NASA/CR-1999-209731

Roll-Out and Turn-Off Display Software for Integrated Display System

Edward J. Johnson, Jr., and Paul V. Hyer
Lockheed Martin Engineering & Sciences Company, Hampton, Virginia

November 1999

The NASA STI Program Office ... in Profile

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

NASA/CR-1999-209731

Roll-Out and Turn-Off Display Software for Integrated Display System

Edward J. Johnson, Jr., and Paul V. Hyer
Lockheed Martin Engineering & Sciences Company, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NAS1-96014

November 1999

Available from the following:

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

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

CONTENTS

1.0 INTRODUCTION

This report describes the software products, system architectures and operational procedures developed by Lockheed-Martin in support of the Roll-Out and Turn-Off (ROTO) sub-element of the Low Visibility Landing and Surface Operations (LVLASO) program at the NASA Langley Research Center. The charter of the LVLASO program is to identify, develop, and demonstrate technologies that can safely maintain clear weather airport capacities in low visibility conditions. The ROTO portion of this program focuses on developing technologies that aid pilots in the task of managing the deceleration of an aircraft to a pre-selected exit taxiway. This report focuses on software that produces a system of redundant deceleration cues for a pilot during the landing roll-out, and presents these cues on a head up display (HUD). The same software produces symbology for aircraft operational phases involving cruise flight, landing, roll-out, and takeoff. It is part of a larger Integrated Display System (IDS) developed for the LVLASO project, which collects, processes and presents information to the flight crew on both a liquid crystal head-down display (HDD) and a HUD. A summary of the IDS software is available as a contractor report [1]. This report is one in a series of contractor reports intended to more fully document individual components of the IDS software.

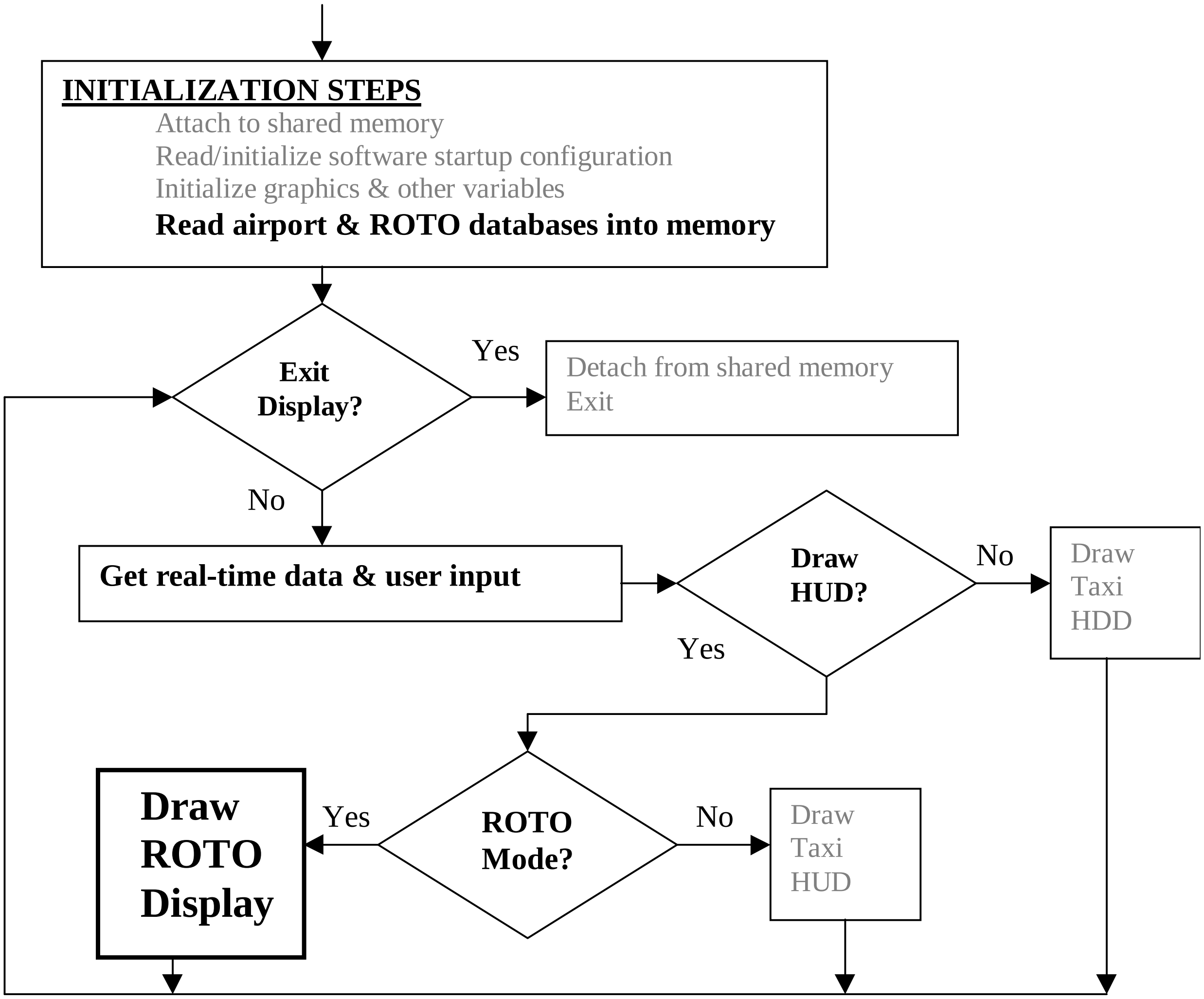
1.1 INTEGRATED DISPLAY SYSTEM OVERVIEW

The LVLASO program established program milestones to develop and flight test a set of prototype flight deck displays that provided guidance and enhanced situational awareness, in low visibility conditions, to a flight crew during all normal flight phases and taxi operations. The on-ground display formats were based on the results of previous flight tests at Atlantic City [2], and human factors studies conducted in the simulation facilities of the NASA Ames Research Center [3]. In-air symbology was adapted from previous research at LaRC and ROTO displays were developed in LaRC simulation facilities. The IDS software was successfully flight tested on board NASA’s Boeing 757 research aircraft (B-757) and demonstrated to more than one hundred invited guests at the Atlanta Hartsfield International Airport in August of 1997 [4,5,6].

The LVLASO program fused information on aircraft location from a Differential Global Positioning System (DGPS), air traffic control (ATC) messages, airport traffic data from Aircraft Surface Detection Equipment (ASDE-3) radar, aircraft identities from the Airport Traffic Identification System (ATIDS), and real-time information from the aircraft flight management computers with databases containing the physical layout of the airport. The IDS provided the architecture to manage all this information and present it to the flight crew. It provided software modules for drawing displays, computing algorithms, monitoring the current aircraft location and status, communicating data, handling of ATC messages, and processing surface traffic information. All data communications with the IDS software were through shared memory interfaces. Memory areas were established for aircraft data, uplinked data, and control data for interprocess communications. Data structures for the ownship state and global positioning information are presented in Appendix A. IDS communication modules were developed to received data from a local area network and distributed it to specific memory locations, in real time. Additional IDS software was developed that permitted operators to view, record and interact with memory contents at selected locations. These viewing and control programs were used extensively during flight and testing operations to verify data on the communications ring and change optional features of the IDS code without stopping the displays. The shared memory architecture was especially useful for development and debugging of the IDS system. Since the IDS was independent of the data sources, synthetic data could be placed in the shared memory to test specific features of the code, and real-time performance could be simulated by “playing back” recorded flight data at the appropriate rate.

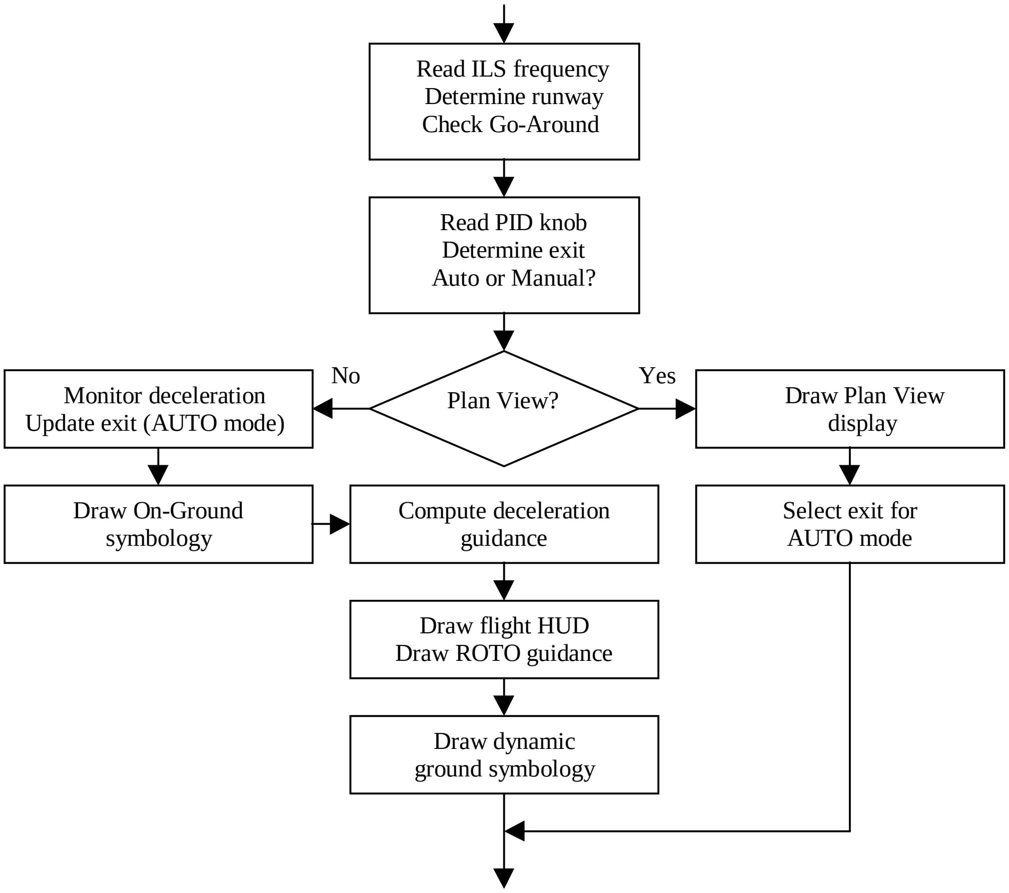
The IDS software was configurable to drive either the HUD or the HDD through a configuration utility. On board the B-757, one copy of the code was configured to drive the HUD and was executed on an SGI Personal Iris computer that supplied signals to the HUD. A second independent copy of the code was configured to drive the HDD and executed on an SGI Indigo 2 Maximum Impact computer. Data from the flight management system, the pilot, and other data sources were assembled by a device known as the I/O concentrator and transmitted over a SCRAMNet$^+$ communications ring to both machines simultaneously. IDS communications software, executing independently on each machine, established three shared memory areas on that machine and distributed data from the ring to them. Figure 1 presents a schematic diagram of the data communications and shared memory interfaces used on the B-757. The research data acquisition system (DAS) of the B-757 also received data over the SCRAMNet$^+$ ring. A pilot input device (PID) shown in figure 2 permitted the pilot to interact with the IDS software, select the desired exit taxiway, and control optional features of the displays.

The basic program structure of the IDS module controlling display functions was a "DO WHILE" loop that executed continuously until specifically terminated. The structure is outlined below:


INITIALIZATION STEPS
Attach to shared memory
Read/initialize software startup configuration
Initialize graphics & other variables
Read airport & ROTO databases into memory
Exit Display?
Yes
Detach from shared memory
Exit
No
Get real-time data & user input
Draw HUD?
No
Draw Taxi HDD
Yes
ROTO Mode?
Yes
Draw ROTO Display
No
Draw Taxi HUD

1.2 ROTO SOFTWARE ARCHITECTURE

The ROTO software was built into the IDS display module in two parts. At startup of the system, an initialization routine loaded a ROTO database into memory. This database contained the nominal exit speeds for all the ROTO exit taxiways and the physical dimensions of all the Atlanta runways. Detailed information about this database is presented in section 3.5. The main ROTO software was built into the do-while loop of the IDS display module. An outline of the ROTO structure is shown below:



Graphics routines associated with this module produced HUD displays for decelerating the aircraft, all flight operations, and selection of runway exit. The formats for these displays are presented in section 2.0. During normal ROTO operations, deceleration guidance to a single selected exit and conventional flight guidance were shown on the HUD. The pilot could select a special “plan view” display to show all ROTO exits for the current runway. At initialization, the computational steps to create the plan view display were executed to select an initial exit for the AUTO operating mode, but no display was shown. Other IDS graphics modules produced the HUD display for taxi operations. Software initialized the IDS to draw either a display from the ROTO module or the taxi HUD display. Program control was passed automatically from the ROTO module to the taxi HUD software, or vice versa, depending on the current phase of flight operations.

For each iteration of the main display loop, the ROTO display software produced one frame of graphics for output to the HUD. Display elements that did not change between frames (e.g. the roll scale) were drawn using OpenGL display lists to minimize processing times. The calculations necessary to draw the dynamic elements in the frame were performed in real time using the latest values in shared memory. This design permitted the computer driving the HUD to execute instructions at its "best rate" completely independent of other computers or the data update rates. The type of display to be drawn (e.g. ROTO approach and landing) was determined based on a mode parameter that was passed among the subroutines creating the display. The parameter was set automatically whenever the ROTO module was initialized. Aircraft parameters and the currently selected ILS frequency were used to select the current phase of flight, since the LVLASO experiment design did not include a mode switch for this purpose. For example, if the ROTO module was started while the aircraft was on the ground and received an appropriate Atlanta runway name, then a takeoff on that runway was assumed. Forced re-initializations of the ROTO module also occurred whenever the selected ILS frequency changed. Those re-initializations were used to transition the HUD display from takeoff to cruise mode or initialize the ROTO system for an approach.

1.3 ROTO SOFTWARE REQUIREMENTS

Development of the ROTO module began from a set of general requirements developed by NASA for the overall LVLASO experimental system. All graphics output from the ROTO module would be displayed on commercially available HUD hardware. The HUD hardware was purchased without software support, so the IDS supplied all graphics output and pilot interaction with the HUD. The Taxiway Navigation and Situation Awareness (T-NASA) format developed by the Ames Research Center was specified as the baseline taxi display format [3]. ROTO displays were constrained to produce runway edge markers in the T-NASA format and to seamlessly transition back and forth between the ROTO and taxi operations. The T-NASA format defined symbology for a single route of ground taxi which was drawn on the HUD in a perspective view so that its apparent location was conformal with actual locations on the airport surface. The IDS was constrained to use the T-NASA format for the Atlanta airport database. This two-dimensional database encoded the locations of runway and taxiway centerlines as a series of straight segments of variable length.

The LVLASO experimental design specified a HUD-based guidance system based on data from the aircraft flight management computers blended with data from a differential global positioning system. The blended data were used to determine aircraft status information and to locate the aircraft relative to a database of the airport physical layout. The ROTO portions of the LVLASO system had origins in previous NASA Langley Research Center projects [7,8]. The ROTO system was required to display the runway and exit taxiway selected by the pilot, provide guidance to this exit relative to a pre-selected deceleration strategy, continuously monitor performance relative to the strategy, 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 taxi operations. Pilots were to control the exit selection process and place the ROTO system in manual or automatic mode using a single rotary switch on the PID. In manual mode, the ROTO system was required to calculate and display guidance to the pilot's 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 were specified based on passenger comfort, which the ROTO system used 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 the allowable limits was necessary to reach the selected exit speed, and the ROTO system was in the automatic mode, it sequenced forward to the next ROTO exit taxiway and generated a new deceleration profile.

ROTO functions and display symbology were developed based on specifications jointly determined by NASA engineers and Lockheed-Martin personnel. Lockheed-Martin was tasked with developing the hardware and software architecture, communications interfaces, and computer graphics necessary to perform ROTO operations from an array of external data sources. NASA personnel were responsible for choosing the format of the displays, specifying aircraft control algorithms, and determining how the IDS would function. Final specifications for the ROTO module and the overall IDS evolved as the project matured and responses to pilot evaluations were implemented. LaRC simulation facilities were used extensively to develop and validate the ROTO system.

2.0 ROTO DISPLAYS

ROTO displays were controlled through a combination of pilot input, automatic criteria, and configuration settings. The pilot initiated a ROTO approach by tuning an ILS receiver to an Atlanta frequency and selecting an exit using the pilot input device (PID). The IDS was programmed to detect the current mode of flight operations and to automatically produce the appropriate HUD display. The remaining display control parameters were configured during the compilation and initiation processes, or from operator input. HUD symbology originally developed by Dr. Randall Harris at LaRC was selected as the baseline symbology for IDS in-air displays [9]. This symbol set provided much of the symbology commonly found on commercial HUDs. It included, among other things, a fuselage reference position, flight path object, artificial horizon, heading indicator, and roll scale. This basic set of symbology was supplemented with additional symbols and textual information for the LVLASO experimental architecture. The suite of ROTO displays is briefly presented below. More detailed information on individual display elements is provided in Section 3.0.

