been acknowledged and which were still waiting to be acknowledged.

**Message scrolling:** When the pilot pressed one of the forward or backward scroll buttons on the PID (see section 3.1.9), the PID interface software called the scrolling function in the message manager, which set the queue indexes for the messages to be displayed in the scroll area of the HDD. Up to three messages could be displayed at any one time, and those could be any of the last six messages in the order they were received. However, if any messages were not acknowledged by the pilot, the scroll buttons were ignored. The scrolling information provided by the message manager, in conjunction with the queuing information described above, was made available to the drawing function to determine what and how messages should be displayed to the pilot.

**Message acknowledgment:** Two different procedures were followed at Atlanta for acknowledgment of messages by the pilot. During the first phase of testing, the pilot was notified of new messages when the LED light on the PID was turned on, and acknowledged receipt by pressing the CONFIRM button (see section 3.1.1). When that event occurred, the message manager initiated data to turn off the LED lights if there were no other messages to be acknowledged, and also sent a downlinked “ROGER” message to notify the LVLASO test controller that the message was received by the pilot. The procedure was difficult for pilots to comply with in a timely manner due to other tasks required during taxi operations. Therefore, during the second phase of testing, the procedure was changed such that all messages were automatically acknowledged by the software. Messages continued to be displayed in the same manner; however, the PID lights were no longer used and the pilot no longer pressed the CONFIRM button. The message manager automatically sent a “ROGER” for all messages immediately upon receipt, except when “UNABLE” messages were sent for bad error status returns.

**Sending downlinked messages:** A separate function in the message manager handled the task of sending downlinked messages when called by other functions. In addition to “ROGER” and “UNABLE” messages discussed above, this function handled downlinked messages for “TAXI DEVIATION” and “TAXI DEVIATION RESOLVED” which were initiated by the taxi monitor function (see section 3.1.11.3). The taxi monitor called the message manager to send a message when a taxi deviation occurred, or when the deviation was resolved by the aircraft getting back on the taxi route. All downlinked messages were handled by the message manager in the display program for the HDD, except messages for “TURNED OFF ON TAXIWAY”. Those messages were initiated and handled by the ROTO HUD software when the aircraft exited the runway.

The technical procedure for sending downlinked messages involved shared memory interprocess communications with the IDS communications program. The message number and downlink

message type were written into shared memory. The communications program placed the data on the SCRAMNet+ fiber optic ring for downlinking the message to the ground controller.

### ***3.1.11 Taxi Routing and Monitoring***

Taxi routing and monitoring were highly interdependent software components of the IDS display program that included: (1) methods of creating and displaying the taxi route clearance received from the ATC controller, (2) implementation of hold short instructions from the controller, and (3) monitoring the location and status of the aircraft during taxi operations. Creation of taxi routes included the use of pre-planned (or canned) routes and a method of dynamically computing the route based on the ATC instructions. Hold short instructions were implemented as hold bars drawn at designated locations on the taxi route. Taxi monitoring included the capability for the pilot and controller to be continuously aware of the aircraft's location and status of taxi operations relative to the cleared route. An overview of each of these areas is provided in the subsections below. More complete technical descriptions will be documented in a separate NASA Contractor Report on taxi routing and monitoring for the LVLASO Integrated Display System.

#### ***3.1.11.1 Dynamic Taxi Routing***

The single most important feature for taxi operations on the HDD and HUD was the taxi route. Taxi routes from the ATC controller were graphically represented as magenta paths on the HDD moving map (see figures 5 and 6), and as pathways with centerlines and width markers on the taxi HUD (see figure 7). For most airports, and especially large airports like Atlanta, the number of possible paths is extremely large. The use of canned routes, although adequate for simulations, was inadequate for actual flight operations. To solve this problem, a routing algorithm was developed to compute the taxi path dynamically when the ATC taxi instruction (message) was received. This algorithm was able to create and draw any valid taxi route path in the airport database, and obviated the need for a large number of canned routes. The dynamic router first searched a database of pre-developed canned routes using a name created from the contents of the ATC taxi route message. If a canned route by that name was found, it was used. If not found, the dynamic routing algorithm was used to compute the route. During flight tests at Atlanta where ATC taxi route instructions were transmitted through the LVLASO test controller, canned routes were not used and dynamic routing was essential.

Canned routes were retained in the IDS primarily for use in flight simulations and flight tests where no controller messages were available. The implementation of canned routes was enhanced by the development of a naming system that converted a controller taxi instruction into a name for lookup in a canned routes database. The canned routes database also included a number that could be used to enter routes through the control program (see section 3.3). Canned routes were used extensively for local flight testing at Langley AFB in June and July 1997.



### ***3.1.11.2 ATC Hold Short Instructions (Hold Bars)***

Hold bars were drawn on the HDD moving map of the airport to graphically depict “hold short” instructions from ATC ground control. They were drawn as a red rectangle with a yellow border at the map location corresponding to the actual painted hold line on the taxiway surface. In cases when there was no painted hold line on the taxiway, the hold bar was drawn across the taxi route prior to the intersection stated in the controller’s hold instruction.

Hold bars were an important part of the taxi clearance and, therefore, were closely related to the taxi route. A taxi route had to be issued prior to any hold instruction in order to determine the location of the hold bar. Often, the hold instruction was issued at the same time as the taxi route, or was issued in conjunction with the taxi route as an end point for a progressive route segment where additional taxi instructions were required to complete the next portion of the route.

Two types of hold bars were defined in the IDS, departure runway hold bars and enroute taxi hold bars. Departure runway hold bars were implied if any taxi route was issued to a runway for takeoff. They were automatically drawn at the intersection of the taxiway and departure runway, even if the controller did not give a specific hold short instruction. The departure runway hold bar could only be removed when the controller issued either a “TAXI INTO POSITION AND HOLD” instruction, or a “CLEARED FOR TAKEOFF” instruction. Enroute taxi hold bars were issued for hold short positions on the taxi path prior to the route destination (runway or ramp). In accordance with ATC procedures, only one enroute hold instruction (and hold bar) could be issued at any one time. This hold must be cleared before the next enroute hold instruction is given. The enroute hold bar was removed when the controller issued a “CONTINUE TAXI” or “CROSS RUNWAY” instruction.

### ***3.1.11.3 Taxi Monitor***

The taxi monitor software performed a number of essential functions for the IDS. It’s primary function was monitoring the status and location of the aircraft (ownership) relative to the taxi route (i.e., current leg, taxiway segment name and numbers, current branch, direction of travel, on route/off route, airborne, on ramp, on runway, on grass). This information was needed by other functions of the IDS such as the dynamic router, message manager, hold bars handler, and ROTO to determine the current aircraft status. For example, the dynamic router required the current aircraft movement, taxiway segment and leg number to determine data for the beginning of the taxi route. The ROTO HUD module terminated the ROTO display when information from the taxi monitor indicated the aircraft had exited the runway. Hold bar placement was determined based partly on the current aircraft position from the taxi monitor.

A significant task for the taxi monitor was to trigger the transition from taxi display on the HUD to ROTO display. The transition was triggered when any of several conditions were detected; when the aircraft was on the last leg of the taxi route and also on a branch to the departure runway; when the aircraft entered the departure runway after a taxi deviation, or the aircraft became airborne.

Determining when taxi route deviations occurred was another important function. During the flight tests at Atlanta, the LVLASO test controller was notified immediately via a downlinked “TAXI DEVIATION” message when the aircraft taxied off the cleared taxi route. The taxi

monitor initiated this message through the message manager when a defined deviation was detected such as making a wrong turn onto a runway or taxiway or not making a turn when required. In some situations, it was possible for the aircraft to be off the route and not be considered a deviation. For example, the aircraft could depart from the end of a route to taxi into the ramp area. Also, the aircraft might appear to be off the route when required to taxi far off the centerline of a taxiway during wide turns, or to make room for other traffic. Deviation messages were not sent in those cases. If a true deviation did occur and a message was sent, the taxi monitor continued to monitor the aircraft to determine if the aircraft returned to the taxi route, and if so, where. If determined that the aircraft was back on the taxi route, the taxi monitor sent a downlinked "TAXI DEVIATION RESOLVED" message to the controller.

The taxi monitor was also responsible for removing the taxi path off the HDD and HUD when the aircraft reached the end of the taxi route at the final destination.

## **3.2 Real-Time Data Communications Module**

Requirements for the data communications software included the capability to receive real-time data that arrived at various rates, record that data, and playback the data. A derived requirement that proved to be essential aboard the B-757 research aircraft was to have a means of monitoring the real-time data. Other derived requirements were to have a means of interacting with the displays during playbacks, to manage uplinked data according to datalink integrity, and to be able to pause and then continue playing back the recorded data for demonstration purposes. Further details about those programs will be published in a separate NASA Contractor Report on the IDS data communications system.

### ***3.2.1 Datablock Design***

A suitable datablock had to be designed in order to provide, in a timely manner, the data required to drive the T-NASA taxi HUD display, the taxi map display, and the HUD guidance for all phases of flight. A search through the Boeing 757 Fault Isolation Manual was made to identify possible sources of required ownship parameters. ARINC-429 data bus charts provided information such as the signal, format, units, binary range and label of data available from a given aircraft system, for example, the Inertial Reference System, the Flight Management Computer, and other systems. If certain parameters were unavailable, an alternate strategy for attaining the necessary data was devised. Since those parameters also had to be available in the simulator, considerable effort was made in coordinating the data between the simulation and flight environments. In many cases, a necessary parameter was available on the aircraft, but had to be computed from other available data in simulation.

Three shared memory segments were created by the communications software regardless of the actual communications interface. Those areas were the aircraft ownship data, the uplinked data, and control data for interprocess communications. That type of arrangement had worked well for the Atlantic City (ACY) flight test and was suitable for use by both the Indigo 2 and the PI. The control shared memory segment, designed to support the flight tests and demonstrations at Langley Air Force Base and Atlanta, also included an area for downlinks and recording of specific ROTO parameters. The Appendix contains the SCRAMNet+ shared memory definition

used for intersystem communications on the B-757 research aircraft. The communications software utilized a subset of that data to populate the three shared memory areas used by the display programs on the two SGIs.

### ***3.2.2 Device Drivers and Communications Programs***

The source of real-time data in simulation was the CAMAC serial highway. A DR11W device driver for IRIX 5.3 was adapted for the Personal Iris, and an application program to receive the real-time data was written. The communications software was migrated to the Onyx with some minor modifications for that architecture. A DR11W applications program written in support of the ACY flight demonstration in 1995 served as a baseline for the new simulation software supporting the 1997 Atlanta flight tests and demonstration. Likewise, a recording and playback scheme having its roots in a previous capability developed in simulation in preparation for the ACY flight tests and demonstration served as the baseline for a simulation playback capability. All recording and playback were performed at 32 Hz, the real-time datablock arrival rate.

Initially, the communications mechanism onboard the B-757 research aircraft was unknown. Preparations had been made for doing either serial interfaces or an ARINC-429 interface on the PI with a network connection to the Indigo2 in order to pass the real-time data. Serial communications software was written and had been tested in the TSRV simulator. A recording and playback strategy where data arrived at different rates on each interface was devised. The availability of a SCRAMNet+ communications ring and an I/O concentrator whose sole function was to read the ARINC-429 data busses and send/receive datalinked items such as message acknowledgments, traffic and runway status information (AMASS hold bar states), ground controller messages, and DGPS information necessitated a total rethinking of the communications strategy. It was clear, however, that the SCRAMNet+ network communications offered the most efficient and timely delivery of real-time data to both platforms. In addition, the data would be readily available to the DAS computer for recording.