2.1 APPROACH DISPLAY

Figure 3 shows the ROTO approach HUD symbology. In this example, the aircraft was approaching runway 26R at Atlanta. The ROTO information box, in the upper right portion of the display, indicated that exit B3 was the selected exit. The target speed for this exit was 50 knots and a braking distance of 4669 feet was available following a nominal touchdown 1500 feet from the runway threshold. The ROTO MAN line indicated that the pilot was to manually decelerate the aircraft using guidance cues from the system. Symbology for the runway and exit taxiway showed that B3 was an exit to the left of the runway. The aircraft was on a heading of 276 degrees with an airspeed of 162 knots and ground speed of 162 knots. It was descending at 1000 feet per minute from a barometric altitude of 2370 feet and a radar altitude of 1600 feet. An approach requiring a crab angle was indicated by the displacement between the flight path symbol and the fuselage reference symbol. The location of the flight path object relative to the symbolic runway indicated that the aircraft velocity vector was aligned with the runway centerline and appropriate for a nominal touchdown. Flight director bars, referenced to the nose object, and ILS deviation scales also confirmed that the aircraft was on the glide path.

2.2 ROLL-OUT DISPLAYS

Following touchdown, the ROTO system automatically transitioned the HUD to display deceleration guidance to the exit. This guidance could be based on either of two primary concepts. The default method involved computing a target deceleration profile and providing the pilot with a visual depiction of the current speed error relative to the desired profile. The alternate method was based on projecting the aircraft speed at the exit, based on the current deceleration, and showing the pilot whether or not this speed was within acceptable limits. The display formats developed for these deceleration methods used a combination of symbology common to both displays, and symbology unique to a particular method. 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 on-board inertial reference units (IRUs). The current Universal Time Code (UTC) provided precise timing to the ROTO software for control of display elements and for making deceleration predictions.

2.2.1 Default Guidance

An example of the default deceleration guidance display is shown in figure 4. The speed error tape descending from the flight path symbol and the acceleration carrot provided guidance relative to the selected deceleration profile. A speed tape below the wing of the flight path symbol indicated that the current ground speed was slower than the profile speed. The location of the acceleration carrot, below the flight path object wingtip, indicated that the current deceleration exceeded the profile deceleration and therefore, the ground speed error would become worse. A speed error tape length of zero and the acceleration carrot pointing to the flight path symbol wingtip would have indicated current conditions matching the intended profile. In this case, the aircraft was projected to arrive at the exit location with a speed of 27 knots. The “goal line” drawn on the symbolic runway indicated the location of the start of the turn onto the taxiway and the “football” symbol indicated the surface location where a ground speed of 50 knots would be attained using the current deceleration. The location of the football, between the current position and the goal line, also confirmed that the aircraft would arrive at the exit below the nominal 50 knot exit speed indicated in the ROTO information box. Brake pressure or reverse thrust could have been reduced to move the football towards the goal line, raise the projected exit speed, and reduce the speed error. The segmented trend vector provided 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. The aircraft was located right of the centerline, but turning to the left at a rate which would bring it back on centerline in three to four seconds. Runway remaining signs, drawn on the HUD, indicated more than 3000 feet of runway remained from the current aircraft location. The word “TURN” was drawn on the HUD above the flight path symbol when projections indicated the aircraft was three seconds from reaching the goal line and decelerations were within acceptable limits. It flashed for 1.5 seconds, then was displayed steadily for five seconds before being removed.

2.2.2 Alternate ROTO Guidance

An example of the alternate deceleration guidance display is shown in figure 5. For this display, the guidance philosophy was to advise the pilot of the maximum acceptable exit speed and visually depict the results of maintaining the current deceleration until the exit. No deceleration profile was provided as pilots slowed the aircraft according to their own discretion. Except for the word “MAX” underlined with an arrow, the symbology used for this display was very similar to that of figure 4. However, the speed tape and acceleration carrot were used differently. The arrow symbol was positioned above the flight path symbol by an amount proportional to the difference between a target exit speed and a maximum allowable exit speed. Both the target and maximum exit speeds for an exit were individually tailored for the exit and read from the ROTO database. The length of the speed error tape indicated the projected difference between the exit speed and target speed, if current decelerations were maintained. Safe aircraft decelerations to the selected exit were assured if the end of the speed tape was located below the arrow symbol. Runway occupancy times were minimized if the pilot maintained a speed tape length between the flight path symbol wing and the arrow. The acceleration carrot provided an additional cue that 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 of the flight path symbol indicated an acceleration that would drive the end of the speed tape towards the bottom of the screen (more negative difference). In figure 5, both the speed error tape and the acceleration carrot were located below the wing. In the next few seconds, the predicted exit speed of 27 knots would have been reduced to an even lower value. Had the pilot reduced brake pressure and/or reverse thrust sufficiently to move the carrot above the wing, the speed tape would have shrunk towards the wing indicating that the predicted aircraft exit speed was converging towards the desired 50 knot value. The rest of the symbology was used in the same manner as on the default display.

2.3 FLIGHT DISPLAYS

As the capabilities of the ROTO module matured, it became apparent that displays for flight operations involving flight and takeoff operations could be developed using subsets of the available on-ground and in-air symbology. NASA engineers and Lockheed-Martin mutually agreed that a more realistic cockpit environment was achieved if the HUD and HDD operated continuously throughout a data run. HUD displays appropriate for takeoff and cruise were required in addition to displays for taxi and landing operations, since each ROTO data run involved a short flight in the airport traffic pattern. Existing capabilities of the ROTO software module were expanded to produce these displays. Figure 6 depicts the HUD display for cruise flight. This display was created from the normal approach display by suppressing the ROTO information box, the ILS deviation scales, and all on-ground symbology. It featured flight director bars, a pitch ladder, roll scale, artificial horizon, heading indicator and alpha-numeric speed, and altitude data. B-757 flight management computer commands, which positioned the flight director bars on electronic attitude indicators in the cockpit, were used to position the flight director bars on the HUD. This ensured consistent flight guidance. These commands were received by the IDS through the SCRAMNet$^+$ communications ring. The flight crew invoked the flight HUD display by selecting a "PARK" position for the captain's navigation receiver or selecting an ILS frequency which did not correspond to an Atlanta ILS approach. This action initiated an in-air restart of the system causing it to search the ROTO Atlanta database for an ILS frequency match. When no match was detected, the system automatically switched the HUD to the cruise symbology.

2.4 TAKEOFF and GO-AROUND

HUD symbology for takeoff operations utilized the full set of flight symbology superimposed on a subset of the ROTO on-ground symbology (see figure 7). Runway edge markers were drawn which outlined the current runway. The ROTO information box, football and goal line were suppressed, but trend vector and runway remaining signs were drawn. The takeoff display was automatically initiated when the taxi monitor program started the ROTO module on the ground, passed it a string variable with the name of the departure runway, and passed it a Boolean variable set to "TRUE" if the aircraft was on a runway. If a matching runway name was located in the ROTO Atlanta database and the aircraft was on a runway, the takeoff display was initiated. A HUD display for the go-around situation was automatically started if one of the flight crew pressed a go-around switch on the throttles (see figure 8). The display consisted of the full in-air symbology superimposed on runway edge markers which identified the location of the runway for the pilot. The ROTO information box and the taxiway edge markers were suppressed to de-clutter the HUD. Input from the ROTO PID knob was temporarily ignored.

2.5 PLAN VIEW OF RUNWAY EXITS

The ROTO module could also produce a special HUD display depicting all of the defined ROTO exits for the runway. The display was intended to provide the pilot with a graphical display of the available choices and the selected exit taxiway. An example is shown in figure 9. It provided a plan view of the runway showing all of the ROTO exits labeled with their appropriate letter designations. Textual information identifying the current airport and information about the intended runway was drawn on the left side of the display. The currently selected exit route was identified by a solid centerline and data in the ROTO information box were correspondingly updated. This temporary display was invoked when the flight crew momentarily placed the ROTO knob on the PID to 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 ROTO knob, as usual. While the display was active, each movement of the ROTO 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 selecting a new knob position before the countdown timer expired.

2.6 CURVED RUNWAY OPTION

One goal of the ROTO symbology was to improve pilot awareness of his/her runway location, relative to the selected exit taxiway, during the landing roll-out. The “goal line”, taxiway edge markers, and symbolic centerline were all intended to aid in the identification of the exit. They were produced on the HUD using a perspective view so that their locations would be conformal with “true” locations on the airport surface. However, the resulting symbology was small and indistinct from the background when the aircraft was on the ground and the exit was a considerable distance away. NASA engineers conceived of an experimental display format in which the on-ground symbology was conformal with the airport surface only within 600 feet of the aircraft. At distances greater than 600 feet, each piece of the on-ground symbology was purposely biased above the surface by an amount related to its distance to the aircraft. The resulting symbolic runway and on-ground objects were conformal and flat locally, but curved upward in the shape of a gentle parabola at greater distances (see figure 10). The “football” was also rotated to match the tangent of the arc. In simulations, the curved runway option did indeed improve the range at which the football and the selected exit could be positively identified during the landing roll-out. But there were obvious human factors issues associated with partially conformal symbology. The curved runway option could be applied to either the primary or the alternate deceleration guidance. It was retained in the flight version of the IDS code for possible evaluation, in a real-world setting, during pre-deployment testing.

3.0 TECHNICAL APPROACH

All software for the IDS was developed on Silicon Graphics computers using the C language, OpenGL graphics calls and X-Windows to facilitate future portability to other platforms. The IDS structure was based on a top down, object-based approach which allowed individual software modules to function independently. This section focuses on the technical aspects of the ROTO software module. An overview of the IDS software and the modules controlling communications, taxi displays, etc. is available as a contractor report [1].

Conformality of ROTO system display elements with the physical world required the use of two different graphics techniques. Objects intended to overlay features on the airport surface were drawn in a three-dimensional perspective view using a North and East database of the airport surface. The objects were positioned in the scene according to their true physical location. The graphics system used the physical viewing angles provided by the HUD hardware to determine what objects were in view as the aircraft moved through the virtual scene. A two-dimensional orthographic mode was used to render screen elements that conveyed information pertaining to flight operations and the ROTO deceleration guidance. The orthographic window was sized to match the solid angle subtended by the HUD combiner glass at the eye point of the pilot. Angular positions in the orthographic window then corresponded with true angles perceived by the pilot. Using this technique, guidance elements, such as the flight path object, could be positioned to correctly depict the aircraft flight path. A two-dimensional format was preferred for screen elements such as the ROTO deceleration symbology and the roll-scale which provided advisory information. Both graphics environments required calibration and alignment with the outside world. These procedures will be discussed in a later section.

3.1 ROTO SYSTEM OBJECTIVES

The primary objectives of the ROTO module were as follows:

- Compute and display aircraft deceleration guidance
- Continuously monitor the aircraft status to ensure safe comfortable decelerations
- Provide both automatic and manual modes of operation
- Respond, in real time, to changes during the landing and roll-out phases of flight.

These objectives were met through the development of tightly integrated computational and graphics software which operated on the most current aircraft status data available. Each frame of graphics output included a call to the computational software to compute the size and/or position of some deceleration symbology. Figure 11 presents a logic diagram of the main ROTO computational software. Functions in the computational framework were accessed to compute guidance parameters based on either a deceleration profile or projected speed at the exit, monitor the deceleration progress during the roll-out, and control the automatic exit selection process. The details of these operations are provided in the sections which follow.

3.2 ROTO ALGORITHMS

Fundamentally, the process of decelerating an aircraft involves dissipating kinetic energy between the touchdown point and the exit taxiway. There are an infinite number of strategies for successfully implementing the deceleration process. Each pilot has his/her preferred technique, and each landing is unique. To understand the ROTO deceleration guidance, it is necessary to consider the mathematics describing the deceleration process. It is a boundary value problem with constraints that result in a solution optimized for a specific purpose. The boundary values are determined by the aircraft speed at touchdown and the desired speed at the exit taxiway. Physical constraints, such as not exceeding the aircraft's ability to decelerate, impose limits on the possible solutions to this problem. Still, a mathematically infinite number of possible velocity and distance functions, U(x), can be found. However, additional constraints can be applied to produce a deceleration profile which is optimized for a particular goal. For example, the deceleration curve can be tailored to match a pilot's "natural" deceleration tendencies. Or it can be tailored for passenger comfort or minimum runway occupancy time. The primary goal of the LVLASO-ROTO project is to provide deceleration strategies which allow pilots to safely maintain airport traffic capacities in low visibility conditions.

The acceleration (a) at any point (x) on the runway is directly related to the speed (v) of the aircraft by the chain rule:

$$a = \frac{dv}{dt} = v\frac{dv}{dx} = \frac{d}{dx}(\frac{v^2}{2})$$

In this case, (x) represents the longitudinal position of the aircraft on the runway, with (x_0) being the touchdown point and (x_e) being the point at which the aircraft must begin to turn off the runway. The forward speed of the aircraft is v(x). Each possible deceleration strategy is described by an analytic function U(x) which satisfies, at a minimum, the boundary conditions:

$$U(x_0) = v_0 \ (\text{ touchdown speed })$$

$$U(x_e) = v_e \ (\text{ specified exit speed })$$

$$\left| \ (\frac{d}{dx}\frac{U^2}{2}) \ \right| < a_{\max} \ (\max \text{ allowed deceleration })$$

The default ROTO deceleration guidance format, (figure 4), uses the function U(x) to provide the pilot with a target deceleration profile, against which the actual aircraft deceleration may be measured. All of the computed deceleration profiles include two additional safety features. A variable length "brake buffer" is assumed just prior to the actual beginning of the turnoff. Thus the exit location (x_e) used in the following formulas actually lies ahead of the actual beginning of the turnoff. The length of the brake buffer is set according to the nominal exit speed and can be as much as 100 ft. The second safety feature involves a provision for the offset of the nose of the aircraft from the GPS position. This offset is included so that the computed deceleration guidance is referenced to the nose of the aircraft. For NASA's B-757, the nose of the aircraft is 75 ft from the GPS navigation reference point.

The deviation of the current aircraft speed from the deceleration profile at a particular point along the runway is:

$$dev(x) = v(x) - U(x)$$

This deviation is displayed as the "speed tape" extending from the wing of the flight path object. A current speed below the computed profile is indicated by a speed tape extending below the wing of the flight path object. The deviation of the aircraft acceleration from the profile value can also be calculated from the function U(x) and the actual track acceleration a(x):

$$adev(x) = a(x) - \frac{d}{dx}\frac{U(x)^2}{2}$$

The acceleration deviation is displayed as a "carrot" symbol (>) and also referenced to the wing of the flight path object.

For the initial LVLASO-ROTO system, NASA engineers selected four profiles to provide deceleration cues. These profiles were based on mathematical functions which could be readily calculated, and whose rate of change could also be easily calculated. The ROTO computational software could produce deceleration guidance based on any one of these profiles. The IDS control program provided a mechanism to change the active algorithm, at any time, without halting the program execution. This was accomplished by inserting a code for the algorithm choice into a special shared memory location which was continuously read by the ROTO software. The intent of this capability was to provide flexibility and an unlimited number of possible deceleration profiles. During flight operations, the algorithm choice was set once, prior to an approach, and never changed while on final approach or during the landing roll-out. A default algorithm was used if no algorithm was specified with the control program

3.2.1 Linear

The simplest function which will satisfy the constraints is a linear function of x:

$$U(x) = \frac{(v_0 - v_e)(x - x_e)}{(x_0 - x_e)} + v_e$$

The corresponding deceleration is:

$$\frac{d}{dx}\frac{U(x)^2}{2} = U(x)\left(\frac{v_0 - v_e}{x_0 - x_e}\right)$$

Since U(x) decreases monotonically, this deceleration has its greatest magnitude at the touchdown point.

3.2.2 Delayed Linear

The maximum deceleration is inversely proportional to the distance from touchdown point to exit point. Runway occupancy time could be reduced if the aircraft were to coast or brake lightly until a point was reached at which the deceleration necessary to achieve the proper exit speed reached some limit A_L. Two regimes are possible, depending on whether the limit has been reached:

$$U(x) = unspecified\ when \quad x < x_e + \frac{v_e}{A_L}(v(x) - v_e)$$

$$U(x) = \frac{v(x_L) - v_e}{x_L - x_e}(x - x_e) + v_e \quad otherwise$$

The limiting point (x_L) at which the constraint becomes active is given by:

$$\frac{v(x_L) - v_e}{x_L - x_e} = \frac{A_L}{v_e}$$

After the limiting point has been reached, the required deceleration is:

$$\frac{d}{dx}\frac{U(x)^2}{2} = \left(\frac{A_L}{v(x_L)}\right)^2\left(x - x_e + \frac{v(x_L)v_e}{A_L}\right)$$

In practice it proved necessary to impose a “latch” condition so that the triggering of the constraint could not be reversed.

3.2.3 Nonlinear

In actual aircraft deceleration situations, aircraft deceleration begins at a low level, then ramps up, and finally trails off as the exit is approached and the reverse thrusters are stowed. A non-linear rule was developed by NASA engineers in order to create a natural-feeling deceleration. The function U(x) is expressed as:

$$U(x) = v_0 - (v_0 - v_e)\xi \exp(\kappa(\xi - 1))$$

$$where$$

$$\xi = \frac{x - x_0}{x_e - x_0}$$

$$\kappa = \frac{v_0 - v_e}{v_0}$$

It can be seen by inspection that the end conditions are satisfied. The corresponding deceleration rule is:

$$\frac{d}{dx}\left(\frac{U(x)^2}{2}\right) = \frac{U(x)(U(x) - v_0)}{x_e - x_0}\left(\frac{1 + \xi\kappa}{\xi}\right)$$

The acceleration gradually intensifies with distance, then tapers off as the exit is approached

3.2.4 Constant Deceleration

If extreme deceleration is to be avoided, but the starting and ending speeds are specified, then constant deceleration would seem to be an optimal strategy. The speed profile and deceleration profile for this case are:

$$U(x) = \sqrt{\frac{{v_0}^2 - {v_e}^2}{x_e - x_0}(x_e - x) + {v_e}^2}$$

$$\frac{d}{dx}\frac{U(x)^2}{2} = \frac{{v_e}^2 - {v_0}^2}{2(x_e - x_0)}$$

3.2.5 Energy and Velocity Comparisons

Figure 12(a) shows a set of speed guidance curves for the four algorithms above under a typical landing scenario. Figure 12(b) shows the same information expressed in a different way, as the kinetic energy per unit mass for the four cases.

The linear profile has the lowest speed at any point between the ends. This suggests that it would have the lowest average speed, and therefore would consume the greatest time between touchdown and exit. The curve for the delayed linear case is shown as starting with a flat curve. This is not necessarily the case; the pilot has the option of applying some braking and intercepting the sloping line at a farther point. The delayed linear case can be seen to increase the average speed compared to the linear profile, but at the cost of a stronger initial deceleration. The curves for the nonlinear case and the constant deceleration cases are rather similar. Both curves tend to maintain a high average speed while satisfying the conditions at the end points. The steepness of the curves at the exit point is not necessarily detrimental, since it is the product of the speed times the spatial rate of change of speed which determines the acceleration sensed by the passengers. The curves in figure 12(b) illustrate the process of dissipating kinetic energy. The linear profiles reduce kinetic energy early on the path, while the other cases take longer. For the constant deceleration case, the kinetic energy curve is a straight line. The slope of these curves is the acceleration, and can be used to compare the cases qualitatively. It can be seen that the nonlinear case is “natural” in the sense that the greatest deceleration occurs somewhere in the middle of the path, although the exact location depends on the landing speed, the exit speed, and the distance traveled.

3.3 PREDICTED EXIT SPEED GUIDANCE

The alternative ROTO guidance (see figure 5) was based on visually depicting the results of maintaining the current deceleration until the exit. Instead of being shown the deviation of the current speed with respect to some desired speed, the pilot was shown his/her expected speed at the exit relative to a nominal exit speed. The prediction was updated each graphics frame and blended with the previous predicted value to produce a smoothly varying estimate. A speed tape extending above the wing of the flight path object indicated a predicted speed above the nominal speed. If the end of the tape extended above the arrow positioned over the flight path object, the current level of deceleration would not result in an acceptable exit speed. Increased decelerations were required, or if necessary, a new exit farther down the runway was selected. Both the nominal speed and the maximum permissible speed for each ROTO exit were encoded in the ROTO database.

Mathematically, the formula used in section 3.2.4 for constant deceleration was used in reverse to predict an expected exit speed given the current position and deceleration:

$$v(x_e) = \sqrt{(v^2(x) + 2a(x) \cdot (x_e - x))}$$

Where v(x) was the velocity at position (x) and a(x) was the current track acceleration. The subscript (e) indicated parameters associated with the exit. The following time filter was applied to the displayed length of the speed tape to dampen the effects of instantaneous changes in acceleration:

$$(v(x_e) - v_{nom})_{filtered} = e^{-\Delta t/\tau} \cdot (v(x_e) - v_{nom})_{previous} + (1 - e^{\Delta t/\tau}) \cdot (v(x_e) - v_{nom})_{current}$$

The computational interval (Δt) was the time interval between computation cycles. In cases where the computation cycle rate exceeded the data update rate, the previous filtered value was carried forward. The filter time constant (τ) was set to 2.0 seconds. These values provided a smooth but responsive operation of the speed tape so it could be used for effective aircraft control.

On this display, the carrot symbol (>) indicated the time rate of change of the projected exit speed. Its value depended on the current rate of change of the track acceleration. In practice, negative values of the expression inside the radical above were trapped out, and the projected exit speed was set to zero. An explicit differencing scheme was employed to determine the time rate of change of the predicted speed. The displayed value was time filtered in the same manner as the speed tape except that a filter time constant of 1.5 seconds was used.

3.4 DYNAMIC SCREEN ELEMENTS

3.4.1 Horizon Line and Attitude Indication

In most operating modes, the HUD display included an artificial horizon. This display was intended to indicate a level flight attitude. When the aircraft was on the ground, this horizon line overlaid the actual horizon, but at altitude, Earth curvature effects caused the true horizon to appear below this artificial horizon. Compass directions were indicated numerically just above the horizon line, in ten-degree increments shown to two figures. Tic marks indicated five-degree intervals. The horizon line anchored a pitch ladder display, consisting of horizontal tic marks to the left and right of center, in five-degree increments, with numerical indicators at ten-degree intervals. Pitch attitude and magnetic heading of the aircraft were indicated by the position of the "nose object" relative to the horizon line and pitch ladder. The pitch attitude of the aircraft was indicated by the perpendicular displacement of the nose object from the horizon line, and the aircraft magnetic heading was shown by the compass direction lying closest to the center of the nose object. The nose object was "fixed" at the center of the HUD and represented the boresight of the aircraft fuselage. The horizon line and pitch ladder were positioned on the HUD according to data received from the flight management system through the shared memory interface. Their locations were updated each graphics frame. This arrangement was necessary in order to achieve conformality with the world outside the cockpit window.

A roll scale was located above the nose object, and, like the nose object, was immovable on the HUD. A pointer, which slid along the roll scale, indicated the aircraft roll attitude according to data from the flight management system. This gave a more quantitative indication of roll than did the tilt of the horizon line.

3.4.2 Flight Path Object

A symbolic aircraft known as the "flight path object" was used to display the inertial vector of the aircraft on the HUD. This symbol followed the original concept of Bray [10] and Harris [9], and featured a circle with a pair of stubby wings to indicate the aircraft's true path through space. This path normally differs from the boresight of the fuselage due to cross winds and aircraft flight characteristics. The relative position of the flight path object with respect to the artificial horizon and the heading scale indicated the aircraft inertial vector. The flight path angle determined the perpendicular displacement from the artificial horizon. This angle was computed for every graphics frame as the inverse tangent of the ratio of vertical speed to the ground speed. The aircraft track angle was used to position the flight path object with respect to the compass scale on the horizon line. All three input parameters were received directly from the flight management system and updated continuously. The flight path symbol remained upright in the aircraft frame of reference. Thus, it also correctly indicated the roll attitude of the aircraft with respect to a level flight attitude.

In order to help de-clutter the display after touchdown, the flight path object was locked into an elevated position as soon as the ground speed fell to 80 kt. The vertical position was 0.5 degrees above the horizon line. In this position, the flight path object functioned as a reference body for the speed tape and acceleration carrot.

3.4.3 Football and Goal Line

The "football" and "goal line" analogy for these display elements arose naturally from the shape of the symbology and its use. The "football" was a two-dimensional oval with semi-major axis aligned with the runway which slid along the centerline to indicate the position at which the target exit speed would be attained if current braking was maintained. The "goal line" was a static line drawn across the symbolic runway at the start of the turn onto the exit taxiway. Adjusting the current deceleration to maintain the football near the goal line, meant the aircraft would arrive at the start of the turn at the pre-selected exit speed. This guidance could be used independently to manage the entire deceleration phase, but its intended purpose was to provide guidance for "fine tuning" the aircraft deceleration when the exit was within a few thousand feet. The size of the football was set according to the runway width contained in the ROTO database. Its semi-major and semi-minor axes were 20 percent and 12.5 percent of the runway width, respectively. The football was constrained to slide along the centerline of the symbolic runway regardless of the aircraft location. This prevented the football from swinging on and off the runway as the aircraft maneuvered to track the centerline. However, it was free to slide beyond the ends of the runway if the current braking would not adequately decelerate the aircraft.

The DGPS position was used to determine the aircraft location (x) on the runway. For the B-757, an offset of 75 feet longitudinally (l_{nose}) was used to account for the displacement of the aircraft nose from the location of the DGPS reference position. The football was then positioned relative to the runway threshold using the following constant deceleration model to calculate the displacement:

$$d(x) = \frac{(v_{nom})^2 - v(x)^2}{2 \cdot a(x)} + (x - x_{threshold}) + l_{nose}$$

The exit speed (v_{nom}) was determined from the ROTO database using indices that identified the selected exit. Anytime the selected exit changed, the exit speed was automatically updated. The longitudinal acceleration and the ground speed were updated each graphics frame with information from the flight management system contained in the real-time shared memory area. Thus, the raw update rate for the football location was equal to the slower of the graphics frame rate or the data update rate. To avoid this machine dependent behavior, NASA engineers specified that the location of the football be time filtered according to the relation:

$$\mathrm{d}(x)_{\text{filtered}} = e^{-\Delta t/\tau} \cdot d(x)_{\text{previous}} + (1 - e^{-\Delta t/\tau}) \cdot d(x)$$

The computation interval ($\Delta\tau$) was set to 0.03125 sec, the rate at which data were updated in the shared memory area. A time filter (τ) of 1.0 seconds, produced a responsive football location that appeared to move smoothly along the centerline.

The location of the goal line was determined whenever the graphics software produced a new symbolic runway with exit taxiway. The exit location was set to the point where the centerline of the path to the exit taxiway deviated from the runway centerline by 5 degrees or more. Because of discontinuities in the construction of the Atlanta database, a two step process was required to determine this location. The ROTO database was accessed to determine the approximate location of the turn onto the exit taxiway. The goal line was placed at the beginning of the first exit path segment that met the centerline deviation test and which was located within 300 feet of the anticipated exit location. This allowed the IDS-ROTO code to ignore anomalies in the Atlanta database unless they occurred reasonably close to the exit anyway. At worst, the goal line could be positioned slightly ahead of the true exit position. The taxiway edge markers and the taxiway centerline were not affected by this procedure and continued to indicate the true location of the exit. A safe exit strategy was assured at a cost of higher than necessary decelerations and slightly increased time on the runway.

3.4.4 Trend Vector

The expected ground track over the next four seconds was shown on the HUD as a segmented ribbon known as the trend vector. The radius of curvature, length, and position of the trend vector were computed each graphics frame, using the current aircraft position, ground speed, and rate of turn. These calculations produced North and East components (relative to the Atlanta database origin) which were used to draw the trend vector. It was rendered on the HUD in the same perspective as the rest of the conformal symbology. Hence, it appeared as a line on the airport surface extending forward of the plane and changing length and curvature as the aircraft moved. The first segment of the trend vector indicated the aircraft ground track over the next two seconds, and the second segment indicated the expected aircraft track from 2.2 to 4.0 seconds. If the plane were moving in a straight line, this ribbon would extend straight in front of the plane. If the plane were turning, the ribbon would be a curved arc whose radius of curvature was inversely proportional to the rate of turn of the aircraft. In either case, its length would be proportional to the speed of the aircraft, and would visibly shorten as the aircraft slowed. The track of an aircraft as it moves through an arc is shown in figure 13. The total change in direction, Ψ, is also the angle subtended by the arc traversed. Given an initial speed v and rate of turn $d\Psi / dt$, which were obtained from the real-time data stream, an extrapolation is made to a forward time t. The horizontal distance traveled, s, and the angle turned, Ψ, are given by:

$$s = vt$$
$$\Psi = t\frac{d\Psi}{dt}$$

Since the distance traversed is known, the radius of curvature of the arc can be calculated. The rate of change of aircraft heading is, by convention, positive in the clockwise sense. Using for convenience a Cartesian system in which x represents the displacement straight ahead of the aircraft and y is the displacement toward the left, the components of the displacement can be calculated from R and Ψ:

$$x_1 - x_0 = R\sin(\Psi)$$
$$y_1 - y_0 = R(\cos(\Psi) - 1)$$

The quantity R can be eliminated using the relationship:

$$R = \frac{s}{\Psi} = \frac{v}{\frac{d\Psi}{dt}}$$

where the denominator is nonzero, leaving an expression for displacement as a function of time and of the present speed and turn rate:

$$x_1 - x_0 = \frac{v}{\frac{d\Psi}{dt}}\sin\left(t\frac{d\Psi}{dt}\right)$$
$$y_1 - y_0 = \frac{-v}{\frac{d\Psi}{dt}}\left(1 - \cos\left(t\frac{d\Psi}{dt}\right)\right)$$

Since the turn rate is a signed scalar, these expressions are valid for both right- and left-hand turns. In the case of zero turn rate, the radius of curvature is infinite and the track is a straight line:

$$x_1 - x_0 = vt$$
$$y_1 - y_0 = 0$$

The final step before plotting was to transform these displacements from an aircraft-centered coordinate system to a runway-oriented reference frame, by means of a rotation about a vertical axis.

3.4.5 Runway Curvature

The curved runway display was conceived as an experimental format to increase the range at which the exit location, football and goal line could be positively identified. When the aircraft was on the ground and curvature was invoked, pieces of the ROTO on-ground symbology became dynamic display elements whose vertical position changed as a function of distance from the aircraft. Objects within 600 feet were drawn to be conformal with the airport surface features, as usual. At greater distances, the individual edge markers, each runway remaining sign, the football and the goal line were purposely biased above the surface according to the relationship:

$$\mathrm{d} \geq 600: \quad z = a + b \cdot d^2$$

$$a = -1.08, \quad b = 0.000003$$

$$\mathrm{d} < 600: \quad \mathrm{z} = 0.0$$

For this work, a parabolic curvature was selected and the coefficients were chosen to produce a vertical bias of 3 feet in the first 1000 feet of distance from the aircraft. The (a) coefficient was selected to produce a (z) of zero at a distance of 600 feet. When the aircraft moved, objects appeared to gently lower to the surface as the aircraft approached them. Curvatures of virtually any shape could be produced by using polynomials of higher degree.

In addition to being raised vertically, the football was rotated to be tangent to the curve. This maximized the visibility of the football and prevented its disappearance, due to being “edge-on”, as the vertical offset neared the height of the pilot eye point above the ground. Differentiation of the curvature polynomial produced an equation for the required rotation angle that was evaluated dynamically as the football location changed. The football itself was not curved. It remained a two-dimensional oval-shaped disk regardless of location.

Use of the curved runway option consumed a great deal of processing cycles since hundreds of trigonometric and square root functions were evaluated for each frame of graphics. The use of OpenGL display lists to draw on-ground symbology was also precluded since pieces of the symbology changed locations each frame. This seriously degraded graphics performance. Since the curved runway option was an experimental format, it could not be invoked by the pilots or operators of the IDS system during flight operations. It was invoked by editing the source code to set a boolean switch prior to compilation of the executable module.

3.4 AUTOMATIC EXIT SELECTION

The capability to automatically select a runway exit and to automatically change the current exit, if necessary, was a basic design objective of the ROTO system. Meeting this objective required different modes of operation depending on whether or not the pilot had selected the AUTO mode and whether or not the aircraft had landed. When the AUTO mode was enabled, the ROTO system used preset deceleration limits to make an initial exit selection. It then continuously monitored the aircraft status to ensure a safe comfortable deceleration to this exit. If excessive decelerations were required, the selected exit was automatically sequenced forward, one ROTO exit at a time, until a satisfactory exit was found or until the last possible exit. In manual mode, the ROTO system could not change the selected exit. It could only monitor the deceleration process and suppress display functions associated with turning onto an exit taxiway if excessive decelerations were required. Different monitoring and initialization procedures were required for airborne and ground operations.

All exit selection logic and monitoring functions were based on computing the average deceleration required to attain the nominal exit speed at the selected exit, given the velocities and distances involved. The basic formula was:

$$\overline{a} = \frac{\left(v_e^{\,2} - v_0^{\,2}\right)}{2 \cdot \left(x_e - x_0\right)}$$

Where (v_e) and (x_e) were the nominal exit speed and location. The speed and aircraft location were (v_0) and (x_0), respectively. The average required deceleration was calculated, every graphics frame, both before and after touchdown. If the required average deceleration exceeded the deceleration limit, then so did the instantaneous deceleration at some point. In this case, the desired limit was set for passenger comfort and to remain well within the deceleration capabilities of the aircraft. NASA engineers selected different deceleration limits and different stopping criteria for airborne and on-ground operations.

If the aircraft was airborne, a mild deceleration limit of 6.5 ft/sec^2 was used as deceleration limit. This was a conservative value that allowed for some error in the touchdown point and the possibility of braking action less than anticipated. Above 400 ft of radar altitude, an assumed maximum touchdown speed of 139 knots and a nominal touchdown point 2000 ft from the runway threshold were assumed. The longitudinal component of the wind speed and a 5.9 knot allowance for the flare maneuver were subtracted from the assumed touchdown speed. Below 400 ft the actual ground speed minus a 5.9 knot allowance for deceleration during the flare maneuver was used in the calculations.

If the aircraft had landed and was rolling out, a different limit strategy was applied. Since rapid decelerations could be generated during the testing of brakes and the process of deploying thrust reversers, it was not practical to sequence the exit selection forward for momentary decelerations above the limit. Instead, a requirement that the average deceleration exceed the selected limit for more than one second was applied. The deceleration limit was also increased to 10 ft/sec^2 since there was no longer any uncertainty about the touchdown point.

A logic flag indicating whether or not the deceleration criteria was satisfied was updated every time the average deceleration was computed. If the ROTO system was in the AUTO mode, this flag was used to sequence the selected exit forward. The word “NEXT” was flashed on the HUD for 3.0 seconds as a cue to the pilot that the exit selection had changed. If the selected exit was the last available ROTO exit, the word “LAST” was flashed on the HUD instead. The last ROTO exit was always at the end of the runway. If the system was in the manual mode, the logic flag was used to suppress the drawing of the “TURN” cue and inhibit the automatic de-clutter of ROTO symbology as the aircraft approached the exit. When calculations indicated the aircraft was three seconds from the start of the exit taxiway turn and the deceleration was within limits, the monitoring process was terminated.