SCRAMNet+ cards were jumpered correctly and installed on the Indigo2 which had an EISA backplane and the PI which had a VMEbus slot available. Device drivers were modified and installed so that the hardware would be recognized by the operating system on each host. Communications software including interrupt handlers were tailored specifically for each host, installed, and tested. Some commonality between the two communications programs existed. Implementations on the two computers were similar, but not identical due to differing details of hardware.

The communications software was designed to read and record from and write to SCRAMNet+ shared memory areas. Subsets of those SCRAMNet+ shared memory areas were contained in the shared memory areas used by the display software and the control software on each host. The SCRAMNet+ memory contained additional information such as interrupt words and datalink status. If a Mode S controller uplink or a VHF traffic and AMASS uplink was found to be corrupt, that data were never passed on to the display software. The status of the DGPS receiver was passed on to the displays that could transition to other states when the ownship position was less accurate. Interaction with the PID by the display software was ultimately provided by two SCRAMNet+ memory words.

Ownship data arrived at a 25 Hz data rate while traffic and AMASS information from the VHF datalink was available only once per second. DGPS datalinked items attained via the Mode S receiver also arrived at a 1 Hz data rate. Mode S controller message uplinks and downlinks occurred as messages arrived. A parallel process for recording data was devised to emulate the DR11W process performing the same function in the VMS simulator. The process was spawned only as required for recording for efficient use of the CPU. This program recorded, at the appropriate data rate, SCRAMNet+ memory instead of shared memory as performed by the DR11W recording process. The purpose for this decision was to have complete traceability as to when datalinked items arrived and when a number of external events occurred. UTC time was included in the ownship datablock. It proved useful for tracking events in time, and for some time dependent functions in the display programs.

### ***3.2.3 Playback Programs***

A playback routine was written to play back recorded real-time information at the aircraft data rate of 25 Hz. A requirement for interactive demonstration of the capabilities of the LVLASO-IDS system led to the development of a special version of the code. It was necessary that the test subject be able to use the PID for selecting what was displayed on the HDD and the HUD during a playback to experience how a pilot would interact with the displays. To accomplish this, a scheme was devised whereby the playback routine would load the shared memory areas with actual recorded data but would override the PID states and replace them with the selections made on the actual PID. This program was a complicated procedure since it had to interact with the SCRAMNet+ memory at the same time it was playing back the data. An added pause capability proved to be invaluable when training pilots on the use of the PID and the HDD.

### ***3.2.4 View Programs***

The view program monitored the real-time data on the aircraft. That proved to be useful for detecting and diagnosing datalink problems and for monitoring the content of the uplinked information and other data. Ownship parameters as well as datalink status, GPS information, traffic data and AMASS hold bar status from the VHF datalink, CPDLC ground controller message information from the Mode S datalink, Mode S downlinked message information, and ROTO parameters for recording could be selected for viewing. The view program attached itself directly to the SCRAMNet+ shared memory. A separate view program, which read from the shared memory areas, monitored recorded data from a playback file. That was useful during post-analysis of a run in determining precisely when certain events occurred.

## **3.3 Control Program**

The control program was developed to provide a capability for the IDS operator/developer to interact with the displays and communications software to perform a number of input tasks, and to provide a capability for testing and simulation. It was a separate program that ran concurrently with the other processes using the same shared memory. The following functions were provided: start/stop data recording; select ROTO algorithms; enable/disable transitions to ROTO; simulate the pilot input device to control displays; enter taxi routes and hold bars by number or name; and simulate ATC messages for testing.

## 4.0 CODE VALIDATION

The T-NASA and ROTO functions were developed in parallel as separate codes. A merged code was created and tested only after each portion had been well defined. That greatly simplified the procedures required to make software changes.

System development and testing occurred concurrently, since specifications changed frequently. Most of the HUD testing took place in the Virtual Motion Simulator (VMS) facility. A virtual HUD display was used, as explained above. The ROTO code was tested by performing repeated simulated landings. In-air symbology was compared with the readings of the cockpit instruments. On ground, the symbology required during deceleration was tested for acceptability of the look and feel, and was also tested quantitatively. The smoothness of transition between modes was also tested. Simulated takeoffs were performed in order to test the transition back to flight. The functionality of the PID rotary knob for exit selection was tested.

The HDD unit and the IDS software supporting it were operationally tested in the EASILY laboratory and on the B-757. Synthetically generated messages, traffic data recorded at the Atlanta airport, playbacks of recordings made in the VMS simulator, and actual data link communications with the aircraft were all used during this process. Conformality of the taxi displays on the HDD and the HUD with the actual conditions surrounding the B-757 aircraft was validated by navigating taxiways at Langley Field.

Software was written to enable the recording of the inputs that drove the simulation. That capability greatly increased the value of the simulation sessions by making it possible to continue testing the code off-line. It was especially valuable for investigating a combination of circumstances in which the code failed. Another testing tool, which was used extensively, was to add print commands to the code to write selected variables to a file for detailed analysis.

After testing in the VMS had crystallized the code into a nearly final form, further testing was done in the Experimental Aircraft Systems Integration Laboratory (EASILY). The functionality of the HDD moving map was tested; choice of zoom level, perspective view, proper identification of taxiways, and message handling capabilities were thoroughly exercised. The code was interfaced with a replica of the actual computer system installed on the aircraft. In this stage, the data blocks were put in their final form, and issues of sign convention, resolution and range of parameters were resolved.

Due to a tight schedule in which experimental configuration activities competed with preparing the aircraft's basic systems for flight, flight testing of the experimental systems was very limited. That testing was essential for validating some inputs that were not available in the VMS or achievable in EASILY. Aircraft testing included taxiing around Langley field, and multiple short local flights that included touch-and-go landings.

## 5.0 LESSONS LEARNED

This section presents some observations and recommendations, related to the overall development of the IDS software. No attempt is made to analyze the results of the Atlanta flight tests or present a comprehensive list of all the “lessons learned”, which molded the IDS into a flight worthy system.

An important element of the success of this project was the fact that the investigators were generally familiar with aircraft dynamics and aircraft operations. Collectively, there was prior experience in aircraft control systems, aircraft computer systems, aerodynamics, and airport traffic control. In addition, some of the investigators were themselves licensed pilots. Many valuable suggestions were received from the LaRC test pilots and airline pilot evaluators who took a keen interest in the project. Since the ultimate goal was industry acceptance of the system, it was most useful to infuse the project with an understanding of the industry’s wants and needs at every stage of development.

Testing was all-important. Every feature of the IDS was pre-tested in LaRC simulation facilities, in the LaRC EASILY laboratory, and through the use of recorded data. Even minor anomalies were pursued to a satisfactory resolution. It turned out that actual time for flight testing was very limited, so it was very fortunate that the codes were reliable before being ported to the plane. Very few shortcomings were discovered during the flight testing phase. The most serious shortcomings discovered were related to the design of the airport database and mismatches between the Boeing 737 aircraft dynamics available in the simulation facilities and the Boeing 757 research aircraft. Since future versions of the LVLASO experimental system will be increasingly intertwined with the aircraft flight management system and aircraft dynamics, the fidelity of simulation facilities will be more critical to mission success. However, it is unrealistic to expect general purpose simulation facilities to reconfigure for every possible combination of airframe, sensors, instruments and data structures found on variants of the same basic aircraft. Many hours were spent on board the B757 verifying data structures, sign conventions, and data formats that were different than expected. Therefore, it was vitally important to flight test the IDS as much as possible. In a highly integrated system, every change must be made cautiously, bearing in mind that other parts of the system might be affected. For example, it proved particularly difficult to track down and change all the parts that were affected by a simple change in the size of the data block that was sent to shared memory. It was important to anticipate situations that could lead to underflows, overflows, or invalid arguments. Data recorded in LaRC simulation facilities and during short local flights were particularly useful for locating and eliminating those kinds of problems.

The Atlanta airport database was designed to support studies of taxi operations along a single predetermined path on the surface. The design philosophy involved a two-dimensional decomposition of the runway and taxiways into fixed width route segments, each of which contained information about the segment centerline. The taxi route was constructed as a linked list of those route segments. The design was flexible and it resulted in a compact database. However, there were shortcomings of that approach. They are addressed below:

**Pavement fillets:** The fillets used between straight segments of the route cannot be addressed with the current database design, since only centerline information is encoded.

This meant that route edge markers or “plops” were not placed at the actual edges of the pavement. This feature was intentional for T-NASA taxi operations. It was undesirable for ROTO operations since it produced an inaccurate perspective of the exit taxiway. The ROTO software module sought to produce a display conformal with the outside world by dynamically adjusting the runway and taxiway width parameters from information in the ROTO database.

**Three-dimensional features:** All airport surfaces are three-dimensional. The Atlanta airport database was a two-dimensional idealization of the airport surface and did not contain elevation data. During taxi operations, graphics drawn on the HUD assumed a pilot eyepoint at a fixed constant height above the local surface. That produced a locally conformal display on the HUD. The apparent location of distant objects depicted on the HUD was in error by an amount proportional to the elevation difference between the current location and the true elevation of the surface at the distant location. That problem was mitigated by not depicting objects more than one thousand feet from the test aircraft. During ROTO and takeoff operations, however, it was necessary to depict the entire runway on the HUD at all times. Since elevations were not encoded in the database, positional data could not be used to determine the pilot eyepoint elevation above the airport surface during the landing approach and the takeoff climb out. Radar altitude was used instead.

**Exit location:** The current airport database was not designed to encode the proper place to begin a turn from the runway onto an exit taxiway or contain a mechanism to recognize the moment when the test aircraft first “cleared” the runway. Route segments were categorized as either “branches” or “ways.” Branches connected ways associated with either the runway or a taxiway. When located on a branch, it was not possible to tell whether the pavement under the aircraft was part of the runway or part of a taxiway. This made seamless transitions of the HUD symbology between ROTO and taxi operations more difficult to initiate.

The IDS software for the Atlanta flight tests was designed to operate in a test environment where features of the code and the displayed symbology could be refined. For daily flight operations, the flight crew would exercise more control over the IDS configuration and its functions. Some suggested control modifications are discussed below:

**Approach type:** The Atlanta IDS software used the ILS frequency as a search key to determine the intended runway. This system worked well since all flight tests involved ILS approaches and all runways had unique ILS front courses. However, to support a variety of approach types and avoid errors associated with inadvertently tuned receivers on non-ILS approaches, the IDS should receive the runway information directly from the flight management system.

**Mode switch:** The Atlanta IDS software was designed to automatically transition HUD symbology between flight phases based on air-ground squat information, the ILS frequency, the current aircraft location, and the taxi destination. This mechanism is limited to ILS approaches and uses a non-intuitive mechanism (a frequency change) to switch from the takeoff to the flight display. A pilot controlled mode switch to select the desired

display would have provided a more positive control over the symbology and simplified the IDS code.