3.5 ROTO DATABASE

All of the airport specific data used by the IDS software were stored in databases that were accessed as required. This resulted in a generic IDS executable code that could be applied to any airport for which the databases had been created. The three-letter airport identifier was used as a key to determine which data sets to use. It was input as a command line argument when the IDS was started. The names of the databases to be used and their location were also keyed to the three-letter airport identifier so they could be uniquely identified from the command line input. The full IDS software system used eight different databases. They contained information for decoding taxi route instructions, known holding locations, and a variety of other data. More information about these databases is available as a contractor report [1].

The ROTO module of the IDS software made use of a database developed specifically for ROTO deceleration applications, and the airport surface database. The remainder of this section outlines the details of the ROTO database. At initialization of the IDS software, this database was loaded into a C data structure. The format of the structure is presented in Appendix B. For the current airport, the structure contained the following information:

AIRPORT NAME	
IDENTIFIER CODE	
LATITUDE	**LONGITUDE**
AIRPORT ELEVATION (MSL)	
NUMBER OF RUNWAYS	
SELECTED RUNWAY	**SELECTED EXIT**

For each runway, information commonly found on ILS instrument approach charts was included in the structure along with the number of exits available for ROTO operations. The numeric part of the path file name was included to serve as a key for constructing the ROTO symbolic runway (see section 3.6). The following information was included:

RUNWAY IDENTIFIER			
THRESHOLD LATITUDE	**THRESHOLD LONGITUDE**		
LENGTH	**WIDTH**	**SLOPE**	**HEADING**
ILS FREQUENCY	**GLIDE SLOPE (deg)**	**TCH(threshold crossing height)**	
THRESHOLD ELEVATION (MSL)			
NUMBER OF ROTO EXITS	**IDENTIFIER FOR PATH FILE**		

A critical feature of the ROTO guidance system was that the data structure contained exit information uniquely tailored for each ROTO exit taxiway. Specific data regarding the geometry of the exit, its location, and the transition to taxi guidance were encoded, in addition to the nominal speeds for ROTO operations. Encoding data this specific about each exit enabled a ROTO deceleration system that was extremely precise and highly flexible. The speed information could be optimized to achieve specific operating goals. For example, the nominal exit speed could be selected for passenger comfort, for minimizing the runway occupancy time, or any other criteria. Its value could be set according to experience, with a particular exit, or set to maintain lateral forces during the turn below a desired limit. The following types of information were included:

EXIT NAME	
NOMINAL EXIT SPEED	**MAXIMUM PERMISSIBLE SPEED**
WIDTH	**ANGLE RELATIVE TO THE RUNWAY**
DISTANCE FROM THRESHOLD	**TRANSITION TO TAXI GUIDANCE SPEED**
DEFAULT PATH FILE FOR TAXI ROUTE	

A default path file for T-NASA taxi guidance on the HUD following shutdown of the ROTO module was associated with each exit. Normally this path file was not used. It was included to provide a default taxi route, if the aircraft cleared the runway and slowed to taxi speed, before the ATC taxi routing instructions were received. The default taxi path could extend to the terminal ramp or provide only a partial path, as desired.

3.6 DRAWING OF EXIT PATH

The requirement to seamlessly transition between HUD displays for ROTO operations and displays for taxi operations meant that ROTO on-ground symbology was constrained to have the T-NASA "look and feel" [3]. The two display codes also shared a common airport database. The T-NASA taxi HUD system was designed to depict a route the aircraft was to follow from a starting location to an intended destination. The route was developed from a database of runway and taxiway centerlines, and used edge markers known as "plops" to indicate the edges of a safe taxi path. The actual path associated with each possible taxi route was encoded in a file and given a unique alphanumeric name. When the taxi route was received from ground controllers, a lookup routine matched the path with a specific path file which was then used to construct the taxi route and the HUD guidance. If the taxi route was changed, the current route was deleted from memory and a new route was constructed from the appropriate path file.

The on-ground portions of the ROTO symbology were drawn using the T-NASA principals. However, the ROTO system created and loaded up to ten paths into memory. For each ROTO exit, a unique path was constructed. It began at the runway threshold and ended where the exit taxiway forked onto another taxiway or a few hundred feet off the runway, whichever condition occurred first. The runway was treated as a special path that went straight from the threshold end to the departure end of a runway. Whenever the ROTO software rendered the symbolic runway with exit taxiway, it used two routes. The route associated with the selected exit was drawn first. This provided edge markers and centerlines from the runway threshold to the end of the exit path. The special runway path was then used to construct the symbolic runway beyond the exit. The ROTO software contained logic to eliminate individual edge markers that obstructed the runway or exit and to prevent overlapping of symbology associated with the two routes. The result was an unobstructed symbolic runway from the threshold to the departure end with an opening at the location of the selected exit and a partial taxi path which led from the exit taxiway opening to a position a few hundred feet from the runway edge.

The ROTO database was key to the rendering of the ROTO symbolic runway. The number of runways and the information for up to nine ROTO exits each were encoded in the ROTO airport database (see section 3.5). For each runway and each exit, the numeric part of a unique alphanumeric path file name was encoded in the database. Path numbers associated with runways incremented in steps of ten, (i.e. 300, 310, etc.). Path numbers associated with exits were assigned sequentially, beginning with the runway path number, in the order the exit would be encountered by an arriving aircraft. For example, runway 26R was path file 300 and its ROTO exits were 301 through 308. When the ROTO system was initialized or restarted, the ILS frequency provided a database search key to the intended runway. The paths associated with the intended runway and its possible exits were loaded and assigned sequential numbers, beginning with zero. This provided a one-to-one correspondence between the exit number painted next to the ROTO exit selection knob on the PID and the order in which the routes resided in memory. To render the symbolic runway, the ROTO software used the ROTO knob position as an index to the exit taxiway path, and route number zero to draw the remainder of the runway. A taxiway at departure end of the runway was always the last possible ROTO exit. If the ROTO knob was rotated to an exit number greater than the number of ROTO exits available, the software defaulted to the first ROTO exit.

Runway and taxiway width information, encoded in the ROTO airport database, was used to automatically adjust the offset of edge markers from the centerline. This process was extremely critical to achieving the correct perspective view on the symbolic runway while on final approach. The illusions associated with runways that appear too wide or too narrow are well documented in Aeronautical Information Manual [11] and many other sources. By default, the offset of the edge markers was set to half the runway width. The offset was set to the encoded taxiway width only when the ROTO software detected an exit, and started drawing a taxi path that deviated from the runway centerline. Logic in the ROTO software detected and suppressed any plops that would have appeared to be on the runway surface.

The ROTO software created an OpenGL display list containing all of the graphics instructions necessary to render the ROTO symbolic runway and partial taxi path the first time they were drawn. Thereafter, the list was used to render the symbology, and the computations involved in locating and testing individual edge markers were avoided. This significantly increased graphics performance. If the runway or exit changed, the display list was destroyed and a new list was created on the first new drawing cycle.

3.7 EYE POINT ALTITUDE

The height of the pilot eye point above the airport surface must be determined in order to render conformal, on-ground symbology (e.g. the symbolic runway). Since the Atlanta airport database contained no elevation data and was based on a “flat Earth” model, special procedures were required to determine the eye point elevation for each phase of flight. All HUD ground-based display elements were rendered in a horizontal plane that was viewed, in a perspective mode, from a vertically displaced view point. The appropriate vertical offset was a dynamic quantity that changed whenever the aircraft was in motion.

For taxi operations, changes due to bumps on the airport surface were generally small and a constant vertical offset could be employed. The ROTO software assumed a pilot eye point 14.66 ft. above the airport surface whenever a squat switch indicated the aircraft nose wheel was on the ground. This produced ground-based display elements that were locally conformal with the true airport surface. Since the ground-based symbology was rendered in a horizontal plane, the apparent elevation of distant display elements was in error by the elevation difference between the local area and the distant airport surface.

For airborne operations, the vertical offset was computed for each frame of graphics from onboard sensor information in conjunction with data from the ROTO database. Special procedures were required to meet the accuracy requirements necessary for the ground-based display elements as the aircraft flew over uneven terrain. During the landing approach, the pertinent elevation was the height of the pilot eye point above the runway touchdown zone and not the distant to the terrain directly below the aircraft. Just prior to touchdown it was desirable to use the radar altimeter as the primary altitude data source, since it directly measured the height of the main landing gear above the airport surface.

At radar altitudes above 100 ft, the ROTO software vertically positioned the pilot eye point according to DGPS or barometric altitude data received though the flight management computers. The elevation of the runway touchdown zone was read from the ROTO database and subtracted from the aircraft altitude to calculate the view point vertical offset. Provisions were made to correct these calculations for the location of the DGPS navigation reference point and the effective location of the aircraft static port but these corrections proved unnecessary in practice. The graphics software always considered the aircraft yaw, pitch, roll, and the displacement of the pilot eye point from the DGPS ownship position when establishing the pilot eye point. The barometric altitude data were only used if the DGPS status flag indicated the DGPS data were invalid.

At a radar altitude of 100 ft, the ROTO software transitioned to the radar altimeter as the primary source of elevation data. This was desirable since the radar altimeter directly measured the distance to the ground and provided the most accurate elevation data available. An algorithm which compensated for the location of the radar altimeter receiver and the pitch of the aircraft was required to compute the pilot eye point from the radar measurement. NASA and Lockheed-Martin engineers developed the following formula from Boeing radar performance specifications and operational experience aboard the research aircraft:

$$h_{eye} = h_{radar} - 41.8 \cdot \sin\left(pitch_{aircraft}\right) + 17.64$$

where (h) was the elevation measured in feet and the ($pitch$) was read from the flight management system.

The transition from the DGPS or barometric altitude to the radar altitude occurred over a 2 second time interval during which the data were blended according to the following algorithm:

$$h_{eye\ transition} = c1 \cdot \left(h_{eye\ radar}\right) + \left(1 - c1\right) \cdot h_{DGPS}$$

$$c1 = \left(t / 2000\right) \quad until \quad c1 = 1$$

where (t) was the time since the start of the transition in milliseconds. The transition was complete when the dimensionless time ratio ($c1$) reached a value of one. A reverse procedure was required on takeoff to transition the pilot eye point from a fixed height above the ground to an elevation based on sensor information. After the nose wheel lifted off, the radar and DGPS altitudes were blended based on the indicated radar altitude above the surface. During the transition ($c1$) ranged from one at the surface to zero at 100 ft. This procedure ensured a positive value for the eye point location even if the lift off point was below the ends of the runway. That situation was encountered at the Atlanta airport.

3.8 ROTO SYSTEM OPERATIONS

The IDS software and its ROTO components were designed as part of a research system intended to demonstrate the readiness of technologies for commercial development. This research system was focused on technologies and guidance formats to support ground taxi and ROTO operations in low visibility conditions. It was not intended to be a general-purpose system supporting all phases of flight operations. Nevertheless, the Atlanta flight demonstrations were designed to present a “full mission” flight scenario. So it was required that the IDS possess capabilities for takeoff, cruise flight, landing and taxi operations (see section 2.0).

A general requirement of HUD-based guidance systems is that they actively manage the display of information by adding or removing screen elements according to the current flight operation. Display elements must be removed when no longer needed, and new elements must be added as required. The underlying motives are economical use of the display space and reduction of the amount of data pilots must process. The ROTO software was subject to this requirement, plus an additional constraint to seamlessly transition the HUD symbology to the T-NASA format as the aircraft reached taxi speeds. These requirements were addressed using a combination of pilot and automatic control of the HUD symbology. An automatic process controlled individual display elements, after a primary display was established. Both aircraft status information and results from the ROTO computational software were employed in the automatic process. The status of some aircraft discretes, (e.g. the main gear and nose gear squat switches) was used directly to control some screen elements. In general, the computed results were used to manage a group of screen elements and affect larger scale transitions of the symbology.

3.8.1 Initializing the ROTO system

Prior to execution, the "Buttonfly" Silicon Graphics menu utilities were used to establish the overall configuration of the IDS. The type of display supported (HUD or HDD) and input data source were among the user selectable options. At run-time the IDS software was self-initializing, except for a single command line argument that identified the airport. The three-letter airport designation was used for this purpose. From the three-letter code, the IDS automatically constructed the file names of all of the airport information files and the required databases. The IDS software driving the HUD was normally pre-configured for a start on the ground and in the taxi guidance mode. The IDS taxi monitor software passed control of the HUD display to the ROTO software only when the aircraft reached the departure runway. If the IDS was started while the aircraft was airborne, the taxi monitor passed control to the ROTO software immediately.

The primary factor for determining the appropriate HUD symbology was the current phase of flight. In flight, the pilot controlled the primary HUD display mode through the current ILS frequency, the position of the ROTO EXIT knob, and activation of the aircraft's go-around mode. On the ground, the ROTO module could only be initialized in the takeoff mode. The software maintained an internal operations code indicating the current flight mode that also enabled switching between primary HUD display formats. Figure 14 presents a functional schematic of the ROTO control procedures and the overall process of creating the guidance symbology.

3.8.2 Approach and Landing

A typical ROTO approach and landing was initiated when the flight crew tuned the captain's navigation receiver to an Atlanta ILS approach frequency causing a restart sequence which loaded information on the intended runway and exit taxiway and reinitialized the ROTO software. The symbolic runway and ROTO information box were added to the in-air symbology used during cruise flight. When capture of the ILS signal occurred, the localizer and glide-slope deviation scales were also added. At main gear touchdown, the deviation scales, airspeed indication, flight director bars, and the roll scale were removed from the HUD. The pitch ladder was retained to provide a visual aid during de-rotation. Rendering of the ROTO ground-based symbology commenced at main gear touchdown. The trend vector was shown to indicate the current ground track of the aircraft and runway remaining signs were added to the symbolic runway. At nose gear touchdown, the pitch ladder and heading scale were removed and drawing of the ROTO deceleration guidance commenced. The speed tape, acceleration carrot, and max speed arrow were sized and positioned according to continuously updated values and parameters read from the ROTO database. Drawing of the football and the goal line also commenced. For better visibility and a smoother transition to the T-NASA taxi format, the numeric ground speed indicator was relocated to the upper center portion of the HUD. Computation and display of the predicted exit speed also commenced. The football, goal line, trend vector, and runway remaining signs "faded in" from black into full view over a 1.0 second interval. This was accomplished by simply ramping the intensity of these objects from zero to full intensity.

The process of transitioning to the T-NASA taxi guidance began if the stopping criteria were satisfied and computations showed the aircraft to be three seconds from the start of the exit turn. The football, goal line, and remaining runway signs were removed and the "TURN" guidance cue was given. The "TURN" cue flashed for 1.5 seconds, then remained solid for an additional 5.0 seconds before being automatically removed. The speed tape was not shown after the aircraft passed the start of the turn onto the exit taxiway. Two factors determined the time and location of the transition to taxi HUD guidance. First, the IDS taxi monitoring software had to report that the aircraft was no longer on a runway. Second, the ground speed had to be below a pre-selected transition speed. If the aircraft exited onto the intended exit, the ROTO transition speed assigned to that particular exit was used. If the aircraft turned onto a non-selected taxiway, a default transition speed of 20 knots was used. When both conditions were satisfied, the ROTO module set a flag which terminated its execution and triggered the IDS software to commence the taxi HUD guidance. This two-step process eliminated a potential loss of HUD guidance while the aircraft was traveling at a considerable ground speed, on a high speed exit. If the stopping criteria were not satisfied, the "TURN" cue was suppressed, but the current exit symbology remained. Selection of a new exit farther along the runway re-activated the ROTO deceleration symbology with guidance to the new exit. The football and goal line were always suppressed if the aircraft ground speed was below the nominal exit speed.

3.8.3 Takeoff and Go-Around Modes

When the IDS system initialized the ROTO software on the ground and passed it the name of an Atlanta runway, the takeoff mode was engaged. The symbology used for takeoffs was developed as a subset of the ROTO approach HUD. The full flight HUD guidance was drawn, regardless of whether or not the aircraft was on the ground. The symbolic runway consisted of edge markers and an electronic centerline for just the runway. Runway remaining signs and the trend vector were drawn while the aircraft was on the surface. They were automatically removed when the aircraft main gear left the ground. No exit path was shown, regardless of the ROTO EXIT knob position, and the ROTO information box was suppressed. The ROTO deceleration guidance, football, and goal line were also not drawn. The takeoff mode was terminated when the pilot initiated an airborne restart of the system by selecting a new ILS frequency or placing the receiver in the "PARKED" status.

Aborted takeoffs triggered a restart of the normal ROTO deceleration mode with guidance to an exit selected according to the ROTO EXIT knob position. An aborted takeoff was assumed if the aircraft ground speed fell below 40 knots after having been above 40 knots during the takeoff run. The internal operations code was changed to the normal ROTO deceleration mode which resulted in the removal of the flight symbology and the reintroduction of the ROTO deceleration guidance. Normal procedures based on the status of the aircraft squat switches were followed to transition the HUD display from the figure 7 to the figure 4 format. Deceleration computations were based on the current aircraft position and speed at the time of the abort. The Atlanta version of the IDS required that the captain's ILS receiver was tuned to the departure runway ILS frequency for the abort sequence to function properly.

The ROTO software automatically switched to a go-around presentation whenever the flight management system was placed in a go-around mode. The display format for this situation is shown in figure 8. The display featured the full flight HUD guidance symbology with a symbolic representation of the runway edges and centerline. To "de-clutter" the HUD and "lock out" the plan view display, no exit information was presented and input from the PID was ignored. The ROTO software remained in a go-around mode until the flight management system was returned to a normal status. A flag from the system was continuously monitored to start and stop the go-around mode. On cessation of the go-around mode, cruise flight symbology was generated.

4.0 HUD ALIGNMENT

The use of symbology intended to be conformal with physical locations on the airport surface placed exacting requirements on the alignment of the HUD graphics and the determination of the aircraft position. Indeed, pilot confidence in, and acceptance of, both LVLASO cockpit displays was directly related to the conformality issue. Neither display would have been as successful without the 3 ft positional accuracy provided by the DGPS system. This is particularly true of the ROTO HUD displays, where even minor positional errors could produce instantly recognizable discrepancies. For example, pilots were quick to point out any misalignments between the symbolic and actual runway while still several miles from touchdown. Different procedures were required to calibrate the HUD graphics software for airborne and simulator based operations. They are discussed below.

4.1 AIRCRAFT

On board NASA's B-757 aircraft, the software and hardware driving the HUD were calibrated as a complete system. This was required since trim potentiometers in the hardware driving the HUD projection unit were adjustable in addition to the IDS software creating the HUD graphics. The first step involved ensuring that the HUD hardware was aligned with the fuselage of the aircraft. This step was completed by the HUD manufacturer, Flight Dynamics, Inc., under a separate contract with LaRC. The procedure involved leveling the aircraft, placing an alignment target directly in front of the pilot eye point, and adjusting the hardware mounts. The initial installation ensured that the HUD projector, pilot eye point, and the center of the HUD combiner glass were all located in a vertical plane that was parallel to aircraft centerline. The second step involved calibrating the HUD potentiometers and the IDS graphics software. An initial calibration was achieved using an alignment target inscribed with fiduciary marks for known displacements and circles mounted on a moveable stand. Field of view parameters in the IDS software and trim potentiometers in the HUD interface unit were adjusted until a test pattern projected on the HUD overlaid the pattern on the target board. The board was positioned at several measured distances from the pilot eye point to ensure that correct viewing angles had been established. This alignment was refined by positioning the B-757 pilot eye point over a carefully surveyed point on an LaRC taxiway and using the IDS databases for the Langley Air Force Base. A series of surveyed points was then used to verify the correct calibration of software viewing parameters and correct positioning of the ground-based symbology. Surveyed points close to the aircraft, other points more than one thousand feet from the aircraft, and the local horizon were all used. Pitch, roll, and heading information from the flight management system was also used during the taxiway alignment process since the B-757 had an innate pitch, and possibly a non-zero roll while resting on its own wheels. These procedures resulted in the following software field-of-view parameters for the HUD graphics: a width of 27.6 degrees and height of 23.0 degrees producing an aspect ratio of 1.2. The line from the pilot eye point through the center of the HUD combining glass was found to be biased 3.95 degrees downward from a horizontal plane.

The calibration procedure produced a "flat Earth" alignment which 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. This well-known phenomenon was expected, since the flight symbology was referenced to a level flight attitude.

4.2 SIMULATION FACILITIES

A different software calibration was required in the LaRC Visual Motion Simulation (VMS) facility, where the actual HUD hardware was not available. A "virtual HUD" was created by mixing the HUD symbology with the video signal comprising the out-the-window scene. The signals were electronically combined using commercial video mixing hardware. This "virtual HUD" display proved quite serviceable. The calibration procedure involved finding IDS software graphics parameters that would produce a match between the HUD symbology and the apparent location of objects depicted in the out-the-window scene. Objects such as the runway, taxiways, and the horizon were used to establish the software parameters. Alignment was verified by checking that the video match was retained as the simulated aircraft changed headings and attitudes. Due to an electronic artifact of the video mixing process, empirical offsets in yaw and pitch were necessary. These offsets 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. The following software field-of-view parameters were determined for the electronically mixed HUD graphics: a width of 53.725 degrees and height of 42.98 degrees in an aspect ratio of 1.25. A horizontal offset of 4.5 degrees and a vertical offset of 0.9 degrees were required due to an unavoidable misalignment of the HUD and the out-the-window video frames.

5.0 CODE VALIDATION

Pre-deployment flight testing of the LVLASO experimental systems was very limited due to a tight schedule in which activities for the project competed with preparing the B-757 aircraft's baseline research systems for flight. Aircraft simulation facilities, off-line playback of recorded data, and simulated control inputs using a mouse and keyboard were used extensively to develop individual components of the IDS software. Modules for ROTO and taxi operations were developed independently within a general overall framework and then merged to produce the flight version of the IDS software. The full IDS flight software could not be tested in simulation facilities since no single facility featured all of the aircraft data, traffic data, text messages, and runway status information the IDS was designed to process and display. The sections below address the development and testing of the ROTO software. The steps involved in developing and testing other components of the IDS will be published elsewhere.

5.1 SIMULATION FACILITIES

The functional capabilities of the ROTO software were primarily developed in the VMS facility using the "virtual HUD" described above. Computers generating the simulation placed real time aircraft and traffic data into shared memory areas that replicated the data structures and data formats used on the B-757. This arrangement permitted development of the ROTO software in an environment that replicated the operating environment on the research aircraft. The software functioned in the same manner whether the shared memory data came from aircraft systems or from the simulation computers.

All anticipated phases of flight were simulated to test the ROTO software and develop the display formats used on the HUD. Repeated simulated landings on all Atlanta runways were performed to ensure that the information displayed on the virtual HUD was in agreement with heads-down cockpit instrumentation and that the ROTO deceleration guidance functioned correctly. The ROTO deceleration symbology was tested both quantitatively and also for acceptability of the look and feel. A rotary switch mounted in the VMS cockpit substituted for the ROTO EXIT selection knob on the aircraft PID. The switch enabled functional testing of the pilot exit selection process, the ROTO automatic exit selection mode, and the ROTO plan view display. Taxi out and takeoff operations were simulated to test the transition from the taxi HUD to the ROTO software takeoff display. Transitions from the ROTO deceleration guidance to the T-NASA taxi HUD format were extensively simulated to ensure the reliability and seamlessness of the transitions. Simulations were conducted to validate the HUD displays for cruise flight and the go-around mode. The use of the ILS frequency to engage or disengage the normal ROTO approach and landing mode was also validated in simulation.

The VMS facility simulated the dynamics and physical characteristics of a Boeing 737 aircraft. Therefore several software parameters required adjustment when the code was ported to the B-757. These included physical constants such as the offsets of the pilot eye point with respect to the DGPS navigation reference point, and parameters related to the performance of aircraft systems, such as the radar altimeter. When the B-757 aircraft was on the ground, the pilot eye point was located 65.45 ft. forward, 1.75 ft. to the left, and 3.56 ft. above the DGPS navigation reference position. This position was 30 ft farther forward than the simulated B737 location. The long displacement from the ownship position necessitated a vertical correction in the pilot eye point for the pitch of the aircraft. Formulas specified by NASA engineers were used to determine the appropriate correction when the radar altimeter, barometric altitude, or DGPS altitude was the primary data source.

Initial validation of the software with the SCRAMNet$^+$ communications ring and ARINC-429 data formats used on the B-757 was performed in the Langley Experimental Aircraft Systems Integration Laboratory (EASILY). The code was interfaced with the actual data processing hardware installed on the

B-757 and tested with control inputs from flight hardware. Full functionality of the PID was verified. In this stage the data blocks were put in their final form, and issues of sign convention, resolution, and range of parameters were resolved.

5.2 PLAYBACK OF RECORDED DATA

As part of the IDS development effort, software was written to record the entire contents of the shared memory areas at real-time rates. Software was available to record actual flight data obtained from the B-757 or the simulated data used in the VMS and EASILY laboratories. It proved invaluable for debugging and validating the ROTO software. By "playing back" the recorded file at an appropriate rate, the real-time performance of the software could be tested off-line. This technique greatly increased the value of simulation sessions and the limited flight test data. It was especially valuable for investigating unanticipated circumstances (e.g. drop out of the DGPS positional information) and issues pertaining to timing (e.g. transition from taxi guidance to takeoff mode) where some combination of inputs caused the code to fail. Off-line, debugging tools such as print commands and saving selected variables to a file for detailed analysis were also used extensively with the recorded real-time data.

5.3 FLIGHT TESTING

Some preliminary flight testing of the integrated IDS flight software and the ROTO guidance system was conducted at the Langley Air Force Base. Six flights in the local area were conducted to test the IDS, the new research hardware aboard the B-757, and the reconfigured airplane itself. Partial IDS databases of the Langley runways and taxiways were created for these tests. Both manual and auto landings followed by decelerations to a pre-selected exit were used to validate the ROTO deceleration guidance. The HUD displays for cruise flight and ILS approaches were validated against heads-down cockpit instrumentation, real-time comparisons with data from the flight management computers, and through post-flight analysis of recorded data. The flight tests were particularly useful for identifying real world conditions (e.g. drifting of the ownship position) that impacted the software mechanisms which transitioned between ROTO and taxi guidance formats. Both taxi and flight operations were conducted to verify the alignment of the HUD graphics. The HUD-based and HDD-based taxi guidance systems were tested by navigating Langley taxiways. The IDS control software [1] provided a mechanism to exercise some features of the code by direct interaction with shared memory areas. For example, text messages containing taxi route instructions could be simulated.

Full validation of the IDS software at the Langley Air Force Base was not possible because the ground-based infrastructure required to produce the traffic data, the runway status information, and the ATC text messages was not available.   The initial deployment of the LVLASO experimental system to Atlanta (July, 1997) provided the first opportunity to test the entire system. Fortunately, the careful pre-deployment tests performed in simulation facilities and aboard the B-757 resulted in robust and reliable IDS software.

6.0 LESSONS LEARNED

The IDS summary report [1] contains a general overview of the lessons learned during the development of the IDS software for the LVLASO project. The following section summarizes the findings explicitly related to the operation of the ROTO system and provides some additional observations concerning specific ROTO functions. No attempt is made to analyze the results of the Atlanta flight tests or present a comprehensive listing. Evaluation pilots were enthusiastic about the ROTO system and found it very easy to use. However, there were fundamental limitations associated with the experimental system. These generally concerned the database of the airport surface and the mechanisms to control software functions.

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 these route segments. The design was flexible and resulted in a compact database, but it also precluded the inclusion of some surface features. For example, three-dimensional features such as sloping runways or runways with an uneven surface could not be conformally depicted since the database contained no elevation data. During taxi operations, a pilot eye point at a fixed constant height above the local surface was assumed. This produced a locally conformal HUD display. However, 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. Since elevation data were not encoded, the pilot eye point above objects in the database could not determined directly. Supplemental information from the ROTO database was required to determine the mean sea level (MSL) altitude of the runway threshold, and the pilot eye point above the database was computed using onboard sensors. The use of a constant width for all route segments precluded the inclusion of surface features such as fillets between straight sections of pavement. This meant that edge markers were not placed at the actual edges of the pavement. That feature was intentional for taxi operations but it was undesirable for ROTO operations since it produced an inaccurate perspective of the turn onto an exit taxiway. The ROTO software compensated for this shortcoming by dynamically adjusting the runway and taxiway segment width parameters from information in the ROTO database.

Further limitations of the Atlanta airport database included overly complex procedures to create and access the database, the possibility of ambiguous positions along a route, and the lack of a mechanism to encode coordinates where centerlines diverged. The task of constructing the database was very labor intensive and involved redrawing the airport surface on top of airport surface CAD files. Construction of ROTO exit paths also required accessing both the raw airport drawing and the finished database. These labor intensive steps represented serious impediments to future commercial applications of the LVLASO technology. The Atlanta database classified route segments in such a way that, at some locations, it was not possible to tell whether the pavement under the aircraft was part of the runway or part of a taxiway. This made transitions of HUD symbology between ROTO and taxi operations more difficult to initiate. Both the ROTO system and the taxi system could have benefited from encoding coordinates for the locations where centerlines for route segments diverged. This would have simplified locating the intended exit taxiway and the dynamic creation of taxi routes.

Early in the development process, it was clear that the LVLASO technologies would enter commercial service as an all-weather system whose benefits extended well beyond low visibility conditions. The pilots who flew the LVLASO system and members of the ROTO development team suggested several improvements aimed at making the system more suited to all-weather continuous daily operations. The improvements ranged from simple matters, such as a requirement that the pilot directly control the HUD intensity, to more substantive issues. The Atlanta ROTO system used the ILS frequency as a search key to determine the intended runway. However, to support a variety of approach types and avoid errors associated with inadvertently tuned receivers on non-ILS approaches, future systems should receive the runway information directly from the flight management system. The Atlanta implementation used a non-intuitive mechanism (a frequency change) to switch between display modes. This was an artifact of the LVLASO development sequence; the flight qualified PID was designed and built before the decision to support multiple display formats on the HUD was made. A pilot-controlled mode switch to select the desired display would have provided a more positive control over the symbology and simplified the ROTO code. 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.

The ROTO software was extensively tested prior to the Atlanta deployment in simulation facilities at LaRC. Every feature of the code was operationally checked and even minor anomalies were pursed to a satisfactory resolution. When the code was ported to the B-757, very few logic errors surfaced. However, the mismatches between the B737 aircraft dynamics available in the simulation facilities and the B-757 research aircraft did result in some unforeseen problems. For example, the B-757 auto land system could require over eight seconds to de-rotate the nose gear. A time delay of this magnitude was not originally factored into the ROTO automatic exit selection logic and the algorithm required adjustment. Since future versions of the ROTO system will be increasingly intertwined with the aircraft flight management system and the 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 faithfully reflect “real world” phenomena such as measurement uncertainties or drifting of the GPS position. Therefore, it is vitally important to flight test future versions of the ROTO system as much as possible.

Feedback from the LaRC test pilots and the airline pilot evaluators of the ROTO system was immensely valuable during the development phase. It was actively sought by members of the ROTO team, and the information received was acted upon. Pilots objected vigorously to the idea of having any symbolic obstructions on the runway surface on operational grounds that it appeared to limit their options if the exit was missed or a go-around was required. Meeting this constraint required extra programming effort but clearly increased pilot satisfaction with the symbolic runway. Several other display elements including the flight director bars, aim point markers, runway remaining signs, and the ROTO box were developed either wholly or partially in response to pilot suggestions. Pilot comments reinforced the basic HUD tenet that the amount of information presented at any single moment be limited, and encouraged rigorous de-cluttering of the display. The area near the center of the HUD display had to be reserved for the most crucial information in order that the symbology not obscure the actual view through the windscreen.

It can not be overstated that HUD symbology intended to be conformal with the outside world must be conformal. Pilots frequently commented on the compelling nature of the HUD conformal symbology and the confidence developed in the system when the symbology matched perfectly with the real world. This places a tremendous burden on developers of HUD guidance systems that use conformal symbology, but it is also the key to the commercial acceptance of these technologies. When the reliability and aviation certification standards are met, these technologies will contribute significantly to safe, comfortable aircraft operations in all weather conditions.

ACKNOWLEDGMENTS

This work was performed under task order number ARTS-0-LDI04 for the NASA Langley Research Center. The authors wish to also thank Sharon O. Beskenis and David F. Green who, as fellow members of the Lockheed-Martin development team, provided assistance, cooperation and the sharing of ideas in all phases of the LVLASO project.

REFERENCES

1. Beskenis, Sharon O., Green, David F., Jr., Hyer, Paul V., Johnson Edward J., Jr.: Integrated Display System for Low Visibility Landing and Surface Operations. NASA/CR-1998-208446, July, 1998.

2. Jones, Denise R. and Young, Steven D.: Flight Demonstration of Integrated Airport Surface Automation Concepts. AIAA/IEEE 14th Digital Avionics Systems Conference. Nov. 5-9, 1995, Cambridge, MA.

3. Foyle, David C., Andre, Anthony D., McCann, Robert S., Wenzel, Elizabeth M., Begault, Durand R., Battiste, Vernol: Taxiway Navigation and Situation Awareness (T-NASA) System: Problem, Design Philosophy, and Description of an Integrated Display Suite for Low-Visibility Airport Surface Operations. 1996 World Aviation Congress. Oct. 21-24, 1996, Los Angeles, CA, Paper No. 965551.

4. Young, Steven D. and Jones, Denise R.: Flight Demonstration of Integrated Airport Surface Movement Technologies. NASA TM-1998-206283, January 1998.

5. 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.

6. Hueschen, Richard M., Hankins, Walter M., Barker, L. Keith: Description and Flight Test of a Rollout and Turnoff Head-up Display Guidance System. 17th Digital Avionics Systems Conference, Oct. 31-Nov. 6, 1998, Seattle, WA.

7. Pines, S. and Hueschen, R. M.: Guidance and Navigation for Automatic Landing, Rollout, and Turnoff Using MLS and Magnetic Cable Sensors, AIAA Guidance and Control Conference, Palo Alto, California, August 7-9, 1978.

8. Goldthorpe, S. H., Dangaran, R. D., Dwyer, J. P., McBee, L. S., Norman, R. M., Shannon, J. H., Summers, L. G.: Guidance and Control Design for High-Speed Rollout and Turnoff (ROTO), NASA/CR-201602, August, 1996.

9. 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.