**Symbology options:** Pilots indicated that options to control the drawing of selected objects or limit their size and/or intensity were desirable. Prevailing weather conditions could render some full size and/or full intensity ground symbology unnecessary. For example, edge markers were not needed for daylight operations in clear weather. A pilot controlled knob or other mechanism to selectively configure the IDS HUD display for IFR or VFR conditions would have partially addressed those desires. Since commercial variants of the LVLASO system will operate continuously, regardless of the weather conditions, future versions of the IDS may need mechanisms to tailor the displays for individual pilot preferences.

**HUD intensity:** The pilot seated behind the HUD must control the display intensity. Circumstances of the LVLASO HUD installation aboard the B757 precluded a cockpit mounted intensity control. Workable solutions to that problem were improvised during the Atlanta flight tests, but they were never fully satisfactory.

Areas for further development of the LVLASO guidance system and the IDS software are addressed below:

**Taxi route uplink:** A mechanism to uplink the assigned taxi route to the aircraft at the earliest possible time is required in order to fully utilize the ROTO capabilities of the LVLASO system. The ROTO module already contains mechanisms that could dynamically tailor the desired exit velocity and the taxi route to follow after turn off. Those features can be exploited to minimize the runway occupancy time or maximize passenger comfort and provide even more seamless transitions between deceleration and taxi guidance displays. The use of pre-defined taxi routes associated with each possible runway exit is less than satisfactory since routing exceptions will always exist due to the dynamic nature of the ground traffic. It is not practical or safe for the IDS system to dynamically determine the taxi route without authorization from ground controllers to begin and end the taxi route at assigned locations.

**HUD hold bars:** Pilots indicated that depicting the locations of ATC and AMASS hold bars on the HUD taxi display would be desirable.

**Message indicator on the HUD:** Pilots indicated that a symbol on the HUD to indicate an ATC message had been received would be desirable. Displaying entire messages on an unused portion of the combiner glass was also deemed potentially desirable.

#### **ATC Message Instructions for Taxi Routes:**

Two types of messages that contained taxi route instructions from the ATC controller were defined for the LVLASO experiment and flight tests at Atlanta (see section 3.1.10). Type 212, "TAXI RUNWAY [runway] VIA [taxi route]", was used for outbound taxi to the runway, and type 219, "TAXI RAMP [ramp] VIA [taxi route]", was used for inbound taxi. These messages were not sufficient to cover all the procedures used at Atlanta to issue taxi instructions.



Usually, controllers issued a taxi route only to the first hold short position and did not give the entire route to destination. The next portion of the route was not issued until the hold short was cleared. A typical controller instruction was: "Taxi via Alpha; hold short of 26 right at Dixie". Since a message format was not available to handle this instruction, the LVLASO test controller had to interpret the instruction in two separate messages for a taxi route and a hold short. For the taxi route portion, the only message that could be used was type 212 for a complete taxi route to a runway; however, this was incorrect since a complete route had not been given by the controller. The test controller had to guess what the route would be after the hold short position. To continue this scenario, a typical instruction for the next portion of the route was: "Cross 26 right; hold short of 26 left". Again, this was interpreted as two messages to clear the current hold short and hold short of the next runway on the route. The route was not explicitly stated but was assumed to be a continuation of the same taxiway the aircraft was currently on. That "implied" taxi route was not recognized by the routing software since the messages only applied to hold short positions. If the type 212 message previously sent did not include the implied route correctly, a new type 212 taxi route message was required.

Clearly, additional taxi route message formats and software modifications were needed. In particular, a format was needed for a compound message to include both taxi routing and hold short instructions in the same message. The format would have allowed progressive taxi routes to be given to a hold short position rather than for a complete route to destination. It is recommended that additional procedures and message formats for taxi routing be determined, and necessary modifications made to the taxi routing software.

#### **Database Development:**

The specifications and process for developing the airport visual database were defined in the T-NASA baseline software and were implemented in the LVLASO IDS. The database developed for the Atlanta airport was very effective for most operations during flight testing in August 1997. However, there were a number of issues and problems concerning the database that need to be addressed for future LVLASO flight tests. Those issues are discussed below.

The specifications for developing the database using the MultiGen modeling tool were difficult to apply and were prone to human errors during development. An extensive amount of time was required for testing and debugging, and many changes had to be made before the database was considered adequate enough for flight and simulation testing. In particular, the process of creating accurate centerlines on taxiways and runways using the surveyed data from Jeppesen (see section 3.1.7) was a major effort to complete. Although the Jeppesen data was highly accurate, the manual process of incorporating the data into the database using the T-NASA specifications resulted in some loss of accuracy. A system was needed to simplify or semi-automate the process of converting the Jeppesen data into the airport database. This system could not be fully automated and some manual entry would still be required. However, the manual workload could be greatly reduced, thereby reducing errors and improving accuracy. Significant changes to the T-NASA database specifications and software would be required to implement this system.

The T-NASA database did not include all data required for taxi routing, hold bars and ROTO functions in the IDS. The additional data had to be generated manually in separate text

formatted data files; and this process was time consuming and prone to human errors. If a new database system referred to in the above paragraph is implemented, it should include the additional data for those functions in order to eliminate the text data files.

During the flight tests at Atlanta, it was evident that 3-D data for elevation of taxiways and runways was needed for the HUD for both taxi and ROTO operations. Since the data was neither available in the database nor implemented in software, the HUD symbologies for centerlines and edges were not aligned vertically when there was a significant change in field elevation. The effect was to make the symbologies appear to be above ground level when the aircraft was on a down slope, and below ground when the aircraft was on an up slope. Since the Jeppesen data includes airport field elevations, the data could be incorporated into the database with appropriate modifications made to the HUD software. Any changes should be part of the new database system referred to above.

There was a limitation in the database for drawing taxi route paths. Some paths could not be drawn because the branches at intersections on the cleared taxi route were either not available in the database or were not connected to form a smooth path between two consecutive close turns. The problem was the result of using the Jeppesen data, which provided coordinates only for the centerlines exactly as they were painted on the airport surface. In some cases at the Atlanta airport, the controller issued a taxi route that did not have a designated painted centerline at an intersection, or the centerlines were available but not connected on close consecutive turns. When those situations occurred, the taxi routing algorithm could not find a complete path in the database and thus sent an "UNABLE" downlinked message to the controller. Those problem areas in the database were known and alternative routes were devised on a case by case basis, but this procedure would not be acceptable in a fully operational system. An overall solution was not readily apparent. A partial solution used for Atlanta was to add virtual branches into the database to make connections where the painted lines did not exist or were inadequate. That technique helped but did not include all the possible problem areas.

More work needs to be done both in developing database specifications and in developing software to solve the limitations and problems discussed above for future LVLASO experiments.

## **ACKNOWLEDGMENTS**

This work was performed under task order number ARTS-0-LDI04, for the Langley Research Center.

## **REFERENCES**

1. Young, Steven D. and Jones, Denise R.: Flight Demonstration of Integrated Airport Surface Movement Technologies. NASA/TM-1998-206283, January 1998.
2. Young, Steven D. and Jones, Denise R.: Flight Testing of an Airport Surface Guidance, Navigation, and Control System. Presented at the Institute of Navigation (ION) National Technical Meeting, January 21-23, 1998, Long Beach, CA.
3. Foyle, David C.; Andre, Anthony D; McCann, Robert S.; Wenzel, Elizabeth M.; Begault, Durand R.; Battiste, Vernol: Taxiway Navigation and Situation Awareness (TNASA) System: Problem, Design Philosophy, and Description of an Integrated Display Suite for Low-Visibility Airport Surface Operations. 1996 World Aviation Congress. October 21-24, 1996, Los Angeles, CA. Paper No. 965551.
4. McCann, Robert S.; Foyle, David C.; Andre, Anthony D.; Battiste, Vernol: Advanced Navigation Aids in the Flight Deck: Effects on Ground Taxi Performance Under Low Visibility Conditions. 1996 World Aviation Congress. October 21-24, 1996, Los Angeles, CA. Paper No. 965552.
5. Harris, Randall L., Sr.; and Hewes, Donald E.: An Exploratory Simulation Study of a Head-up Display for General Aviation Lightplanes. NASA TN D-7456, 1973.
6. Jones, Denise R. and Young, Steven D.: Flight Demonstration of Integrated Airport Surface Automation Concepts. AIAA/IEEE 14th Digital Avionics Systems Conference. NN. S-9, 1995. Cambridge, MA.

## APPENDIX

### SCRAMNET MEMORY LAYOUT

This Appendix contains the SCRAMNet+ shared memory definition used in the LVLASO Integrated Display System for intersystem communications on the B-757 research aircraft. The communications software utilized a subset of this data to populate the three shared memory areas used by the display programs on the two SGI computers. See section 3.2.

```
#define MAX_TARGETS 55
```

```
/*Shared Memory Area 1 - 172 bytes + 4 bytes for valids*/
```

```
struct FMS_Dat {
```

```
    double    y_lat;           /*latitude; IRS or blended IRS/DGPS*/
    double    x_long;         /*longitude; IRS or blended IRS/DGPS */
    float     UTC_time;       /*time in millisecs past midnight GMT*/
    short     radarAlt;       /*radar altitude in feet AGL*/
    short     DGPSalt;        /*DGPS altitude in feet MSL*/
    short     gSpeed;         /*ground speed in knots*/
    short     vertSpeed;      /*vertical speed in knots*/
    float     heading;        /*deg true north; CW from north is pos*/
    float     yawRate;        /*deg/sec; nose right is pos*/
    float     pitch;         /*deg; up is pos*/
    float     pitchRate;     /*deg/sec; up is pos*/
    float     roll;          /*deg; right wing down is pos*/
    float     rollRate;      /*deg/sec; right wing down is pos*/
    float     trackAng;      /*track angle; deg true; CW from
                             north=pos*/
    float     lonAccel;       /*in G's (32.2ft/sec**2); forward is pos*/
    float     latAccel;       /*lateral acceleration in G's; right is pos*/
    float     vertAccel;      /*in G's; up is pos*/
    float     trackAccel;     /*in G's; forward is positive*/
    float     crosstTrkAccel; /*in G's; right is pos*/
    float     rightEngEPR;    /*right engine EPR; ratio always positive*/
    float     leftEngEPR;     /*left engine EPR; ratio always positive*/
    short     trueAirSpd;     /*true air speed in knots*/
    short     calAirSpd;      /*calibrated air speed in knots*/
    short     windSpd;        /*wind speed in knots*/
    short     windDir;        /*wind dir in deg true; CW from
                             north=pos*/
    short     totalAirTemp;   /*degrees C*/
    short     throttlePos;    /*thrust degrees; forward thrust is pos*/
    short     rudderPos;     /*degrees; right rudder is pos*/
    short     elevatorPos;   /*degrees; surface up is positive*/
    short     air_ground;     /*air ground system; discrete 0 or 1; main
                             gear on ground = 1*/
```

```

short      noseWheelSquat;    /*nose wheel squat system; discrete 0 or
                               1; nose wheel on ground = 1*/
float      acWeight;          /*weight of aircraft in lbs.*/
float      pitchFltCmd;       /*pitch flight director command*/
float      rollFltCmd;        /*roll axis (lateral) flight director cmd*/
float      fltPathAng;        /*flight path angle in degrees (+180 to-
                               180)*/
float      fltPathAccel;      /*flight path acceleration in Gs (-4 to +4)*/
float      ILSfrequency;     /*ILS frequency*/
float      ILSlocDev;         /*ILS localizer deviation (+0.4 to -0.4 );
                               DDM*/
float      glideSlopeDev;     /*glide slope deviation (+0.8 to -0.8);
                               DDM*/
short      ILScaptureFlag;    /*ILS frequency captured; discrete*/
short      baroAlt;           /*corrected baro altitude in feet MSL*/
short      GoAround;         /*0 or 1; 1 = Go Around operative*/
short      FMSreserve1;
int        FMSreserve2;
unsigned int FMSvalids;      /* bit map of FMS valids*/
};

struct GPS_Dat {
float      lat;               /*GPS or DGPS latitude*/
float      lon;               /*GPS or DGPS longitude*/
short      GPSaltitude;       /*GPS altitude in feet*/
short      GPSreserve1;       /*can also get trk hdg,eastvel,northvel*/
unsigned int GPSstatus;       /*GPS status word; 0 = good, 1 = no
                               differential corrections received but the
                               smoothed solution is still good, 2 =
                               smoothed solution bad but corrections
                               now arriving, 3 = no corrections and
                               the smoothed solution is bad*/
};

struct rtdata_aircraft {
unsigned int inputFlags;      /*pilot input control switches*/
short      opMode;           /*used in simulation*/
short      runNum;           /*used in simulation*/

struct FMS_Dat  ownship;      /*Flight Management System (FMS) data*/
struct GPS_Dat  dgps;         /*Global Positioning System (GPS) data*/
};