10. Bray, Richard S. and Scott, Barry C.: A Head-up Display for Low Visibility Approach and Landing. AIAA 19th Aerospace Sciences Meeting, January 12-15, 1981, St. Louis, Missouri. Paper AIAA-81-1030.

11. Spanitz, Jackie, ed: Federal Aviation Regulations and Aeronautical Information Manual. 1997 edition, Aviation Supplies & Academics, Inc., 1997.

APPENDIX A – 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.

```
    #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*/
};
```

APPENDIX B - ROTO C Data Structure

This appendix contains the structure of the ROTO database for airport information, runways and exit taxiways.

```
/* roto_airport_data.h */

struct    airport_params {
          char      Name[40];           /*long name*/
          char      Code[6];            /*6-char identifier*/
          float     Lat;                /*latitude of airport*/
          float     Long;               /*longitude of airport*/
          float     AirportElev;        /*airport elevation*/
          short     Nrunways;           /*number of runways for this airport*/
          short     rIndx;              /*index for chosen runway*/
          short     eIndx;              /*index for chosen exit*/
          short     spareShort1;
          struct    runway_params       *thisRunway;        /*data for selected runway*/
     };
struct runway_params {
          char      Name[6];            /*runway identifier*/
          short     baseRoute;          /*base route for the runway*/
          short     Nexits;             /*number of exits for this runway*/
          float     Lat;                /*threshold*/
          float     Long;
          float     Elev;
          float     GS;                 /*glide slope in degees*/
          float     TCH;                /*threshold crossing height*/
          float     Freq;               /*ILS frequency for runway*/
          float     Length;             /*length (ft) of runway*/
          float     Slope;              /*slope of runway*/
          float     Width;              /*width of runway*/
          float     Heading;            /*magnetic compass heading */
          struct    exit_params         *thisExit;          /*params for chosen exit*/
     };


struct    exit_params {
          char      Name[6];            /*identifier for exit*/
          float     Width;              /*specific for exit*/
          float     Distance;           /*distance from threshold*/
          float     Angle;              /*angle of exit w.r.t. runway*/
          float     exitSpeed;          /*speed at which to make safe exit*/
          float     maxExitSpeed;       /*max speed to consider making an exit*/
          float     tranSpeed;          /*speed at which to transition display*/
          short     taxiRoute;          /*default taxi route for this exit*/
          short     SpareShort1;
      };

struct    airport_params Rairport;
struct    runway_params Rrunways[10];
struct    exit_params Rexits[10][10];
```