```

```
/*Shared Memory Area 2 - 1892 bytes*/
```

```
struct traffic_Dat {  
  
    int         id;           /*mode-S id or unique # for each target*/  
    char        fltNum[12];  /*flight number*/  
    float       y_lat;       /*y-coord or latitude*/  
    float       x_long;      /*x-coord or longitude*/  
    short       altAGL;      /*altitude AGL*/  
    short       heading;     /*course heading or track*/  
    short       gSpeed;      /*ground speed*/  
    short       reserve;     /*reserved*/  
};  
  
struct rtdata_ground {  
  
    /*Traffic datalink - VHF */  
    /*Traffic-target data*/  
    struct traffic_Dat target[MAX_TARGETS]; /*data for each target*/  
    int         numTargets;           /*# of targets each transmission*/  
    int         targetScanCnt;       /*a scan of traffic data is received*/  
  
    /*AMASS data*/  
    unsigned int  AMASS_Hbars[4];    /*AMASS holdbars bit map*/  
    short         rwyWindSpd;        /*runway wind speed in knots*/  
    short         rwyWindDir;        /*runway wind direction in degrees*/  
    short         rwyFrictCoef;      /*not used - rwy friction coefficient*/  
    short         rwyRVR;            /*runway visual range in feet*/  
    unsigned int  VHF1status;        /*VHF datalink 1 health;  
                                     1=corrupted,0=good*/  
    unsigned int  SCRAM1intr;        /*SCRAMNET interrupt word - VHF  
                                     memory area */  
  
    /*Controller data link - Mode S */  
    short         msgNum;             /*msg number from controller message*/  
    short         msgType;            /*msg type from controller message*/  
    char         message[80];        /*controller uplinked message*/  
    unsigned int  modeSstat;         /*Mode S health; 1=corrupted,0=good*/  
    unsigned int  SCRAM2intr;        /*SCRAMNET interrupt word - Mode S  
                                     memory area */  
};
```

*/\*Shared Memory Area 3 - 56 bytes including system\_sync which formerly was the next 4 bytes beyond the LVLASO specific SCRAMNET definition \*/*

```
struct lvlaso_outputs {  
    short      ATC_msgNum;      /*ATC msg being ACKed (0 if element id =  
                                202 or 203*/  
    short      ATC_elementID;  /*CPDLC downlink msg; 1=UNABLE,  
                                3=ROGER, 202=TAXI DEVIATION,  
                                203=TURNED OFF ON TAXIWAY,  
                                204=TAXI DEVIATION RESOLVED*/  
    char       taxiway[4];     /*Taxiway turned onto during ROTO; left-  
                                justified, blank filled (element id=203)*/  
    unsigned int SCRAM3intr;   /*SCRAMNET interrupt word - Mode S  
                                downlink memory area */  
    int        PIDstates;      /*Pilot input device MSG and CLR states*/  
    float      velCmd;         /*ROTO velocity command*/  
    float      velDev;         /*ROTO velocity deviation*/  
    float      latPathDev;     /*ROTO lateral path deviation*/  
    short      rotoKnob;       /*ROTO knob selection*/  
    short      rotoAlgRWY;     /*ROTO rwy selection */  
    float      rotoRWY_x;      /*ROTO rwy x-coordinate*/  
    float      rotoRWY_y;      /*ROTO rwy y-coordinate*/  
    int        rotoAlgNum;     /*ROTO algorithm used (0 - 4)*/  
    int        rotoSpare1;  
    unsigned int SCRAM4intr;   /*SCRAMNET interrupt word - ROTO  
                                datablock memory area */  
    unsigned int system_sync;  /*SCRAMNET synchronization interrupt  
                                word - lat/longs intact */  
  
};
```

**Figure 1 – LVLASO IDS AIRCRAFT HARDWARE ARCHITECTURE**

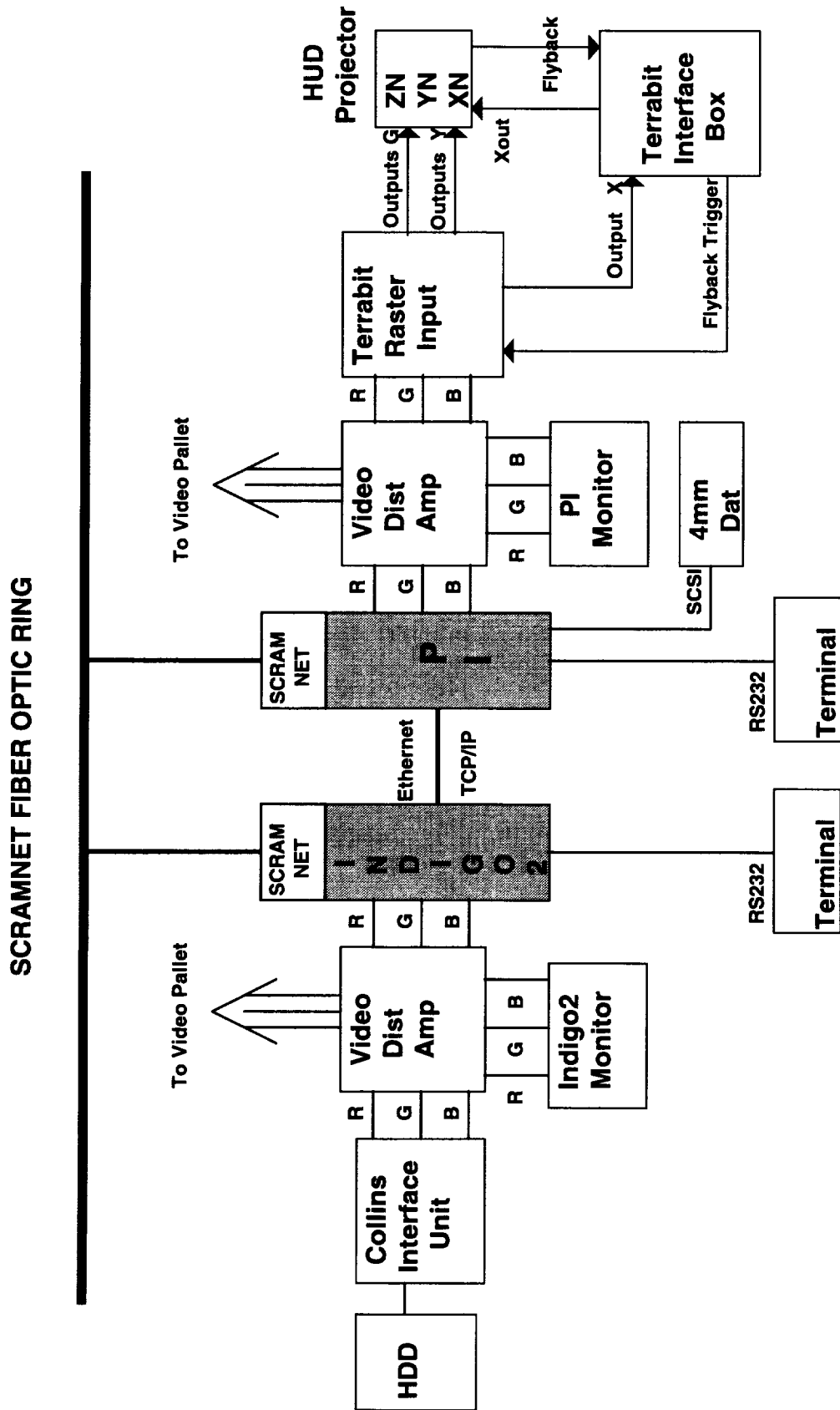






Figure 2 - Pilot Input Device

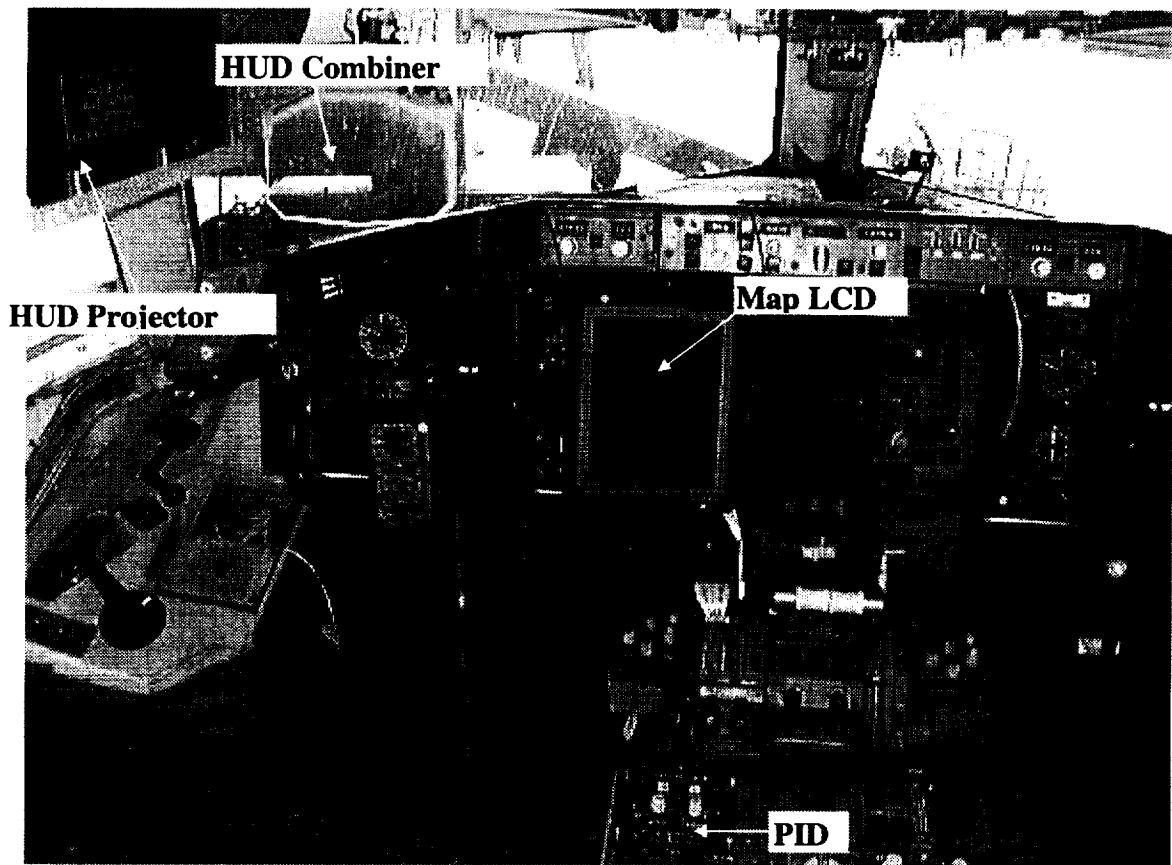
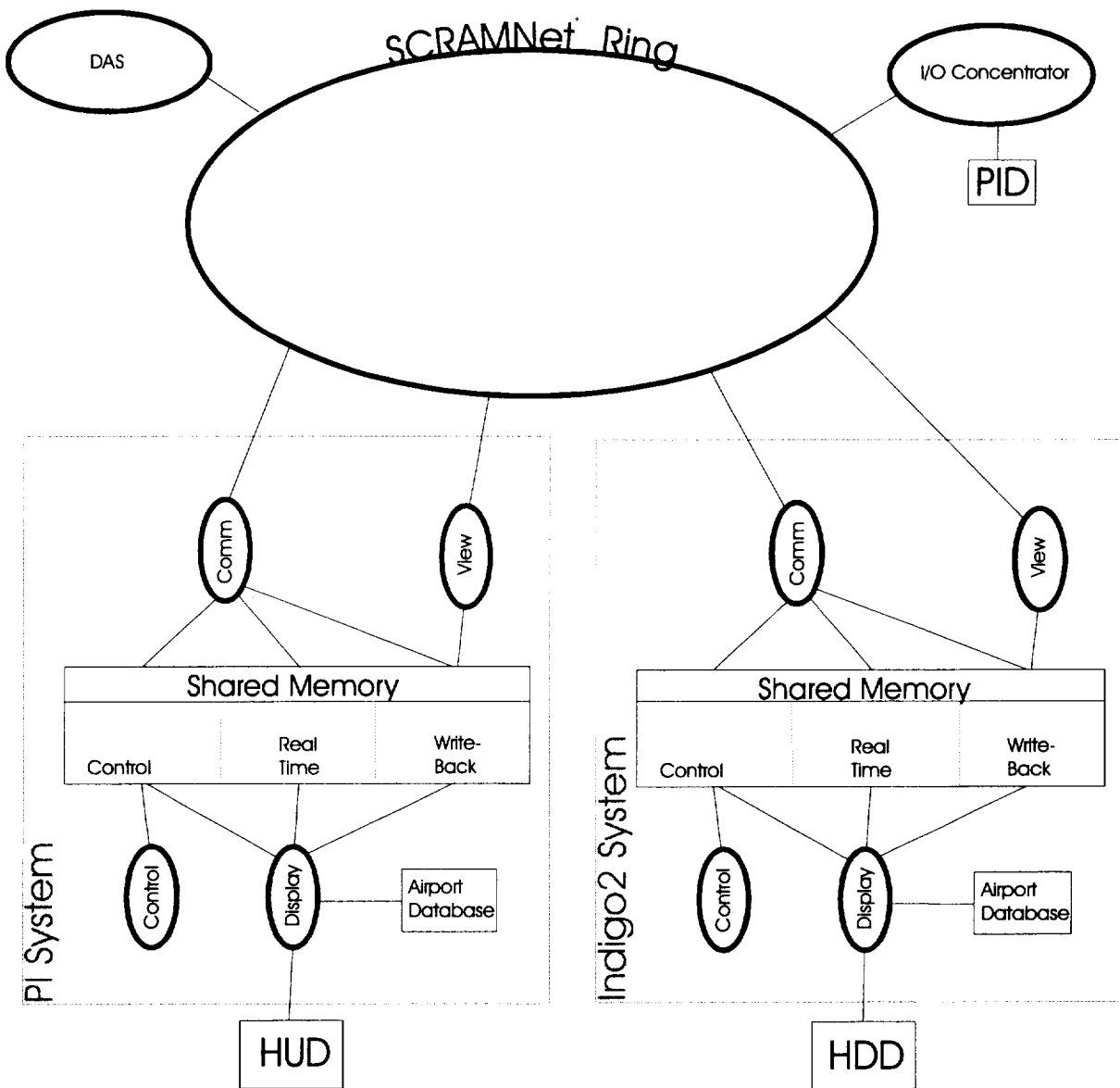
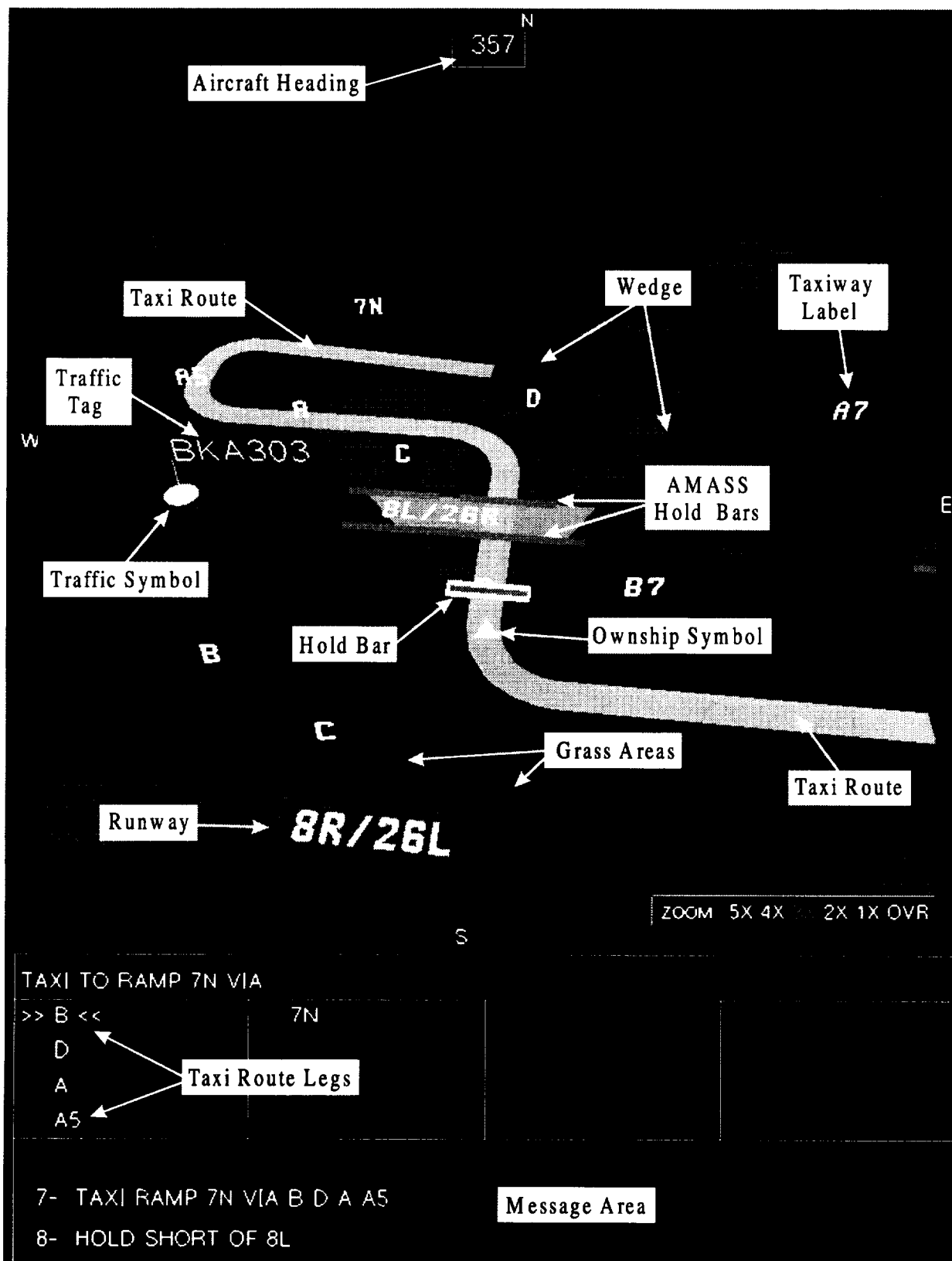


Figure 3 - NASA B-757 Flight Deck



**Figure 4 - High Level Software Architecture on Aircraft**



**Figure 5 - HDD Moving Taxi Map - Inbound Taxi**

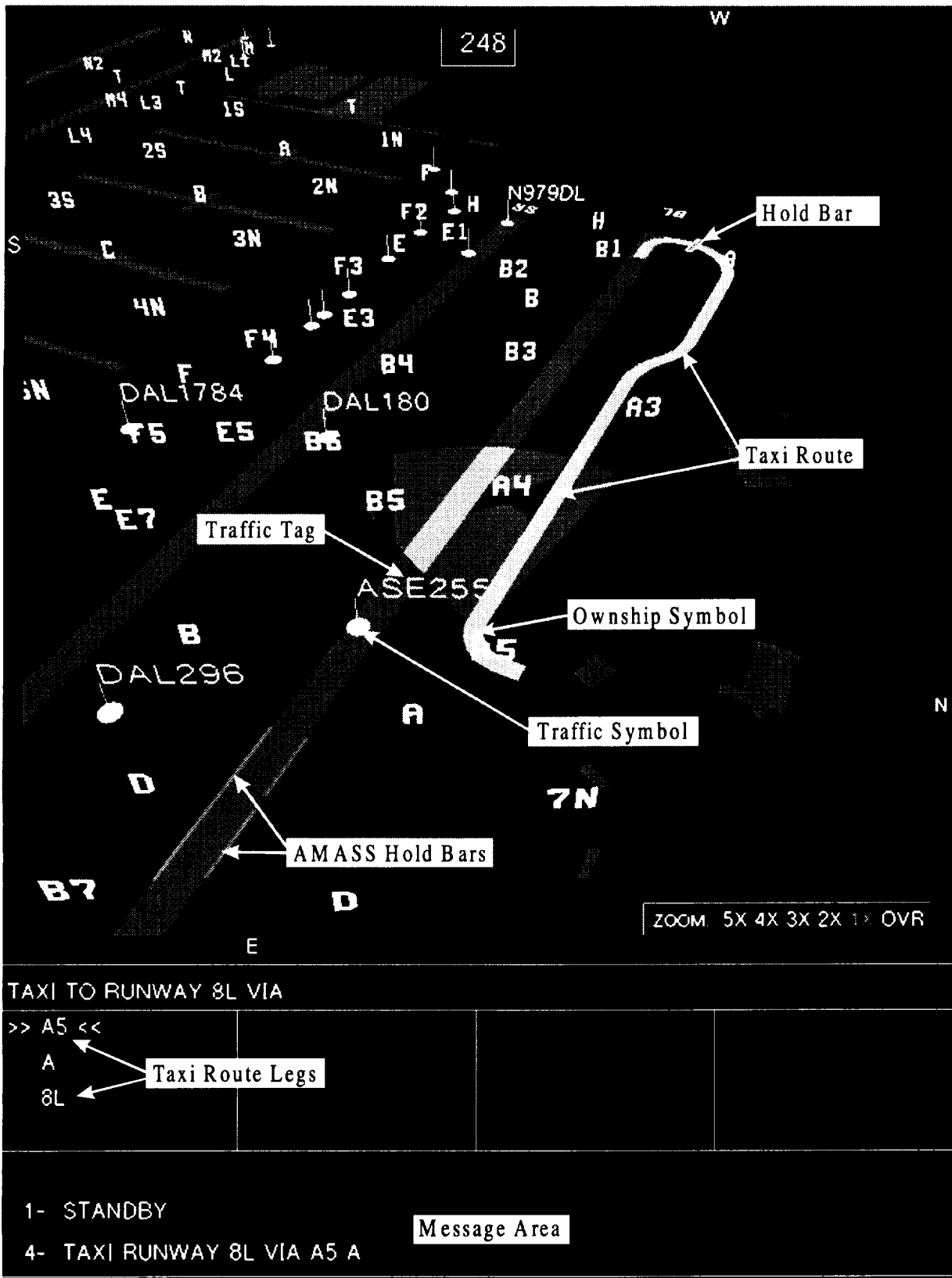


Figure 6 - HDD Moving Taxi Map - Outbound Taxi

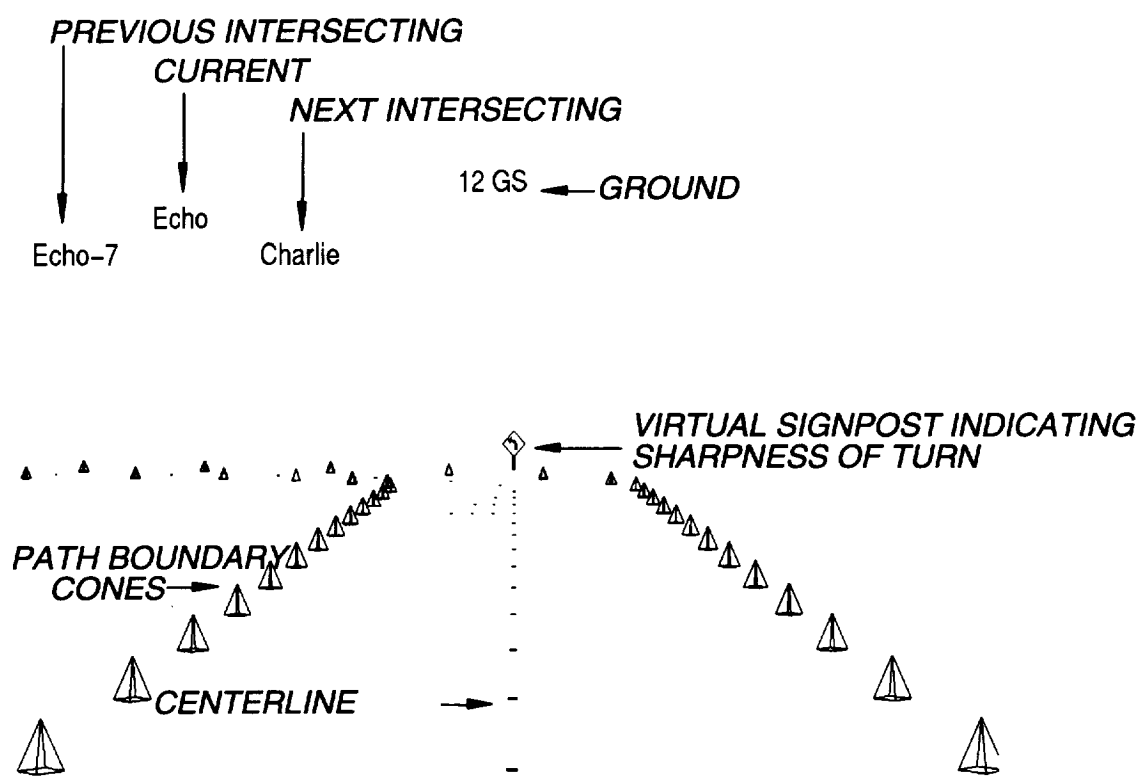
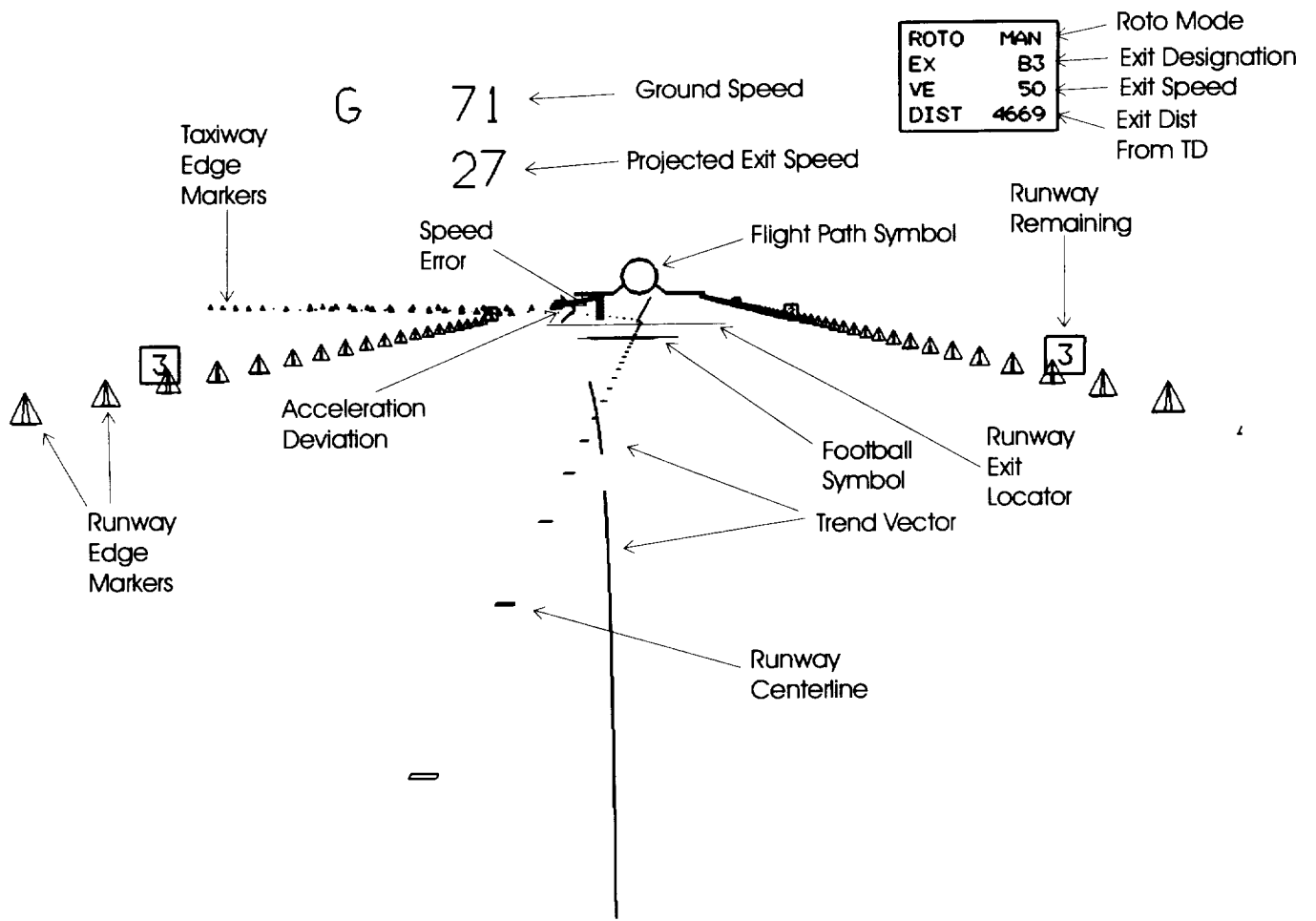
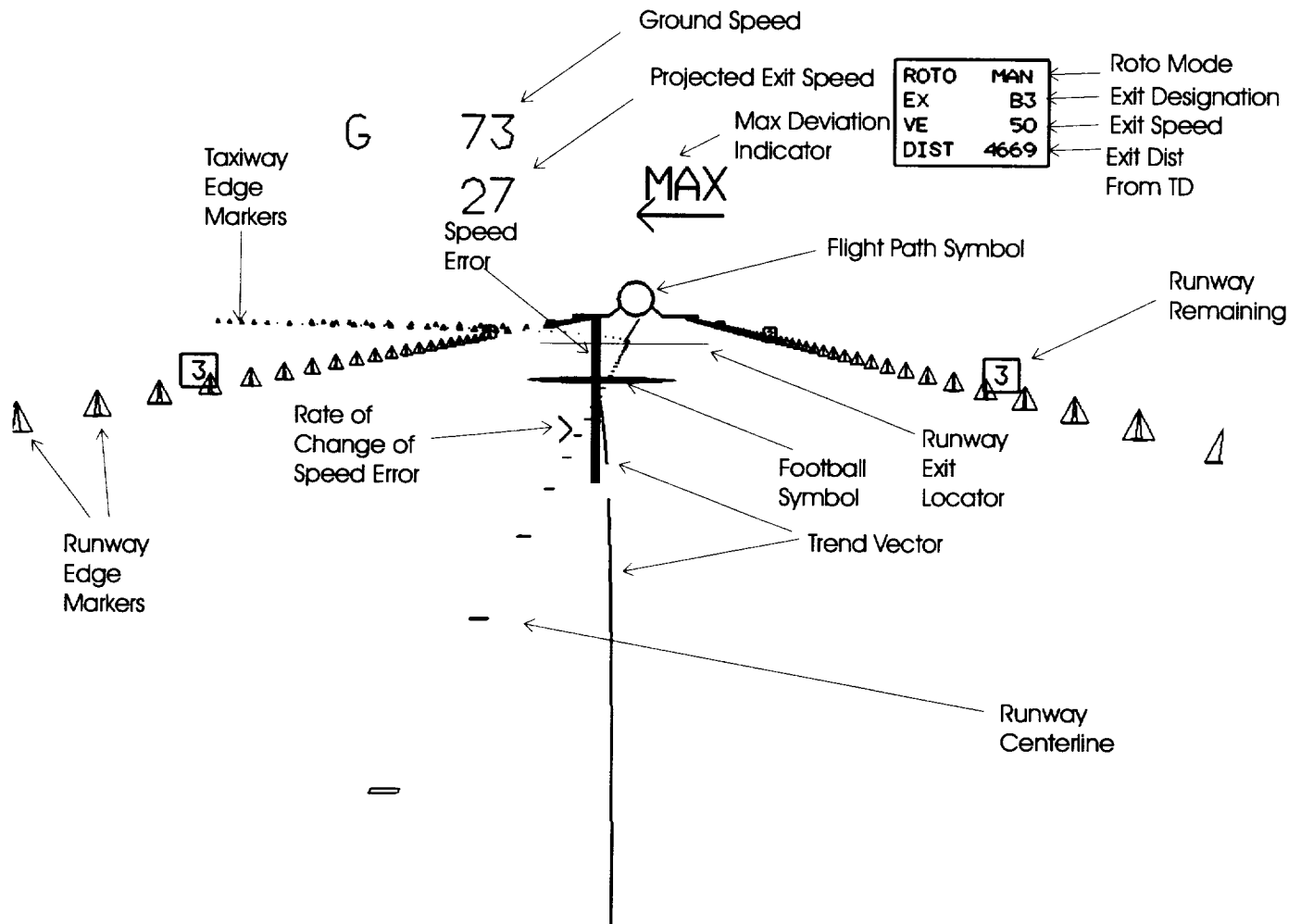


Figure 7 - Taxi HUD Symbolologies

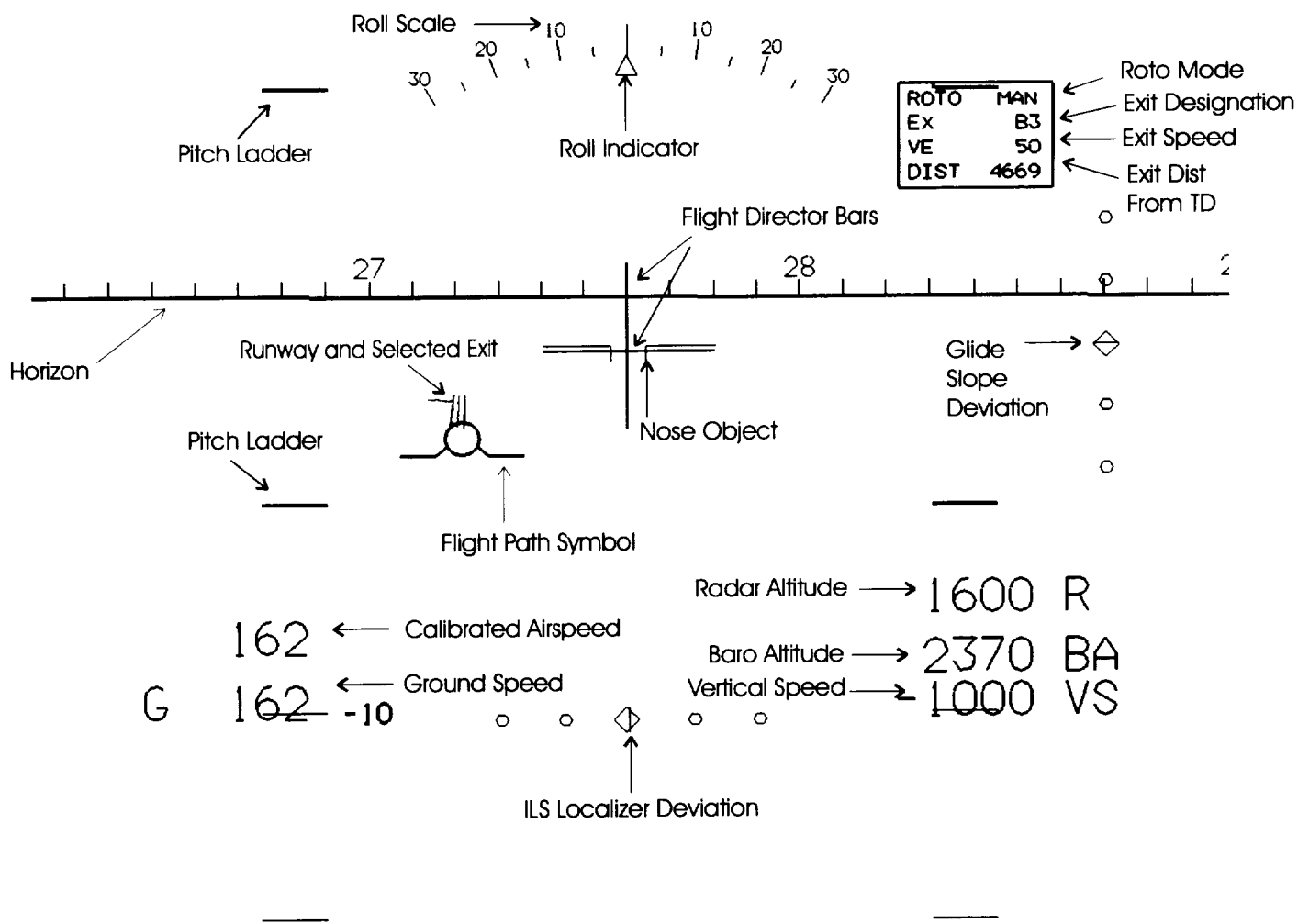


**Figure 8 - Example of Deceleration Guidance Display Format**

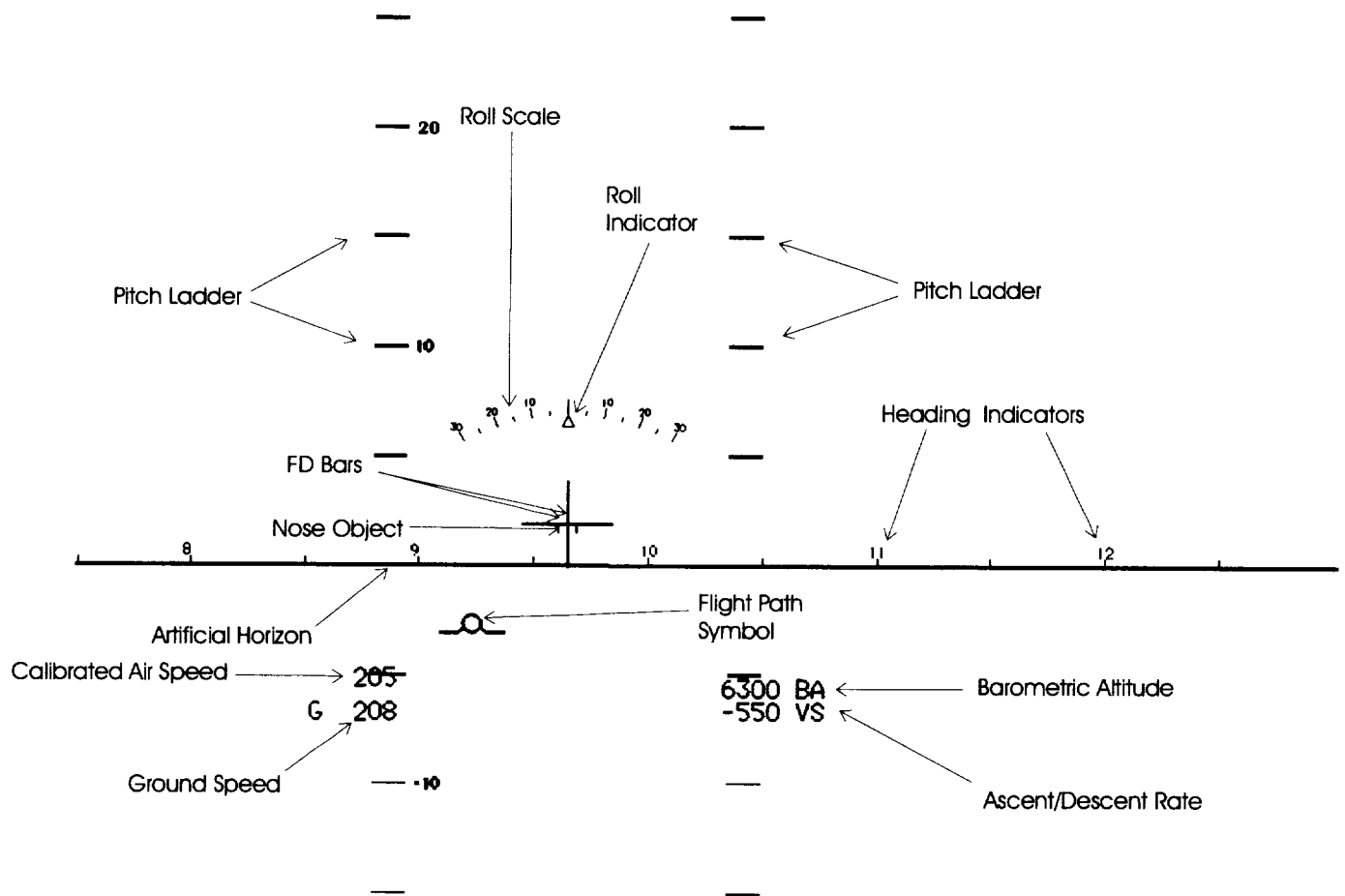


**Figure 9 - Example of Alternate Deceleration Guidance Display Format**

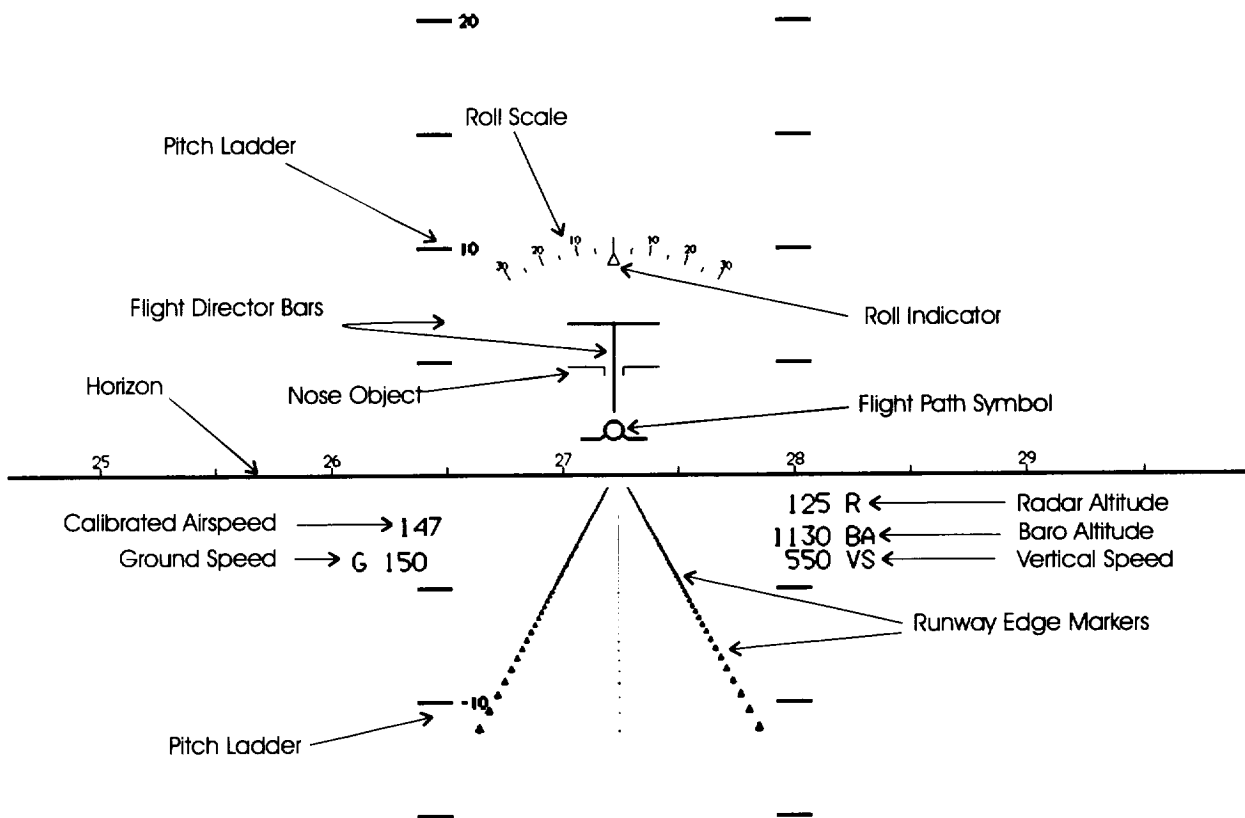




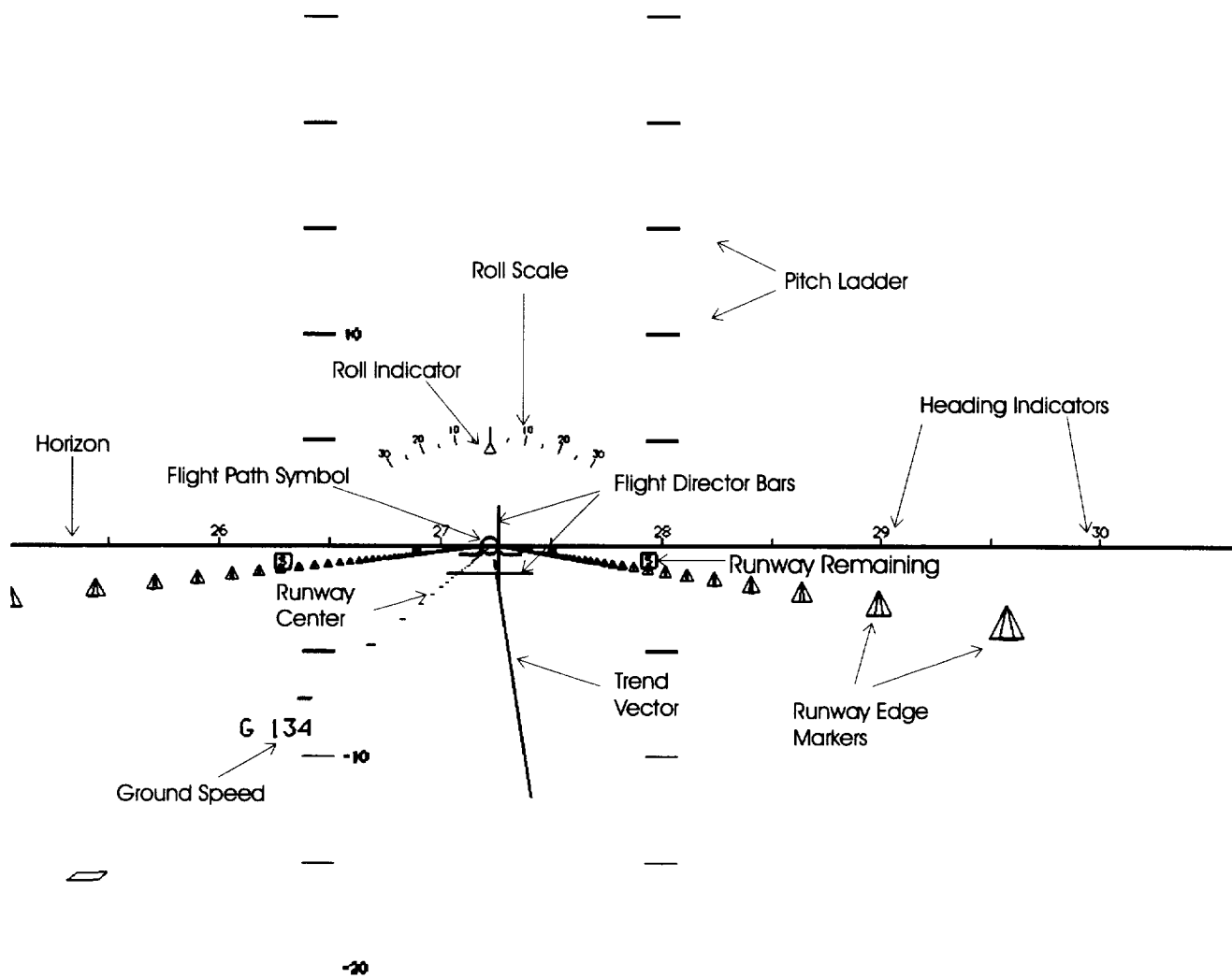
**Figure 10 - Example of Approach HUD Display Format**



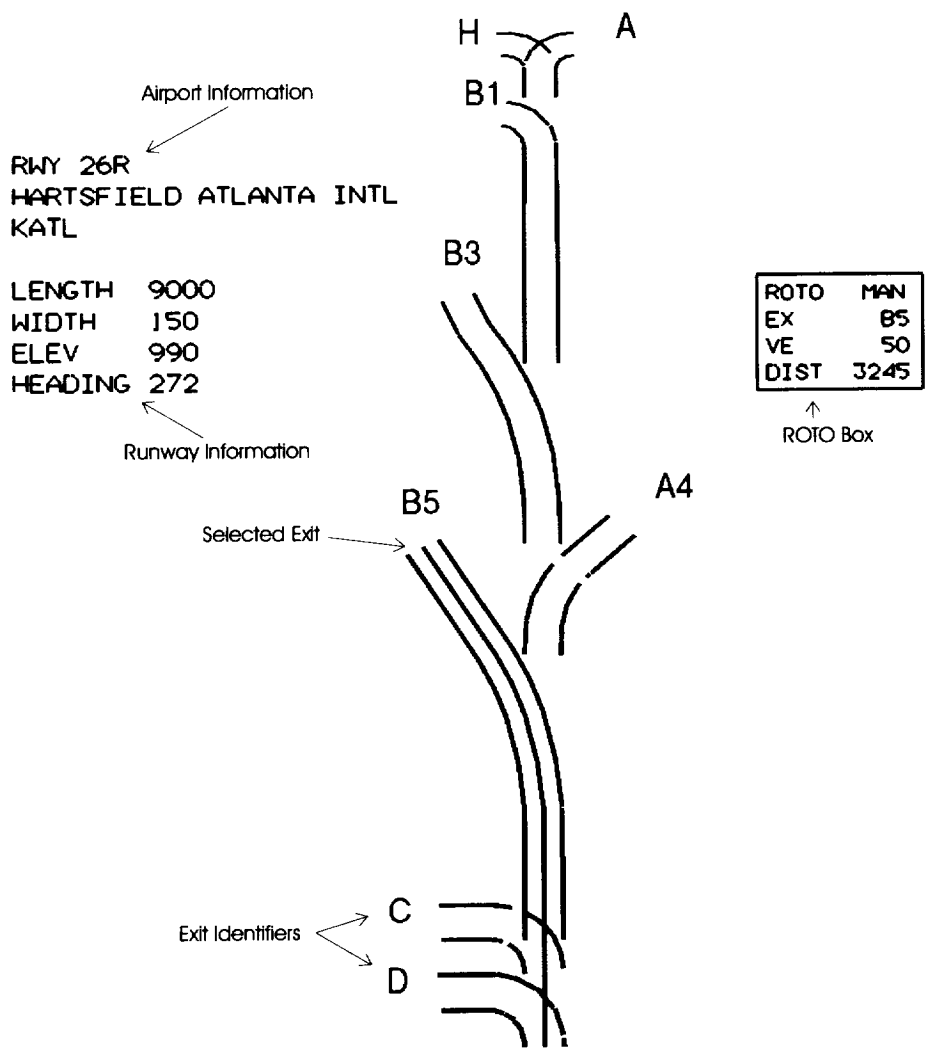
**Figure 11 - Example of Cruise Flight Display Format**



**Figure 12 - Example of Go-around Mode Display Format**



**Figure 13 - Example of Display Format During Takeoff**



**Figure 14 - Example of Display of all Defined Exits for Runway**

**Figure 15 - ATC Message Types**

<u>Type ID</u>	<u>Message</u>
<u>Uplink</u>	
1	STANDBY
2	REQUEST DENIED
3	ROGER
117	CONTACT [icaounitname] [frequency]
120	MONITOR [icaounitname] [frequency]
153	ALTIMETER [altimeter]
200	*HOLD SHORT OF [position]
212	*TAXI RUNWAY [runway] VIA [taxiroute]
219	*TAXI RAMP [ramp] VIA [taxiroute]
220	*CROSS [position]; CROSS [position] WITHOUT DELAY
221	*CONTINUE TAXI
223	*TAXI INTO POSITION AND HOLD
224	*CLEARED FOR TAKEOFF
<u>Downlink</u>	
1	UNABLE
3	ROGER
202	*TAXI DEVIATION
203	*TURNED OFF ON TAXIWAY [taxiway]
204	*TAXI DEVIATION RESOLVED

\* New messages defined not already in DO-219



REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE July 1998		3. REPORT TYPE AND DATES COVERED Contractor Report
4. TITLE AND SUBTITLE Integrated Display System for Low Visibility Landing and Surface Operations			5. FUNDING NUMBERS NAS1-96014 538-04-13-02	
6. AUTHOR(S) Sharon Otero Beskenis, David F. Green, Jr., Paul V. Hyer and Edward J. Johnson, Jr.				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Lockheed Martin Engineering & Sciences Langley Program Office Langley Research Center, Mail Stop 371 Hampton, VA 23681-2199			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration NASA Langley Research Center Hampton, VA 23681-2199			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA/CR-1998-208446	
11. SUPPLEMENTARY NOTES Langley Technical Monitor: Steven D. Young				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category 03                      Distribution: Standard Availability: NASA CASI (301) 621-0390			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This report summarizes the software products and system architectures developed by Lockheed Martin in support of the Low Visibility Landing and Surface Operations (LVLASO) program at NASA Langley Research Center. It presents an overview of the technical aspects, capabilities, and system integration issues associated with an integrated display system (IDS) that collects, processes and presents information to an aircraft flight crew during all phases of landing, roll-out, turn-off, inbound taxi, outbound taxi and takeoff. Communications hardware, drivers, and software provide continuous real-time data at varying rates and from many different sources to the display programs for presentation on a head-down display (HDD) and/or a head-up display (HUD). An electronic moving map of the airport surface is implemented on the HDD which includes the taxi route assigned by air traffic control, a text messaging system, and surface traffic and runway status information. Typical HUD symbology for navigation and control of the aircraft is augmented to provide aircraft deceleration guidance after touchdown to a pilot selected exit and taxi guidance along the route assigned by ATC. HUD displays include scene-linked symbolic runways, runway exits and taxiways that are conformal with the actual locations on the airport surface. Display formats, system architectures, and the various IDS programs are discussed.				
14. SUBJECT TERMS aircraft systems integration, simulation, flight test, flight deck displays, HUD communications, taxi map, roll-out and turn-off, guidance, situational awareness			15. NUMBER OF PAGES 64	
			16. PRICE CODE A04	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	