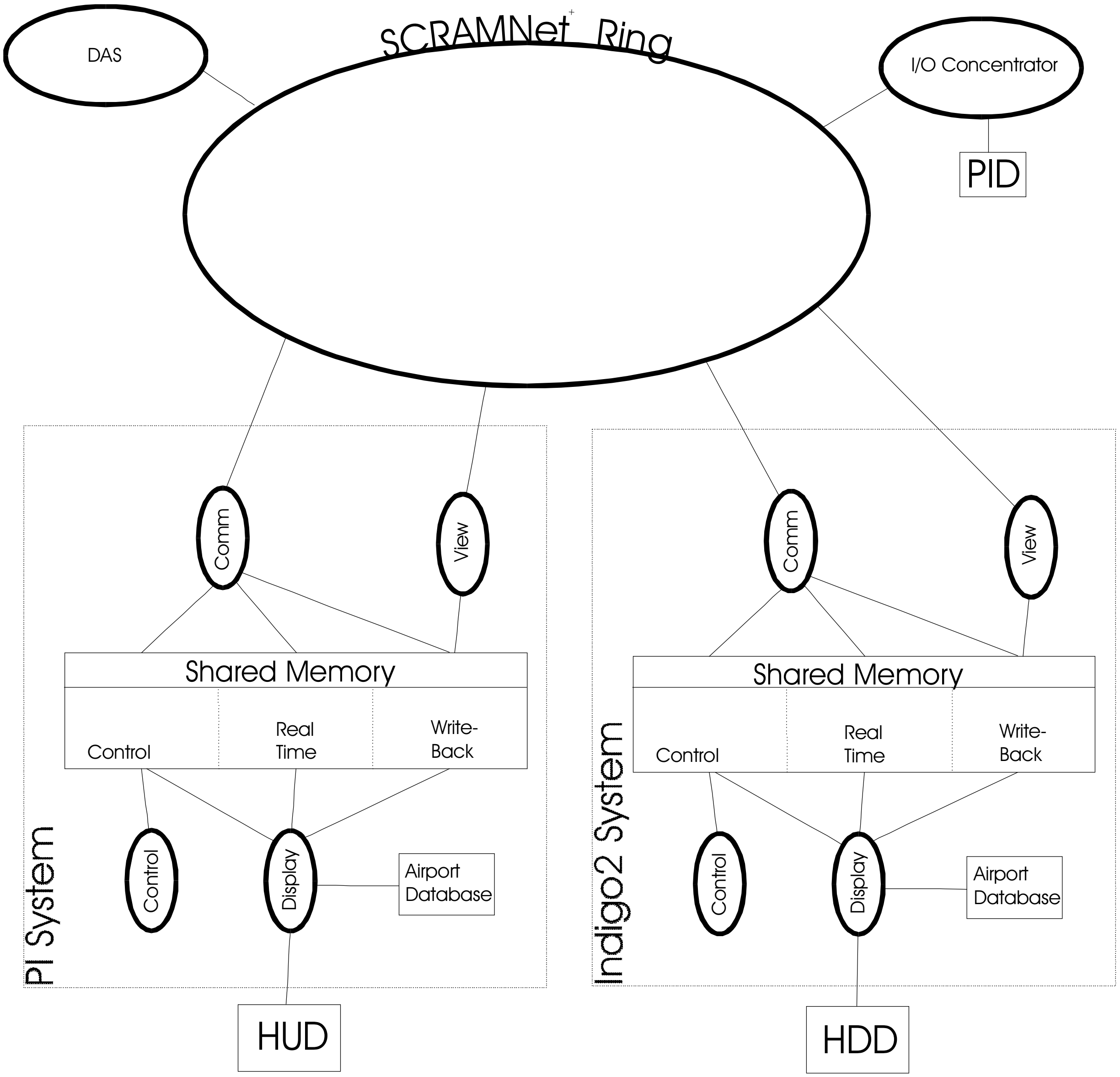


Figure 1. High level software architecture on aircraft

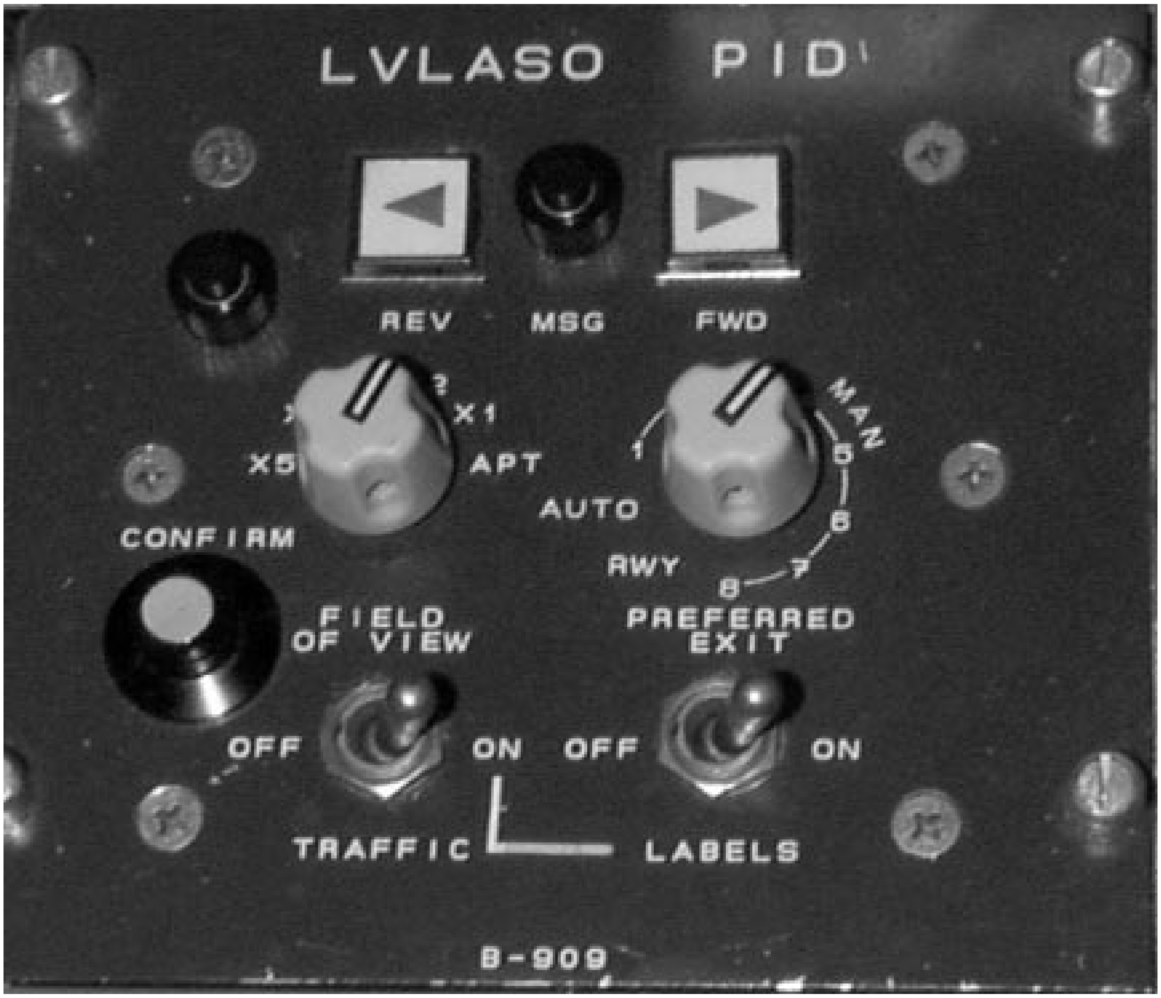


Figure 2. Pilot input device

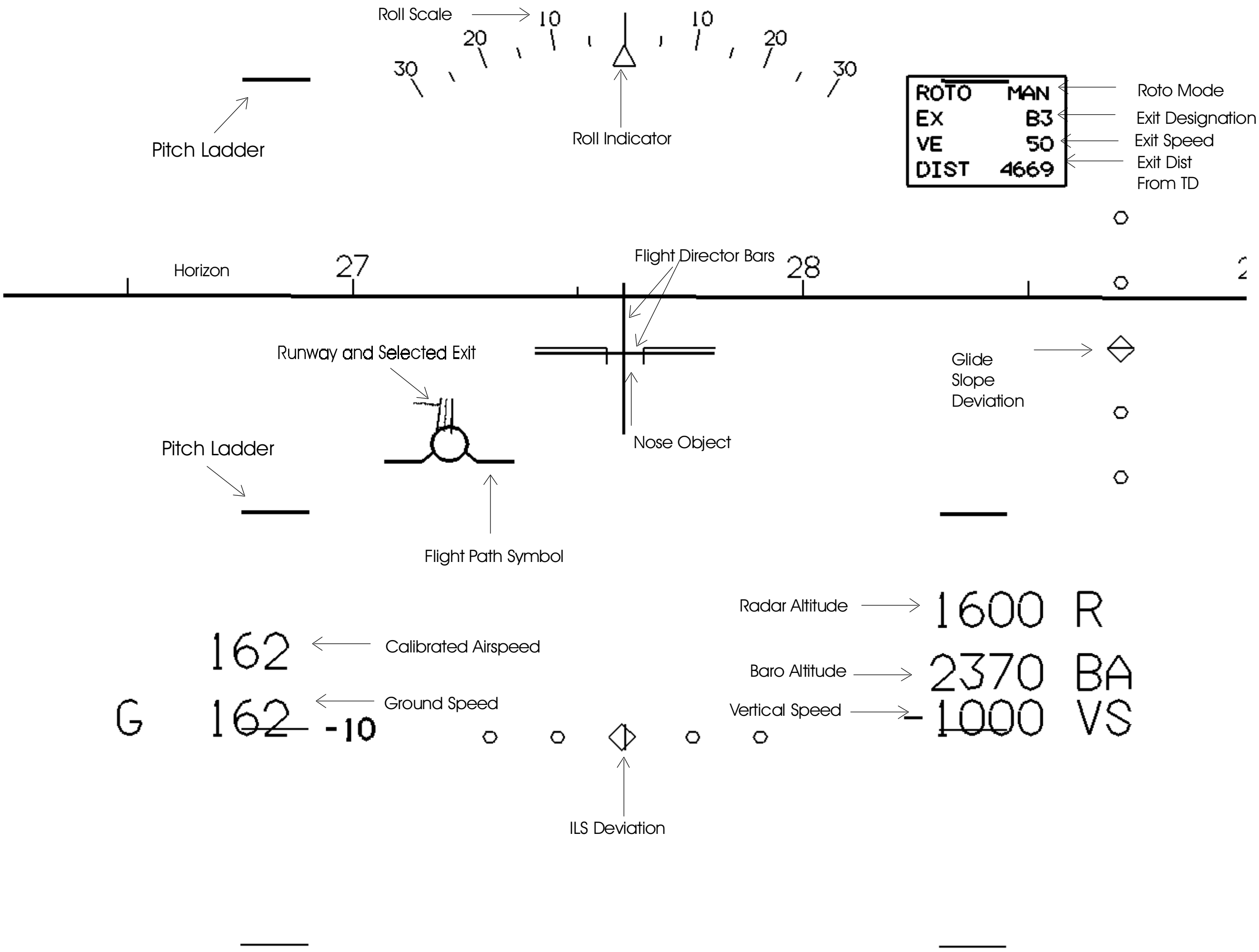


Figure 3. Example of approach HUD display format

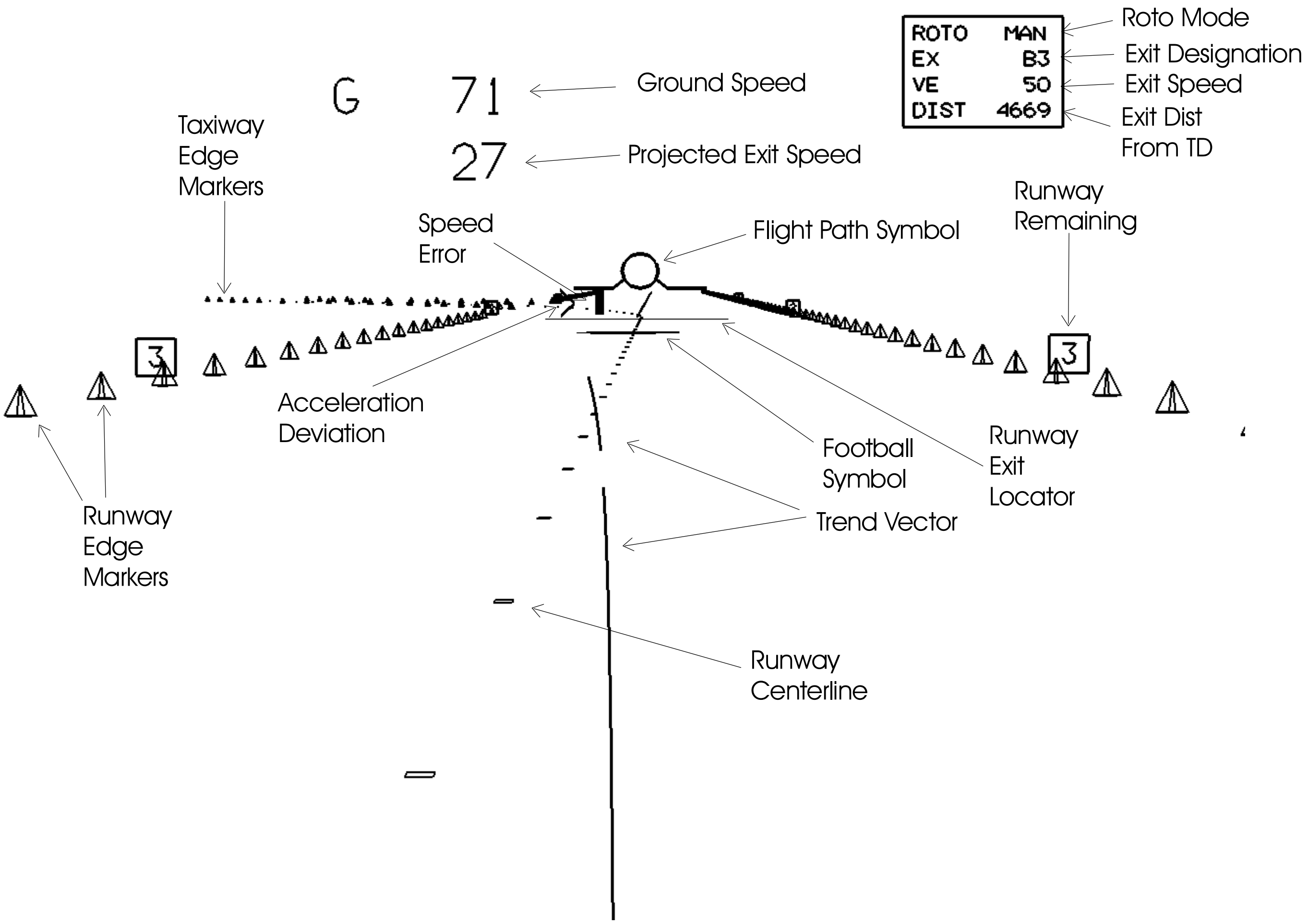


Figure 4. Example of deceleration guidance display format

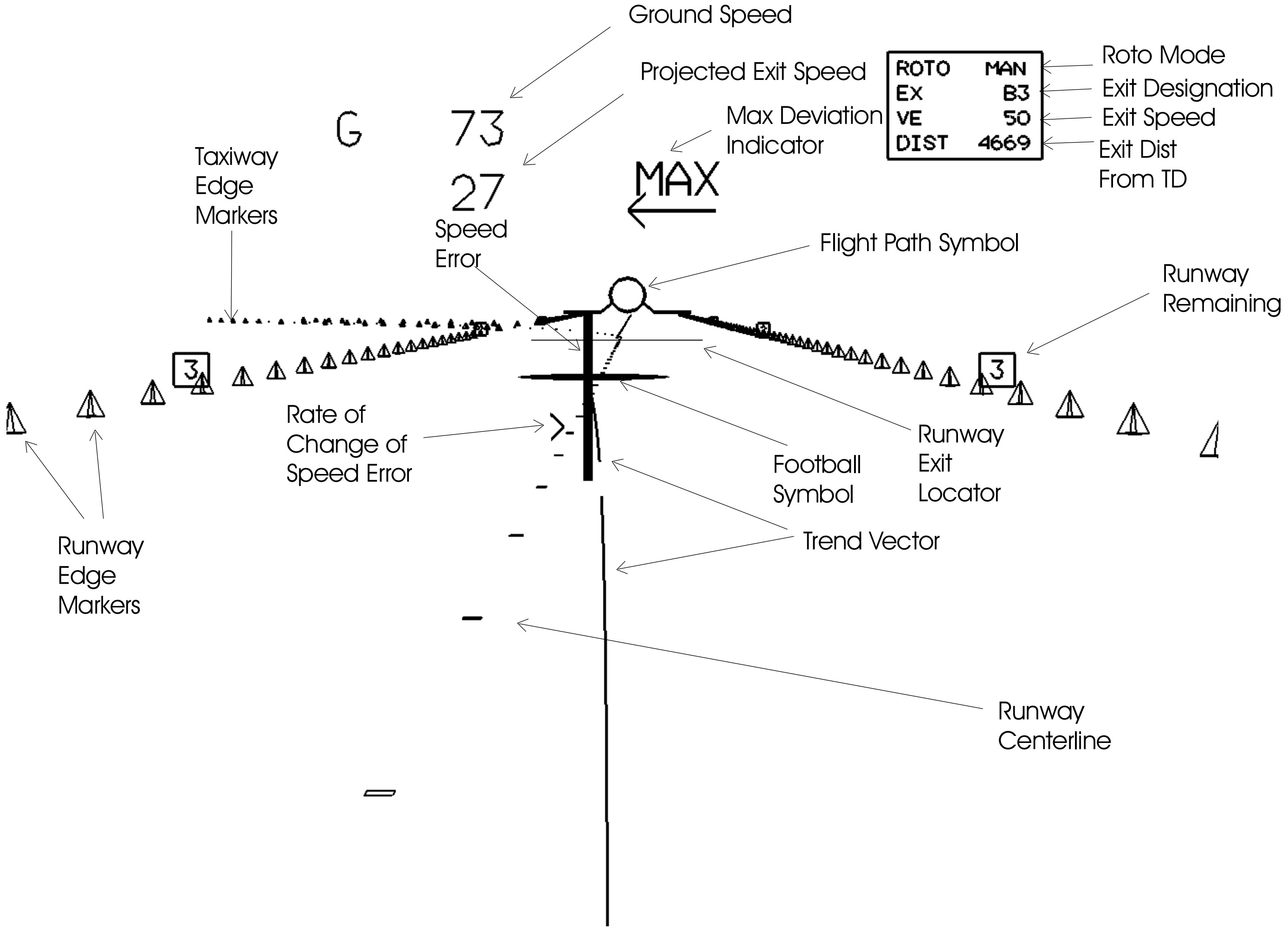


Figure 5. Example of alternate deceleration guidance display format

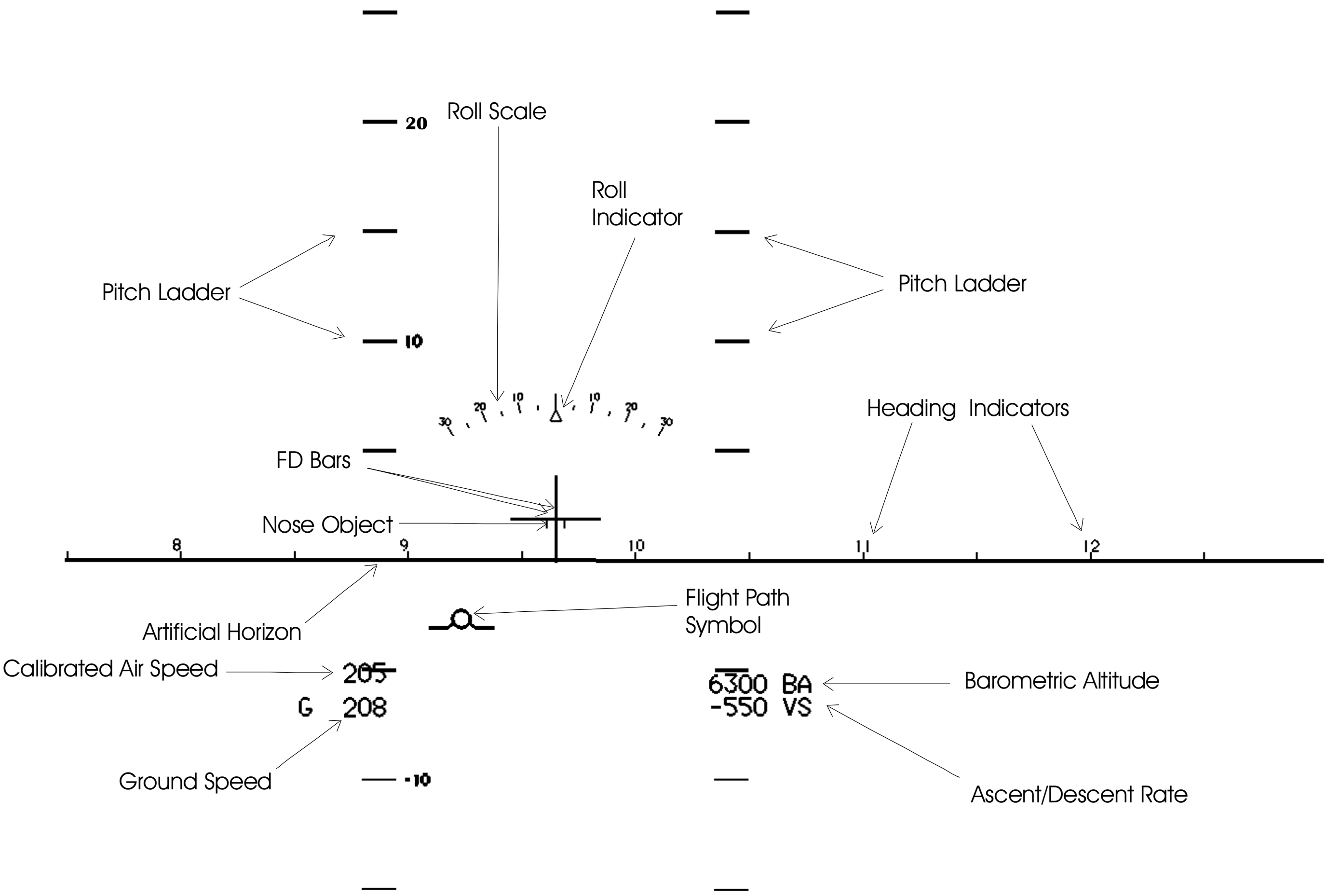


Figure 6. Example of cruise flight display format

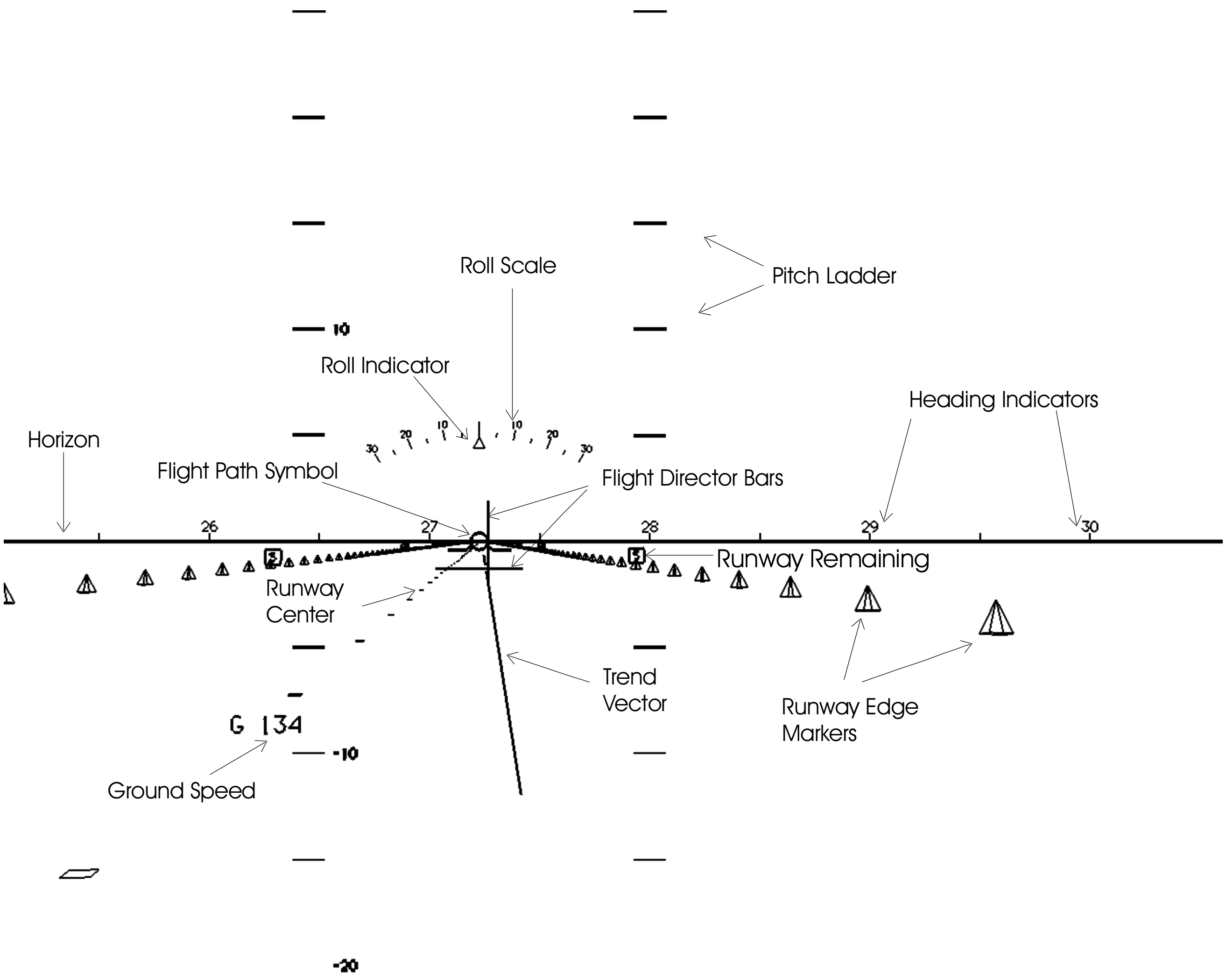


Figure 7. Example of take-off mode display format

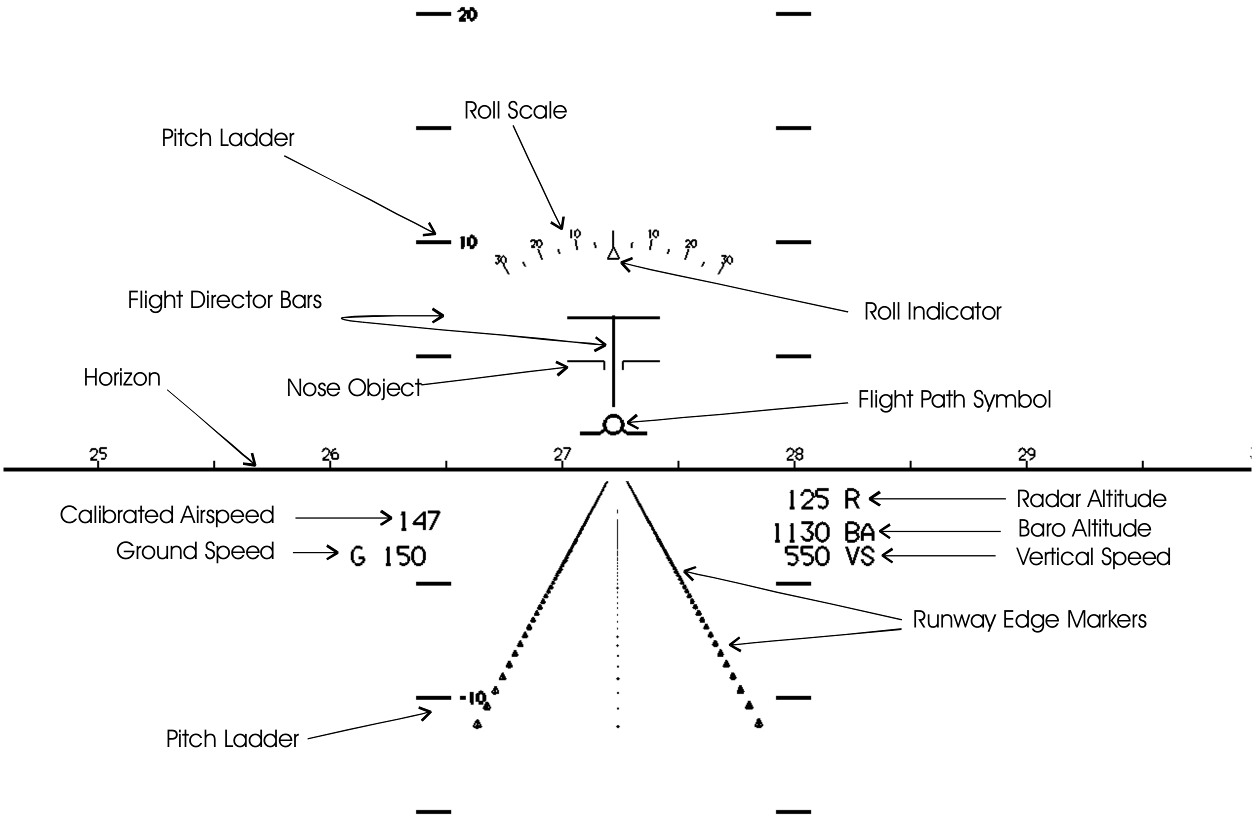


Figure 8. Example of go-around mode display format

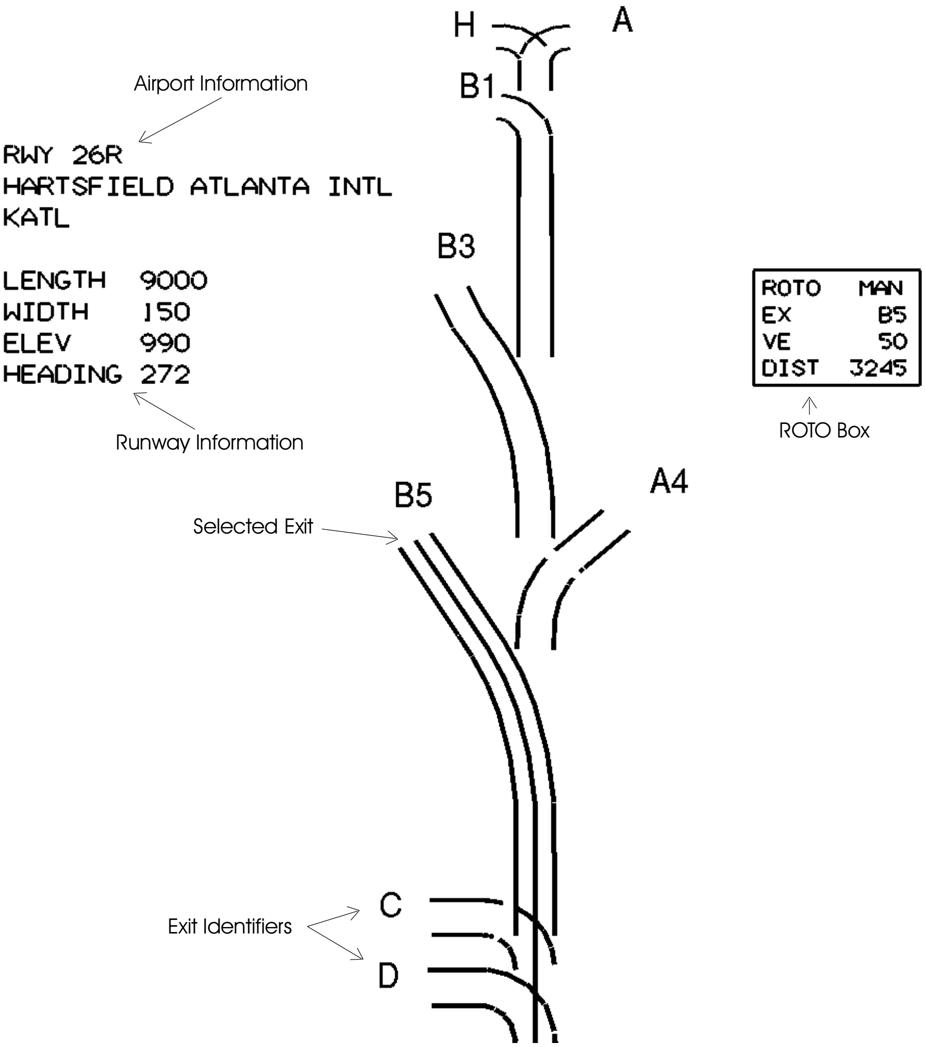


Figure 9. Example of ROTO plan view display

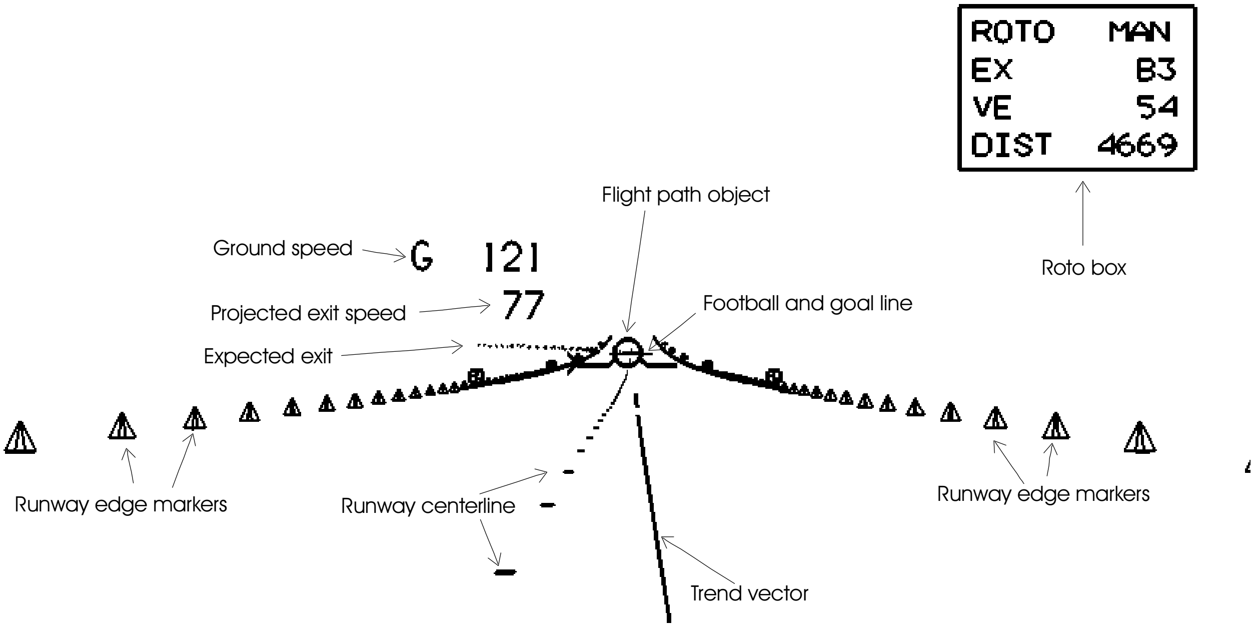


Figure 10. Example of curved runway display

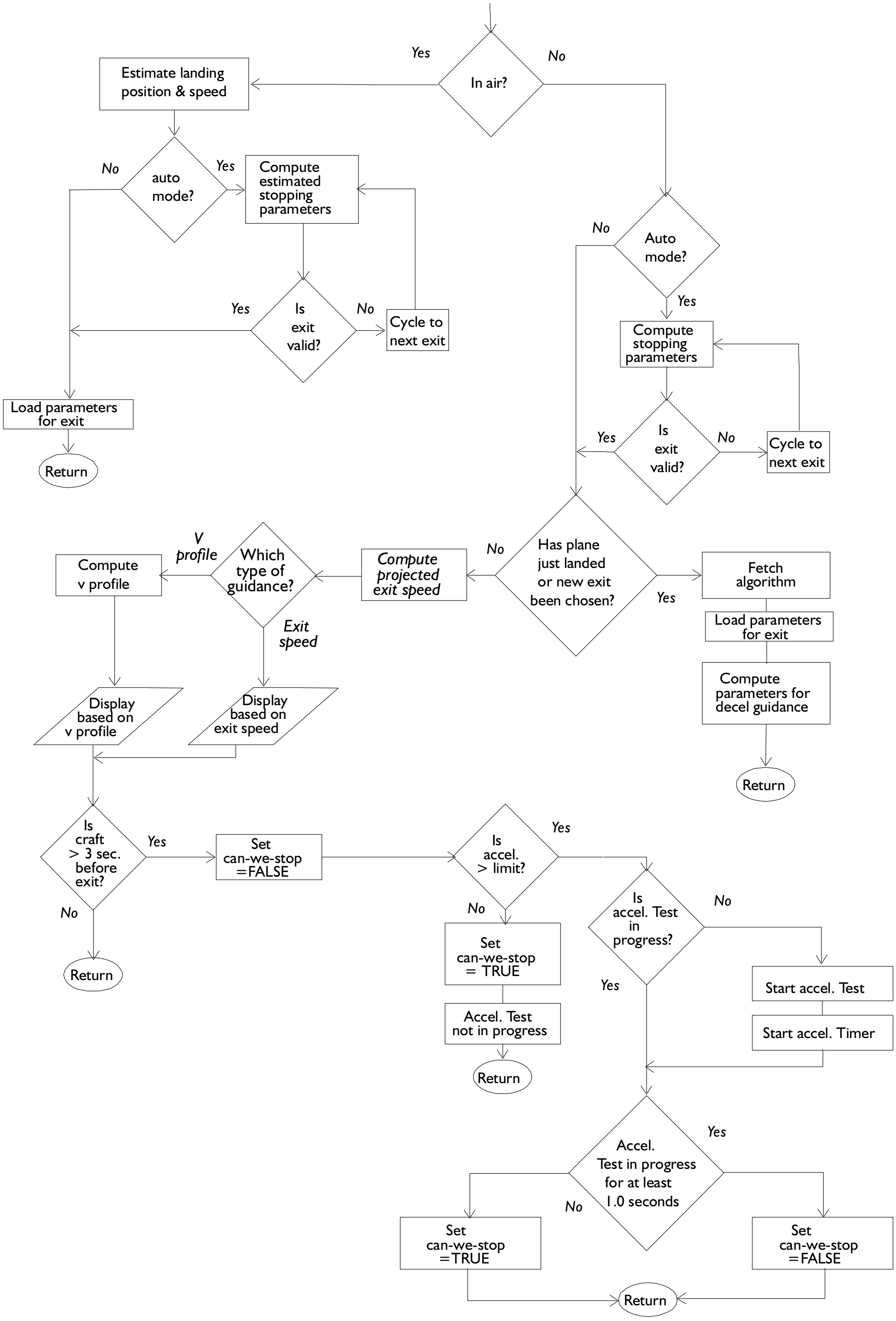
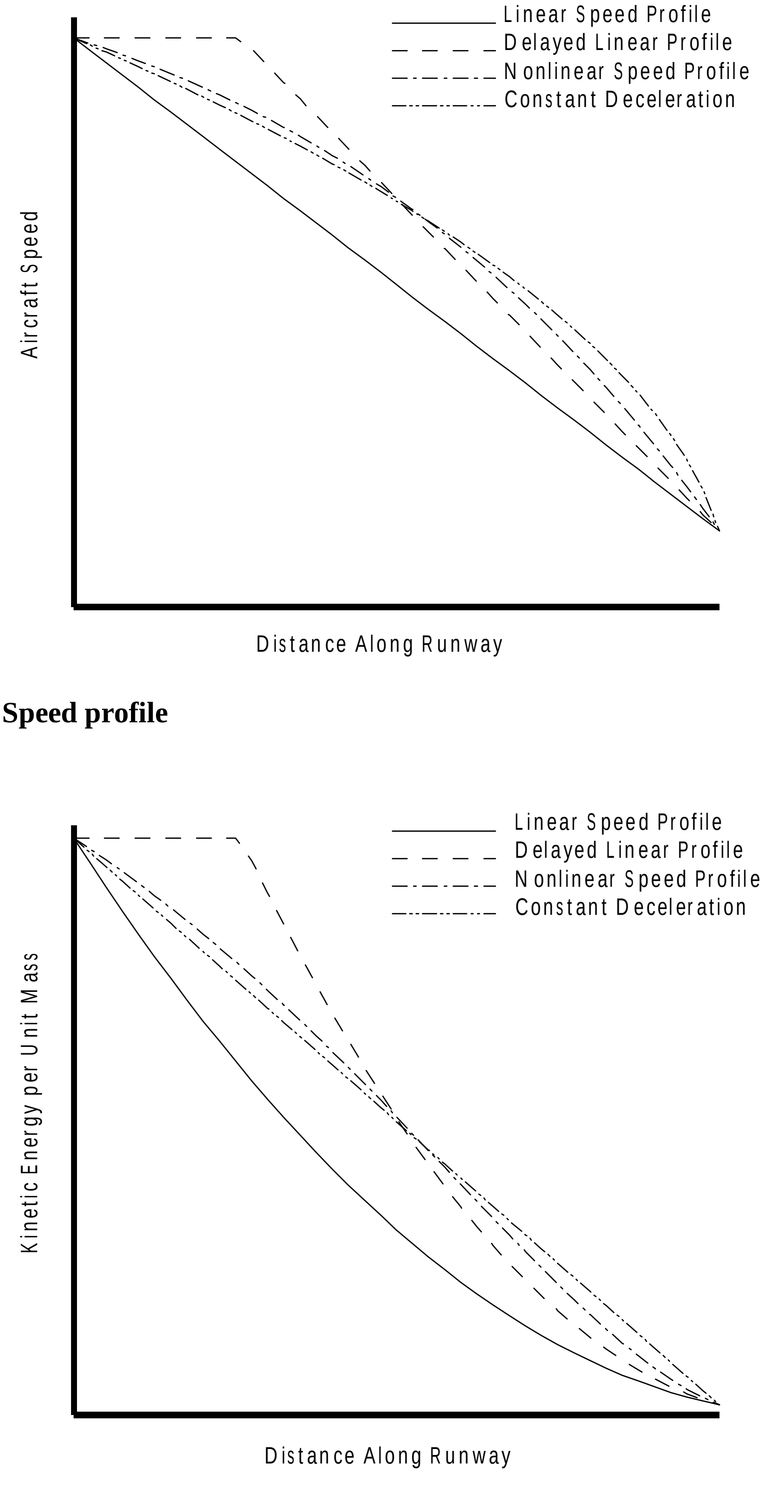


Figure 11. Logical flow diagram for deceleration algorithms



(a) Speed profile

(b) Kinetic energy profile

Figure 12. Comparison of deceleration profiles for a typical case

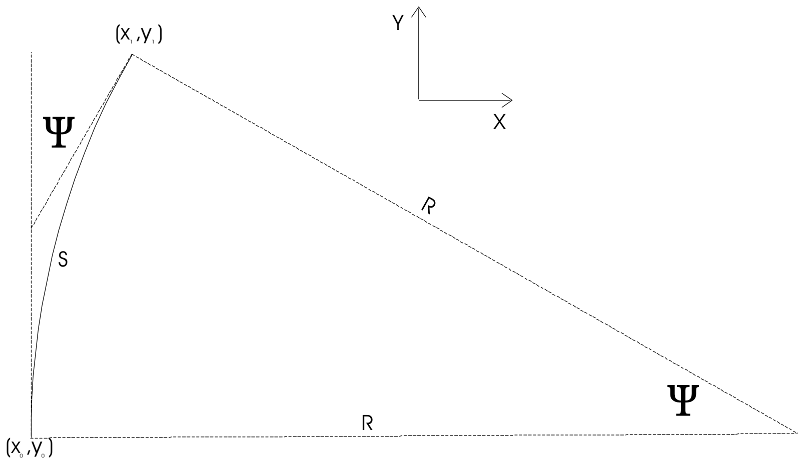


Figure 13. Coordinates for calculation of the trend vector

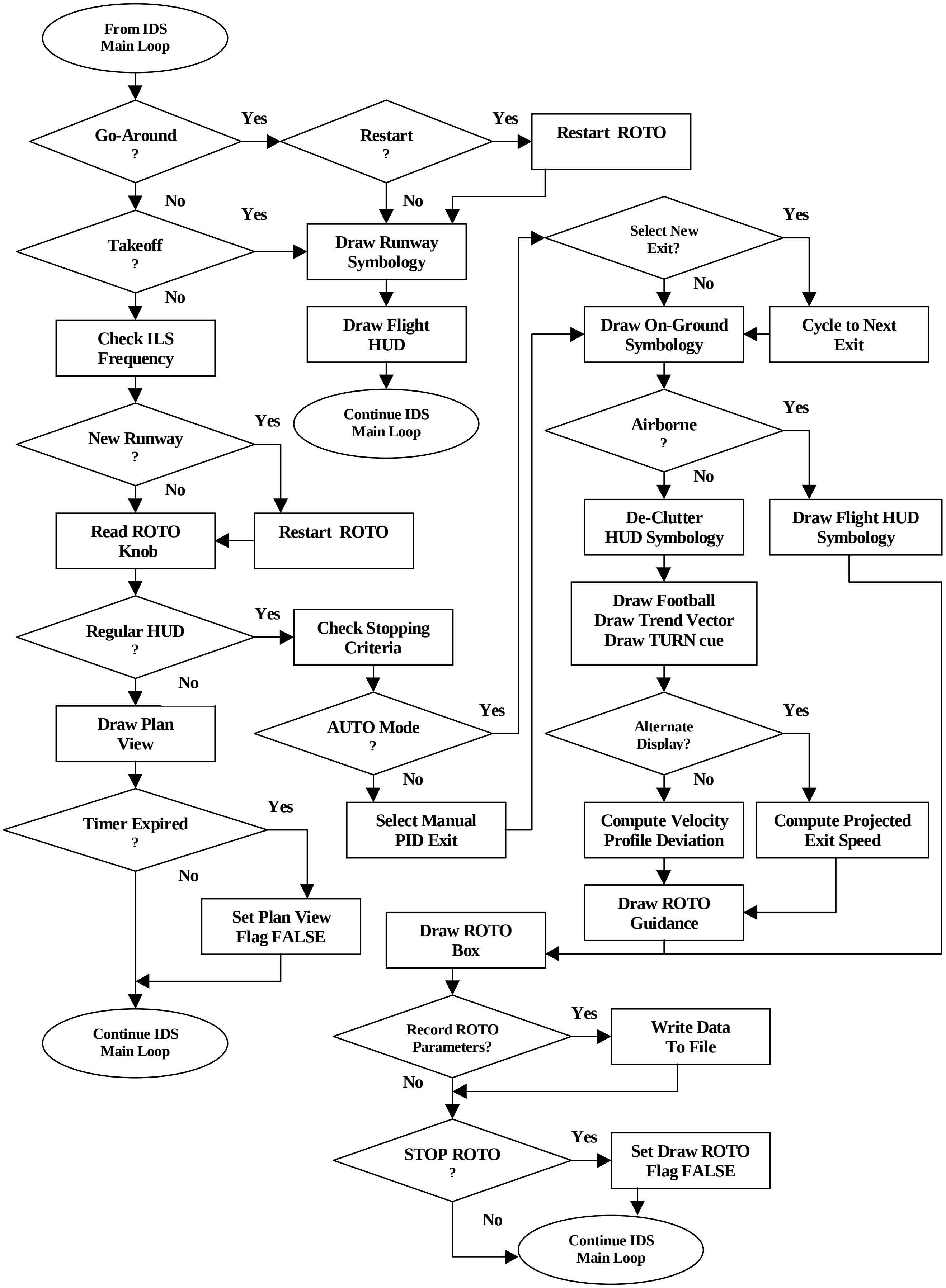


Figure 14. Logic flow for ROTO control procedures

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
November 1999

3. REPORT TYPE AND DATES COVERED
Contractor Report

4. TITLE AND SUBTITLE
Roll-Out and Turn-Off Display Software for Integrated Display System

5. FUNDING NUMBERS
NAS1-96014
538-04-13-02

6. AUTHOR(S)
Edward J. Johnson, Jr., and Paul V. Hyer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Lockheed Martin Engineering & Sciences Company
Langley Program Office
Langley Research Center, Mail Stop 371
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-2199

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/CR-1999-209731

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 describes the software products, system architectures and operational procedures developed by Lockheed-Martin in support of the Roll-Out and Turn-Off (ROTO) sub-element of the Low Visibility Landing and Surface Operations (LVLASO) program at the NASA Langley Research Center. The ROTO portion of this program focuses on developing technologies that aid pilots in the task of managing the deceleration of an aircraft to a pre-selected exit taxiway. This report focuses on software that produces a system of redundant deceleration cues for a pilot during the landing roll-out, and presents these cues on a head up display (HUD). The software also produces symbology for aircraft operational phases involving cruise flight, approach, takeoff, and go-around. The algorithms and data sources used to compute the deceleration guidance and generate the displays are discussed. Examples of the display formats and symbology options are presented. Logic diagrams describing the design of the ROTO software module are also given.

14. SUBJECT TERMS
roll-out and turn-off, HUD, guidance, flight deck displays, flight test, simulation, aircraft systems integration, aircraft deceleration

15. NUMBER OF PAGES
56

16. PRICE CODE
A04

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL