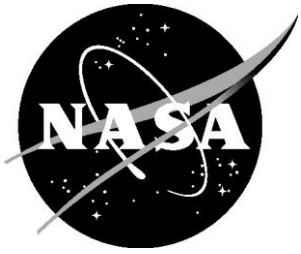
NASA/TM-2013-217804



The Simplified Aircraft-Based Paired Approach with the ALAS Alerting Algorithm

Raleigh B. Perry, Michael M. Madden, Wilfredo Torres-Pomales, and Ricky W.Butler
Langley Research Center, Hampton, Virginia

February 2013

NASA STI Program . . . 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 plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and 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 also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

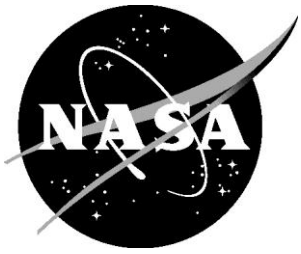
- E-mail your question to help@sti.nasa.gov

- Fax your question to the NASA STI Information Desk at 443-757-5803

- Phone the NASA STI Information Desk at 443-757-5802

- Write to:
  STI Information Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076-1320

NASA/TM-2013-217804



The Simplified Aircraft-Based Paired Approach with the ALAS Alerting Algorithm

Raleigh B. Perry, Michael M. Madden, Wilfredo Torres-Pomales, and Ricky W. Butler
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

February 2013

Acknowledgments

The authors want to thank Cesar Munoz for his important contributions to the design of the ALAS algorithm. He developed the ALAS_lines and ALAS_circle functions in the PVS theorem prover and established some key mathematical properties of these functions. These functions are conflict probes that are of primary importance to the algorithm. He also contributed to the design of the control structure of this algorithm.

The authors also want to thank Jeffrey Maddalon for the technical insight he provided that helped us create a practical alerting algorithm.

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
443-757-5802

Abstract

This paper presents the results of an investigation of a proposed concept for closely spaced parallel runways called the Simplified Aircraft-based Paired Approach (SAPA). This procedure depends upon a new alerting algorithm called the Adjacent Landing Alerting System (ALAS). This study used both low fidelity and high fidelity simulations to validate the SAPA procedure and test the performance of the new alerting algorithm. The low fidelity simulation enabled a determination of minimum approach distance for the worst case over millions of scenarios. The high fidelity simulation enabled an accurate determination of timings and minimum approach distance in the presence of realistic trajectories, communication latencies, and total system error for 108 test cases. The SAPA procedure and the ALAS alerting algorithm were applied to the 750-ft parallel spacing (e.g., SFO 28L/28R) approach problem. With the SAPA procedure as defined in this paper, this study concludes that a 750-ft application does not appear to be feasible, but preliminary results for 1000-ft parallel runways look promising.

Table of Contents

Acronyms

ADS-B	Automatic Dependent Surveillance - Broadcast
AGL	Above Ground Level
AILS	Airborne Information for Lateral Spacing
ALAS	Adjacent Landing Alerting Systems
ALT HLD	Altitude Hold mode
API	Application Programming Interface
APP	Approach mode
ATC	Air Traffic Control
CAS	Calibrated Air Speed
CMF	Cockpit Motion Facility
DOT	Department of Transportation
EPU	Estimated Position Uncertainty
FAA	Federal Aviation Administration
FAF	Final Approach Fix
GPS	Global Positioning System
HDG	Heading mode
HDG SEL	Heading Select mode
HITL	Human-in-the-Loop
IGE	In-Ground Effect
ILS	Instrument Landing System
IMC	Instrument Meteorological Conditions
IS	Intruder Ship
LOC	Localizer mode
MASPS	Minimum Aviation System Performance Standard
MCP	Mode Control Panel
MOPS	Minimum Operational Performance Standard
NACp	Navigation Accuracy Category for Position
NACv	Navigation Accuracy Category for Velocity
NGE	Near Ground Effect
OGE	Out-of-Ground Effect
OS	Ownship
PRM	Precision Runway Monitor
RNAV	Area Navigation
RTCA	Radio Technical Commision for Aeronautics
RWY	Runway
SAP	Stabilized Approach Point
SAPA	Simplified Aircraft-based Paired Approach
SFO	San Francisco International Airport
SPD	Speed mode
tALAS	Test simulator for ALAS
TAS	True Air Speed
TCA	Time of Closest Approach
TCAS	Traffic Collision Avoidance System
TSE	Total System Error
UTC	Coordinated Universal Time
VERT SPD	Vertical speed mode
VMC	Visual Meteorological Conditions
WAAS	Wide Area Augmentation System

1. Introduction

The Simplified Aircraft-based Paired Approach (SAPA) is a proposed concept for the operation of closely spaced parallel runways in Instrument Meteorological Conditions (IMC). SAPA offers an important opportunity for a significant increase in the rate of flight operations that approaches the arrival rate achievable under Visual Meteorological Conditions (VMC) [Johnson2010]. Based on constant-width navigation performance, the SAPA concept leverages advanced navigation and flight-guidance technology and Automatic Dependent Surveillance – Broadcast (ADS-B) to share precise position and velocity data between the paired aircraft. The SAPA concept allows one aircraft to pass the other aircraft during the approach segment, while keeping the paired aircraft within a defined conformance zone to avoid any wake vortex encounters.

Using the SAPA concept, aircraft are initially established on final approach with a minimum of 1000 ft of vertical separation and with Air Traffic Control (ATC) responsible for initially pairing the aircraft with appropriate relative longitudinal positioning. During SAPA operations, the paired aircraft utilize onboard flight guidance speed cues to maintain longitudinal alignment within the conformance zone, and an escape maneuver (climbing turn away from the paired aircraft) is required when either lateral or longitudinal position error is beyond tolerance, or there is a loss of ADS-B or other required flight-navigation capability.

Conducted under FAA reimbursable funding, this study (Phase I) focused on the development and validation testing of an on-board algorithm that alerts intrusions from paired aircraft. The new algorithm, called the Adjacent Landing Alerting System (ALAS), was developed and tested for the SAPA concept adapted to the close parallel runway spacing (750 ft) of Runways 28L and 28R at San Francisco International Airport (SFO), and it is adaptable to other parallel runway spacings with greater separations. The alerting algorithm was tested using low fidelity (kinematic) and high fidelity simulation capabilities. Phase I deliverables consist of the alerting algorithm software source code, and this report documenting the alerting algorithm development and validation testing.

An optional future study (Phase II) is defined that would further refine the alerting algorithm for operational robustness. Phase II would examine in more detail the system architecture required to support SAPA operations including ATC/pilot procedures and the escape maneuver. A preliminary simulation plan would also be developed with input from the DOT/FAA Team for a future Human-in-the-Loop (HITL) simulation study that leverages experience from earlier Airborne Information for Lateral Spacing (AILS) and other closely spaced operations simulation studies.

2. The SAPA Procedure

2.1. SAPA Concept

The Simplified Aircraft-based Paired Approach (SAPA) procedure allows two aircraft to perform parallel approaches under instrument conditions on runways spaced as close as 750 ft apart. Johnson et al. [Johnson2010] describe the basic concept of the SAPA procedure as illustrated in Figure 2-1. SAPA allows the paired aircraft to have different approach speeds. The "faster" aircraft has the higher approach speed; the "slower" aircraft has the lower approach speed. If both aircraft have the same approach speed, the aircraft that begins the procedure in the

trailing position assumes the "faster" aircraft role. The "faster/slower" designation applies only to approach speed; prior to establishing approach speed, the "faster" aircraft may flight at or slower than the speed of the "slower" aircraft to conform to the procedure.

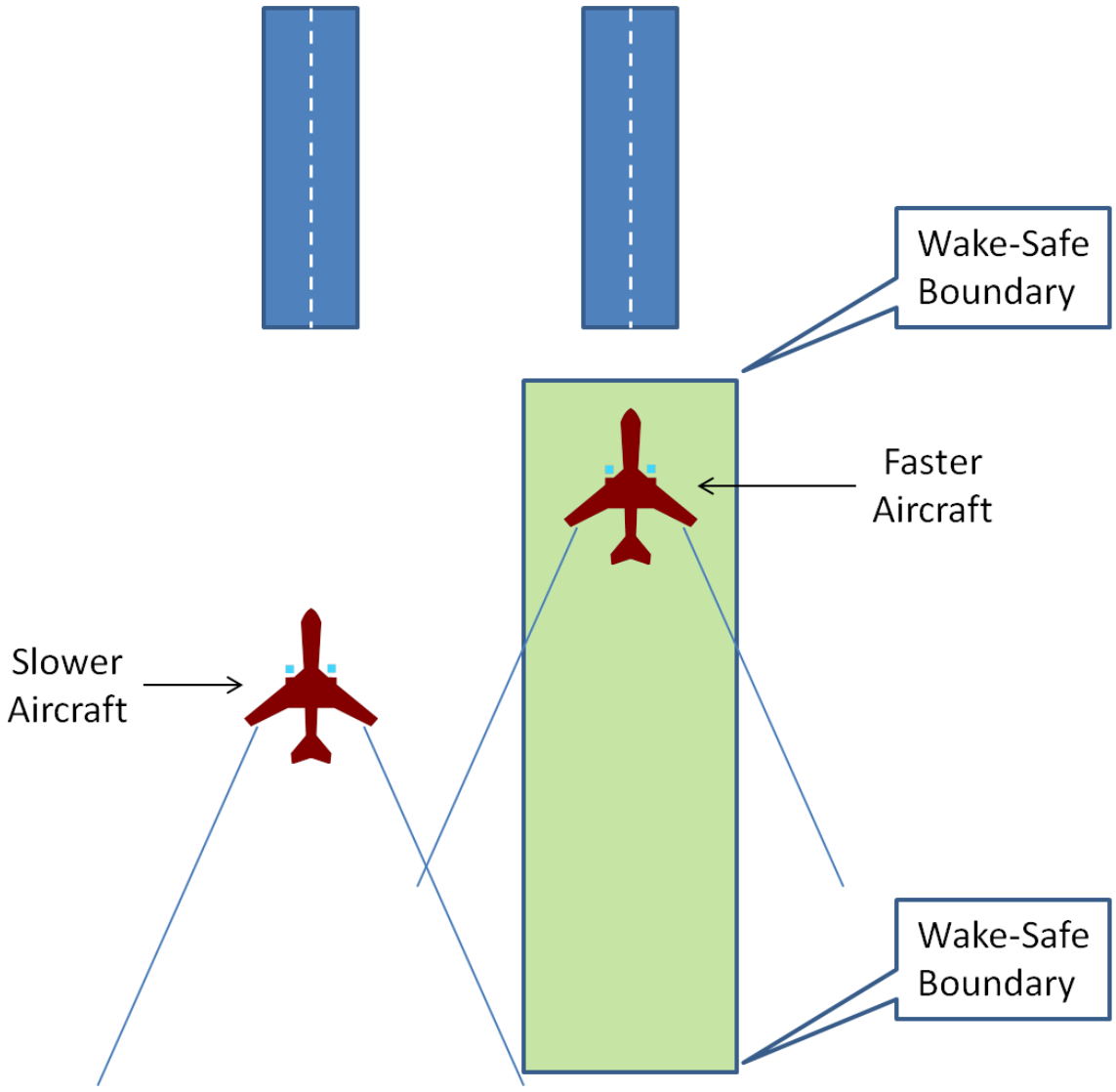


Figure 2-1: SAPA Concept

From the start of the procedure until touchdown, the relative along-track positions of the aircraft must remain within a forward and rear boundary that avoids wake vortex encounters. The rear boundary represents the furthest trailing distance where the faster aircraft avoids the wake from the slower aircraft. The forward boundary represents the furthest leading distance where the slower aircraft avoids the wake of the faster aircraft. The faster aircraft begins the procedure in a trailing position relative to the slower aircraft. The faster aircraft maintains this trailing position

until the slower aircraft reaches the final approach fix (FAF) and begins to decelerate to final approach speed. The faster aircraft is then permitted to pass the slower aircraft before landing.

The procedure can be divided into four segments:

- Initiation – Air traffic control (ATC) vectors each aircraft onto the approach. One aircraft is placed at a 1000-ft vertical separation from the other. The higher aircraft can be the faster or slower aircraft. The faster aircraft must establish initial along-track separation before the higher aircraft begins to descend on the glidepath. This segment was not evaluated in the study.

- Constant Speed – The slower aircraft maintains a constant airspeed as assigned by Air Traffic Control (ATC) until reaching the FAF. The faster aircraft adjusts speed to maintain separation. In this segment, both the high and low aircraft will initially travel straight and level on the runway approach course until each aircraft intercepts their glidepath. Each aircraft will then descend on the glidepath. The aircraft initially use Traffic Collision Avoidance System (TCAS) for collision avoidance until the vertical separation drops below 800 ft. At that point, the aircraft suppress TCAS alerts and rely on the ALAS algorithm for collision avoidance.

- Deceleration – At the FAF, the slower aircraft begins its deceleration to final approach speed. The faster aircraft performs one of two actions: 1) continue at constant speed until reaching the FAF and then decelerate to final approach speed or 2) continue to match the ground speed of the slower vehicle as the slower vehicle decelerates until the faster vehicle reaches its final approach speed. Johnson et al. [Johnson2010] refer to the latter option as 'speed management'; the former option will be referenced as "no speed management". Each aircraft is expected to stabilize on its final approach speed before the stabilized approach point (SAP), a point on the glidepath at 1000 ft above ground level (AGL).

- Approach Speed – Both aircraft fly their final approach speed to the runway threshold. Then they decelerate and flare to landing speed. The ALAS algorithm deactivates alerts once decision height is reached.

The speed schedule of the last three segments to decision height is depicted for the speed management option in Figure 2-2.

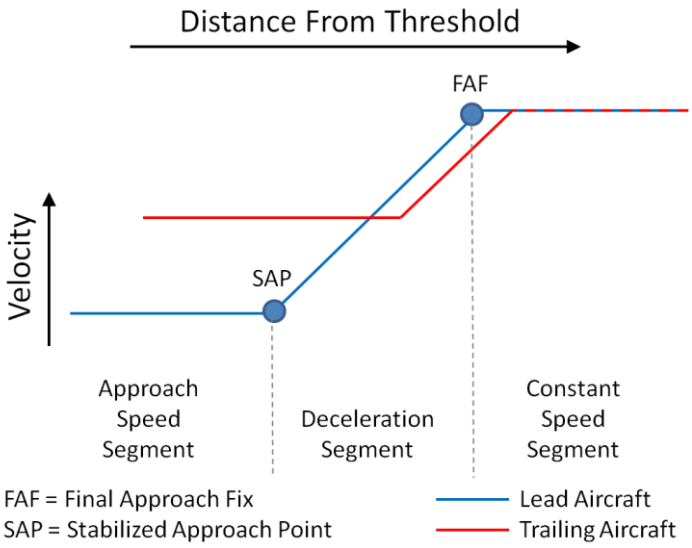


Figure 2-2: Velocity Profile of SAPA Procedure

2.2. Applying SAPA Concept to San Francisco International Airport

San Francisco International Airport has a pair of runways, 28L and 28R, that are 750 ft apart and currently used for parallel approaches under Precision Runway Monitor (PRM) procedures. In this study, the SAPA concept is applied to these runways. The first consequence of the SAPA concept is that the low altitude vehicle needs a long level segment prior to capturing its glideslope. Using a 3° glidepath and 1000-ft vertical separation, the low altitude aircraft must be at least 3.14 NM from its glidepath when the higher-altitude aircraft intercepts the glidepath. Additional distance will be required to allow the faster aircraft to stabilize its separation prior to intercepting the glidepath. This study does not examine the ATC scheduling of SAPA aircraft onto the approach. Instead, scenarios begin with the aircraft stabilized on initial separation. Nevertheless, the existing area navigation (RNAV) approaches on runways 28L and 28R provide a long level segment that extends at least 5.4 NM [FAA2012]. Figure 2-3 shows the vertical profile of the SAPA procedure for runways 28L and 28R with the higher aircraft assigned to 28L for illustration; the slower or faster aircraft can be assigned to 28L. (The low altitude of 4100 ft was chosen to be coincident with the ILS approach for RWY 28R since the high fidelity simulation conducts the approach using ILS.)

The scenario starts with the high aircraft at PONKE. The low aircraft is positioned at the appropriate initial separation between waypoints DUMBA and MEHTA. The scenario starts in the constant speed segment of the procedure. Both aircraft travel straight and level on the runway approach heading. The high aircraft will intercept the glidepath at approximately 15.7 NM from the threshold. At approximately 15.0 NM from the runway threshold, each aircraft will switch from TCAS to ALAS for collision avoidance. At approximately 12.5 NM, the slower aircraft will intercept its glidepath and begin to descend. From this point forward, the vertical separation of the two aircraft will be less than 160 ft due to glidepath geometry. The aircraft continue on the glidepath at constant speed until the slower aircraft reaches its FAF at AXMUL. This begins the

deceleration segment of the SAPA approach. The slower aircraft will then decelerate to its final approach speed. Under speed management, the faster aircraft will begin its deceleration either upon recognizing the deceleration of the slower aircraft or upon reaching its FAF at DUYET, whichever occurs first. With no speed management, the faster aircraft will begin its deceleration at DUYET. The final approach segment of the SAPA procedure begins when both aircraft stabilize on their final approach speed. When the aircraft reach the decision height, the ALAS algorithm deactivates alerts. Both aircraft proceed at their final approach speed to the runway threshold, then decelerate and flare to landing speed.

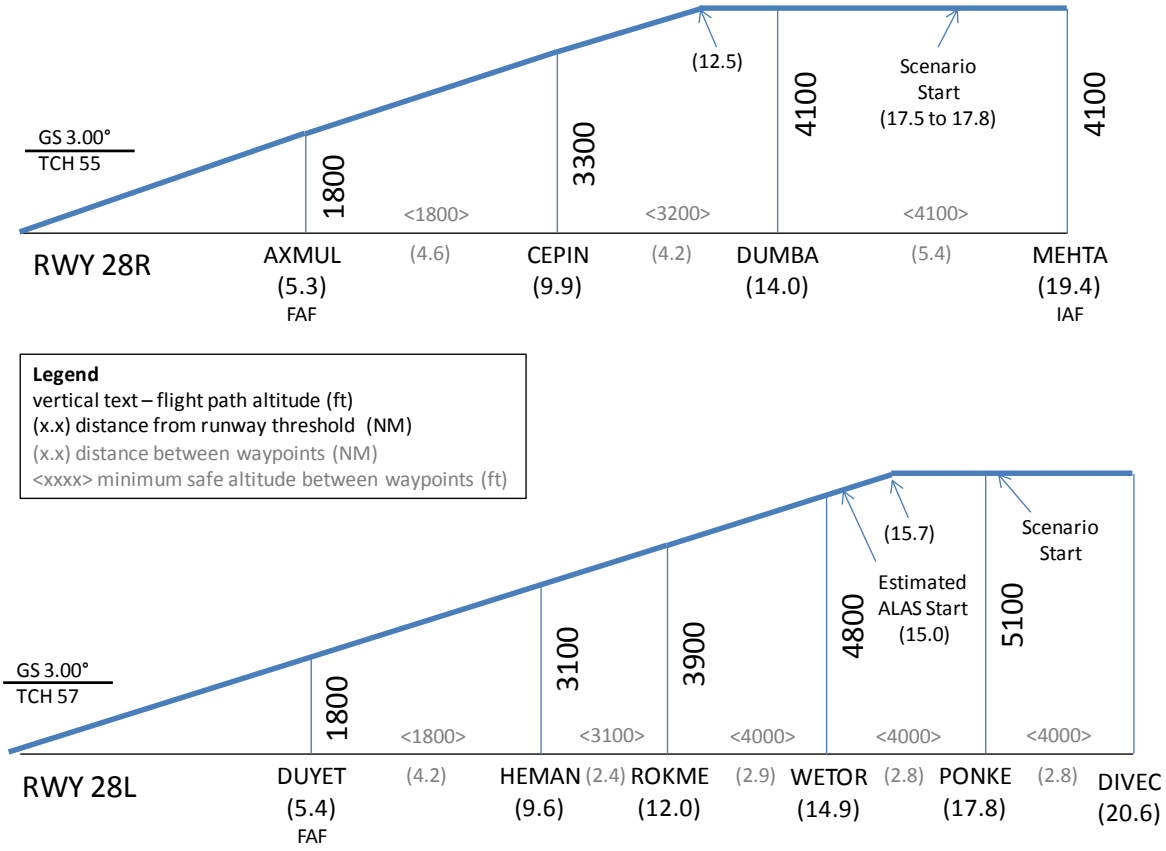


Figure 2-3: Vertical Profile of SAPA Approaches at SFO

2.3. Initial Separation at the FAF

The SAPA procedure requires that the along-track separation of the participating aircraft remain within wake-safe boundaries throughout the procedure. Ground effects influence the transport of wake vortices. Therefore, the wake-safe boundary changes with altitude. Johnson et al. [Johnson2010] defined the wake-safe boundary for three altitude regions: in-ground effect (IGE), near-ground effect (NGE), and out-of-ground effect (OGE). These regions are depicted in Figure 2-4. Note that distances are measured from the glideslope intercept and not the runway threshold. At SFO runways 28L and 28R, the intercept of the ILS glideslope occurs at the standard distance of ~1000 ft down the runway.

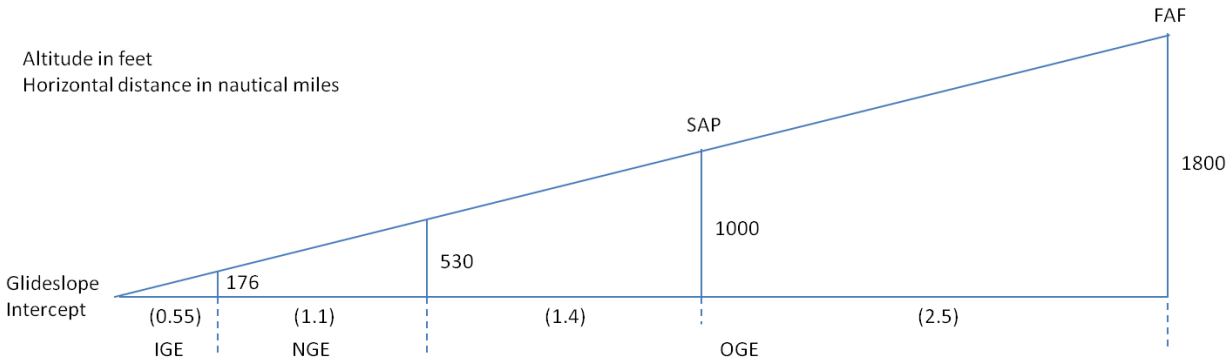


Figure 2-4: Regions with Different Wake-Safe Boundaries

The wake study of Johnson et al. [Johnson2010] looked at the worst-case condition of two Boeing 747-8 aircraft traveling with a 15 KT adverse crosswind to parallel runways with 750 ft spacing. Under these conditions, the wake-free boundary for each region is given in Table 2-1. The boundary is defined as a longitudinal distance of the faster aircraft from the slower aircraft. The boundary extends fore (slower aircraft avoids wake of faster aircraft) and aft (faster aircraft avoids wake of slower aircraft).

Table 2-1: Wake-Safe Boundaries

Region	IGE	NGE	OGE
Wake Safe Boundary (ft)	1000	2600	3000

In dependent operations, the aircraft could use the full length of the conformance zone. However, within the SAPA procedure, the aircraft transition to independent operation within the deceleration segment, i.e. at or shortly after reaching the FAF. Therefore, the aircraft must be positioned at the FAF such that their independent operations do not cause either aircraft to exceed the wake-safe boundary before touchdown. Prior to the FAF, the SAPA concept requires the trailing aircraft to maintain speed (and, therefore, along-track separation) with the lead aircraft; thus, the along-track separation at the FAF is also the initial separation. Since the SAPA concept defines the velocity profile of each aircraft, determining the FAF separation window simply requires a kinematic back-trace from touchdown to the FAF. Under ideal conditions, the FAF separation window is a function of the wake-safe boundaries, the final approach speeds of the aircraft, the assigned speed for the constant speed segment, the glidepath angle, and, in the speed management case, the latency of dependent operations. This study looked at final approach speeds for the slower aircraft ranging from 110 KT to 155 KT. The faster aircraft declares an approach speed equal to or greater than the slower aircraft. How much faster the faster aircraft could fly the approach would be answered by this exercise. SFO uses the standard 3° glidepath for runways 28L and 28R. The assigned speed for the constant speed segment was set at 170 KT. To define a latency of operations, a 1 s ownship response was added to the allowable 2 s latency for ADS-B OUT under the FAA Rule [FAA2010]; this results in a total latency of 3 s. Additionally, the kinematic back-trace was simplified. True and calibrated airspeeds (TAS and CAS) were treated as equal from the SAP to touchdown. Aircraft speed in the deceleration segment was modeled as a constant deceleration from the true airspeed at the FAF to the approach speed at the SAP for the slower aircraft and, when speed management was not used, for the faster aircraft. Under the speed management case, the faster aircraft matched the deceleration

of the slower aircraft. Deceleration and flare to landing speed were not modeled. All cases were run with zero winds. Accommodating constant winds is a straightforward addition but is reserved for future work. The separation error that the TAS = CAS simplification injects is very small since the contribution from each aircraft partially cancels out. For differences in speed between fast and slower aircraft of up to 20 KT, the injected error is estimated to be less than 10 ft of separation in the no speed management case and less than 25 ft in the speed management case. The separation error for not modeling the flare is less than 16 ft. Otherwise, the remaining separation error in the kinematic model depends on the ability of real aircraft to fly the speed schedule in the model; therefore, the remaining separation error is a function of the total system error (TSE) for each aircraft.

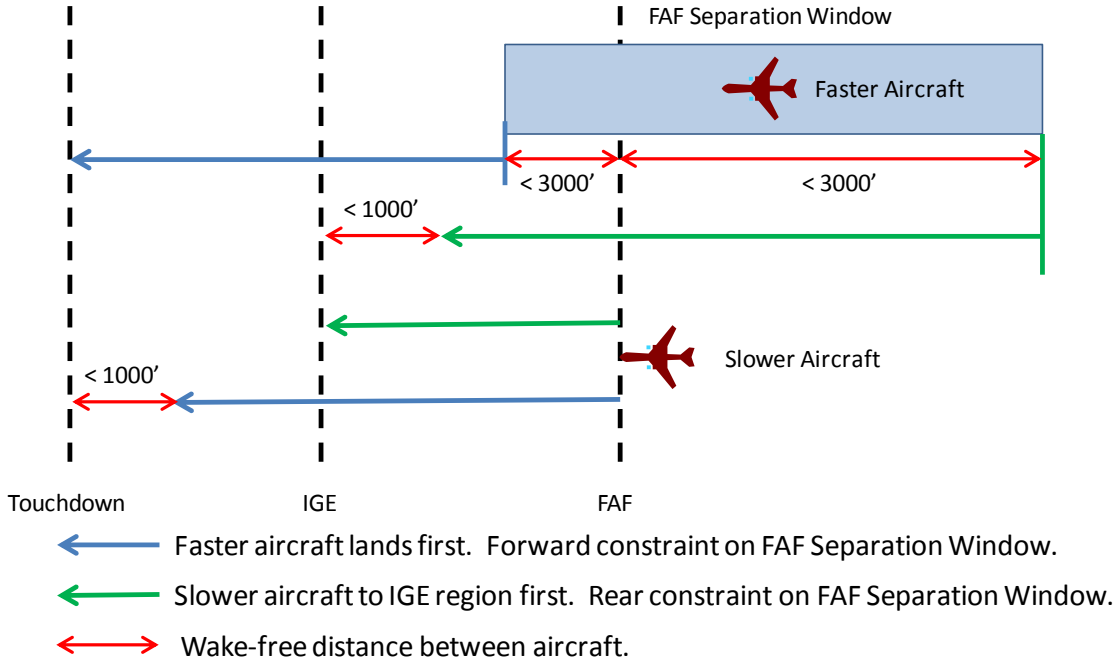


Figure 2-5: FAF Separation Window

Figure 2-5 depicts the constraints that define the forward and rear edge of the FAF separation window. The forward edge of the FAF separation window is constrained by two conditions: 1) the faster aircraft cannot be ahead of the slower aircraft by more than the IGE wake-safe boundary at touchdown and 2) the faster aircraft cannot be ahead of the slower aircraft by more than the OGE wake-safe boundary at the FAF. The rear edge of that window is constrained by two conditions: 1) at the IGE/NGE transition, the faster aircraft can be no further back than the IGE wake-safe boundary and 2) the faster aircraft can be no further back than the OGE wake-safe boundary at the FAF. Due to the geometry of the wake-safe boundaries with distance, the NGE wake-safe boundary does not impact the FAF separation window. A faster aircraft near the NGE wake-safe distance at the OGE/NGE transition would have to fly so much faster than the slower aircraft to meet the IGE wake-safe distance at the IGE/NGE transition that the faster aircraft would have to begin beyond the OGE wake-safe distance at the SAP. The constraints above are conservative because they do not take into account that one of the aircraft will be downwind in an adverse crosswind. For example, the forward edge constraint that the faster aircraft cannot be ahead of the slower aircraft by more than the IGE wake-safe distance at touchdown and the rear

edge constraint that the faster aircraft cannot be behind the slower aircraft by more than IGE wake-safe distance at IGE/NGE transition apply the same 15 KT adverse cross-wind wake-safe boundary to both aircraft. This application of the constraints enables execution of the procedure without introducing into the decision process a new variable of aircraft position relative to wind. To treat one aircraft as downwind in the constraints would require an additional wake study to establish the wake-safe boundaries for light and variable winds (to handle the worst case of no defined down-wind side).

Table 2-2 and Table 2-3 show the results for the case of the slower aircraft approaching at 110 KT for the cases without and with speed management respectively. The 110 KT scenario was selected because it provides worst-case results for large speed differences. This data is also depicted pictorially in Figure 2-6 and Figure 2-7.

Table 2-2: FAF Separation Window w/ No Speed Management – Slower Aircraft at 110 KT Approach Speed

Increase in Final Approach Speed for Faster Aircraft (KT)	FAF Separation Window			
	Forward Edge (ft)	Rear Edge (ft)	Length (ft)	Error-Adjusted Rear Edge (ft)
0	+1540	-1536	3103	-1350
5	-58	-2888	2830	-2702
10	-1560	-3000	1440	-2814
15	-2957	-3000	43	None
20	None	None	None	None

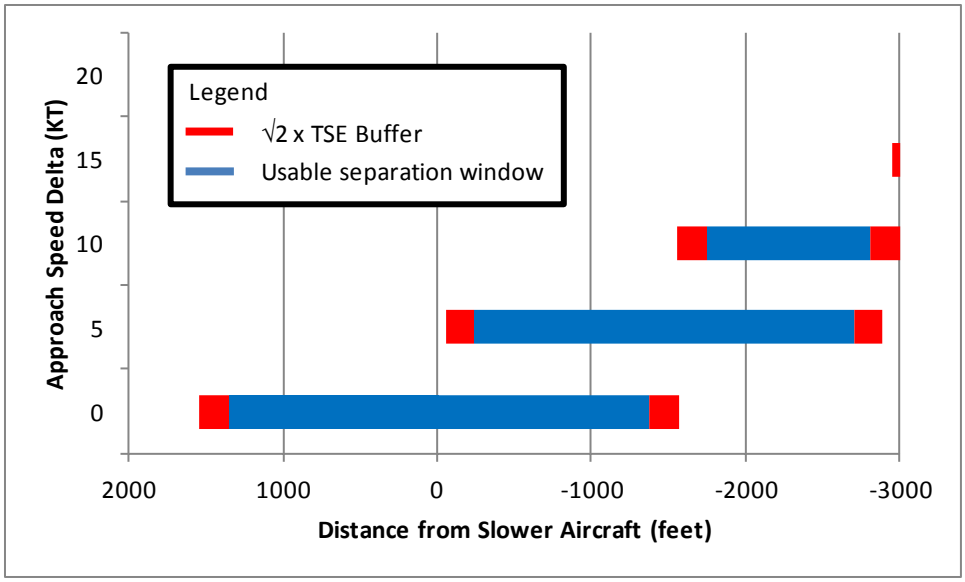


Figure 2-6: FAF Separation Window w/ No Speed Management – Slower Aircraft at 110 KT Approach Speed

Table 2-3: FAF Separation Window w/ Speed Management – Slower Aircraft at 110 KT Approach Speed

Increase in Final Approach Speed for Faster Aircraft (KT)	FAF Separation Window			
	Forward Edge (ft)	Rear Edge (ft)	Length (ft)	Error-Adjusted Rear Edge (ft)
0	+676	-1318	1994	-1132
5	-58	-2029	1971	-1843
10	-998	-2783	1786	-2597
15	-1903	-3000	1097	-2814
20	-2852	-3000	148	-2814

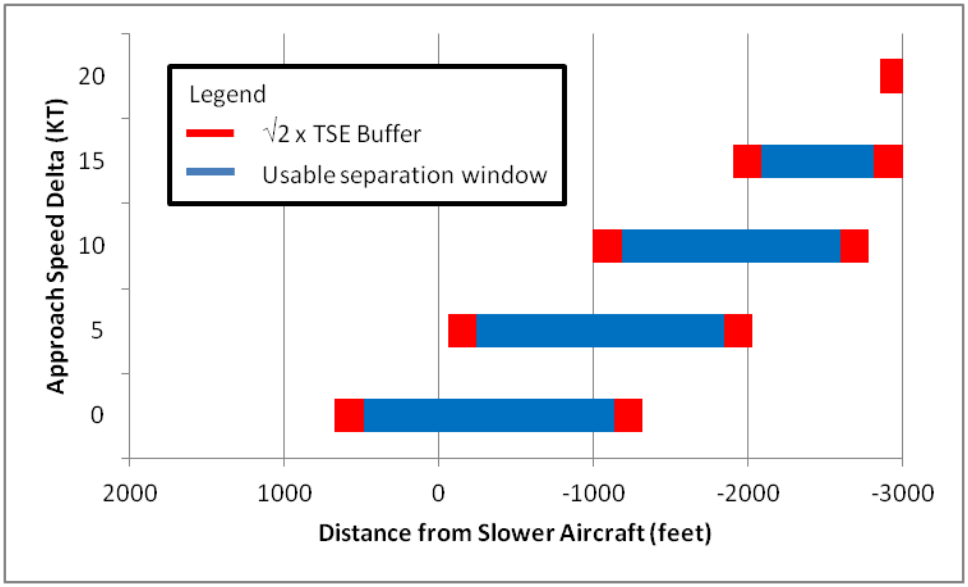


Figure 2-7: FAF Separation Window w/ Speed Management – Slower Aircraft at 110 KT Approach Speed

Three conclusions can be drawn from the data. First, the scenario without speed management provides a larger separation window when the approach speeds of the two aircraft differ by less than 10 KT. Second, speed management is necessary to make the procedure available to aircraft pairs with a speed difference greater than 10 KT while still providing an adequate window for low speed differences. Therefore, to keep the SAPA procedure simple, speed management can be used as the sole mode of operation. Third, even within speed management, there is no single separation window that accommodates all speed differences. The initial separation must be tailored for the approach speeds of both aircraft.

As described above, this data is still subject to the TSE of each aircraft. TSE values are defined as the 95% (2σ) expected deviation of the aircraft true position from the flight plan. A

simple means of incorporating TSE into the FAF separation window is to reduce the forward and rear edge by $\sqrt{2}$ x TSE[1] each to produce a separation window that provides 99.99% confidence of remaining in the conformance zone to touchdown; thus, the window must be at least $2\sqrt{2}$ x TSE in length to be feasible. Johnson et al. [Johnson2010] argue, on the basis of maintaining sufficient cross-track separation, that the maximum allowable TSE for the SAPA procedure with 750-ft runway spacing is 131 ft (40 m). Therefore, the FAF separation window must be at least 371 ft (113 m) long to be feasible. The kinematic back-trace indicates that the largest speed difference the SAPA procedure can accommodate based on this criteria is 18 KT. The last column of Table 2-2 and Table 2-3 show the rear edge of the window after applying $\sqrt{2}$ TSE error. Figure 2-6 and Figure 2-7 also show a forward and rear buffer of $\sqrt{2}$ TSE in red and the remaining usable FAF separation window in blue. In all cases where the SAPA procedure is feasible, the error-adjusted rear-edge of the window provides an initial horizontal separation of at least 1000 ft.

However, TSE may not be the best metric for adjusting the separation window to account for flight technical error and navigation error. TSE is a position metric. Along-track separation from FAF to touchdown is determined by the speed schedule. Therefore, the separation window should be most sensitive to errors in following the speed schedule. The results in Figure 2-6 and Figure 2-7 can be used to illustrate this. Take the case where the declared speed difference is 10 KT. If the aircraft errors in following the speed schedule produce a ±5 KT uncertainty in the speed difference, then the FAF separation window that can accommodate this error is approximately the window formed from the overlap of the windows for the 5 KT, 10 KT and 15 KT speed differences. The overlap produces a range of -1903 ft to -2029 ft, which has a length of 126 ft. This is an order of magnitude smaller than the windows for an exact speed difference of 10 KT. For a declared difference of 15 KT, an uncertainty of ±5 KT cannot be accommodated because no overlap exists with the 10 KT and 20 KT separation windows. To examine the effect of velocity tracking errors on the FAF separation window, the kinematic model was modified to compute the FAF separation for each declared speed difference from the overlap of three windows: error low, no error, and error high. In the error low case, the initial speed at the start of the deceleration segment and the final approach speed for the faster aircraft are reduced by the defined velocity bias, and the initial speed at deceleration and the final approach speed for the slower aircraft are increased by the defined velocity bias. In the no error case, no errors are added to the speeds of the slow and faster aircraft. In the high error case, the defined error is added to the initial speed at deceleration and to the final approach speed of the faster aircraft, and the velocity error is subtracted from the initial speed at deceleration and the final approach speed of the slower aircraft. The low error scenario for the 0 KT declared speed difference requires an additional adjustment. In this scenario, the 'fast' aircraft will actually be slower than the 'slow' aircraft. This opens the possibility that the 'fast' aircraft can actually fall beyond the IGE wake safe difference in the IGE region. This necessitates a third constraint for computing the rear edge of the FAF separation window: if the slower aircraft touches down first, then the trailing aircraft must be no further behind than the IGE wake-safe boundary.

The revised kinematic model employs the mean velocity error (i.e., velocity bias), averaged over the distance from FAF to touchdown. It assumes contributions from the remaining instantaneous random fluctuations are negligible. The velocity bias includes contributions from

[1] The TSE of one aircraft is not correlated with the other. Therefore, the error in the along track separation of the two aircraft for a given confidence value is the square root of the sum of the squares of the TSE for each aircraft.

navigation error and flight technical error. Establishing a realistic value, however, is a challenge. The only FAA requirement on velocity performance is the minimum NACv requirement for ADS-B OUT. But NACv only identifies the upper bound of the 95% navigation accuracy for the reported velocity, and navigation accuracy is often assumed to have a mean of zero [Mohleji2010]. Any significant bias must, therefore, come from flight technical error, i.e., the difference between the commanded velocity and the estimated flown velocity. No sources were found that define flight technical error for velocity in modern aircraft. However, Johnson et al. [Johnson2010] used a ±2 KT uniform uncertainty for velocity in the Monte-Carlo simulation that established the wake-safe boundaries. This uncertainty was randomly computed for each aircraft and persistently applied throughout the run. Once the velocity bias is selected, navigation uncertainty for position is used to assess the feasibility of the separation window. The separation window must still be longer than the uncertainty in separation due to the position navigation accuracy of each aircraft. The chosen navigation accuracy was 33 ft (10 m) which is the estimated position uncertainty (EPU) of the largest NACp category in ADS-B reports that is compatible with the maximum eligible TSE of 40 m for the SAPA procedure. The 99.99% uncertainty bounds are therefore $2\sqrt{2}$ EPU = 93 ft (28.3 m). Using this criteria, the SAPA procedure is infeasible at all velocity differences for a velocity bias of ±2 KT. The maximum velocity bias to retain a feasible SAPA procedure for approach speed differences up to 15 KT is ±1.45 KT for each aircraft (i.e., a speed difference uncertainty of ±2.9 KT).

Table 2-4 and Table 2-5 show the separation window that results from applying a velocity bias of 1.45 KT and adjusting the rear edge of the separation window by $\sqrt{2}$ EPU. Figure 2-8 and Figure 2-9 provide a pictorial view of the data. This separate application of velocity and position errors produces a more tightly constrained window for the initial separation. Moreover, the window for the 0 KT difference scenario positions the aircraft with less than 400 ft of along-track separation before the FAF.

Table 2-4: FAF Separation Window w/ No Speed Management – ±1.45 KT Velocity Bias and 10 m EPU

Increase in Final Approach Speed for Faster Aircraft (KT)	FAF Separation Window			
	Forward Edge (ft)	Rear Edge (ft)	Length (ft)	Error-Adjusted Rear Edge (ft)
0	+428	-423	851	-376
5	-1166	-1904	738	-1857
10	-2646	-3000	354	-2954
15	None	None	None	None
20	None	None	None	None

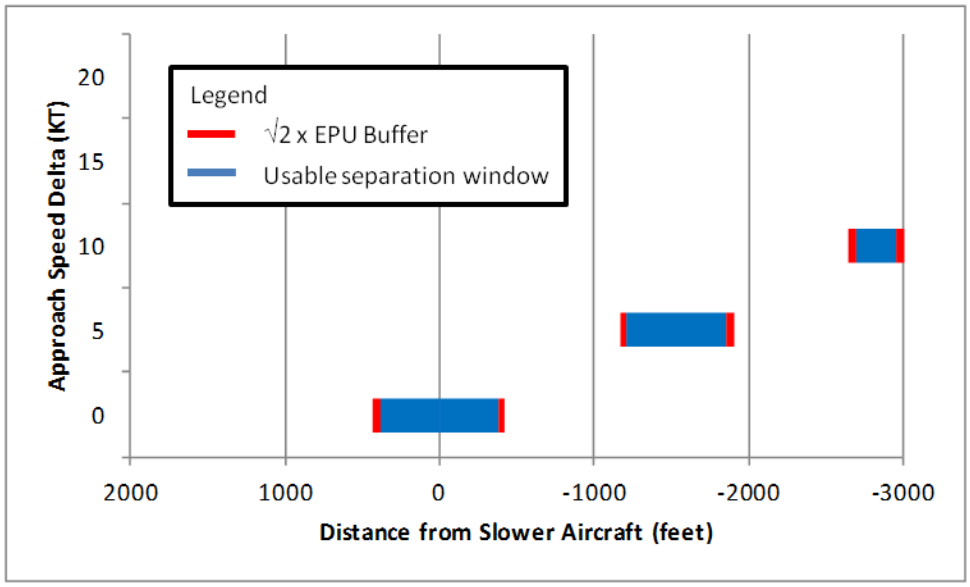


Figure 2-8: FAF Separation Window w/ No Speed Management – ±1.45 KT Velocity Bias and 10 m EPU

Table 2-5: FAF Separation Window w/ Speed Management – ±1.45 KT Velocity Bias and 10 m EPU

Increase in Final Approach Speed for Faster Aircraft (KT)	FAF Separation Window			
	Forward Edge (ft)	Rear Edge (ft)	Length (ft)	Error-Adjusted Rear Edge (ft)
0	+428	-423	851	-376
5	-943	-1302	359	-1256
10	-1816	-2047	231	-2001
15	-2733	-2836	103	-2790
20	None	None	None	None

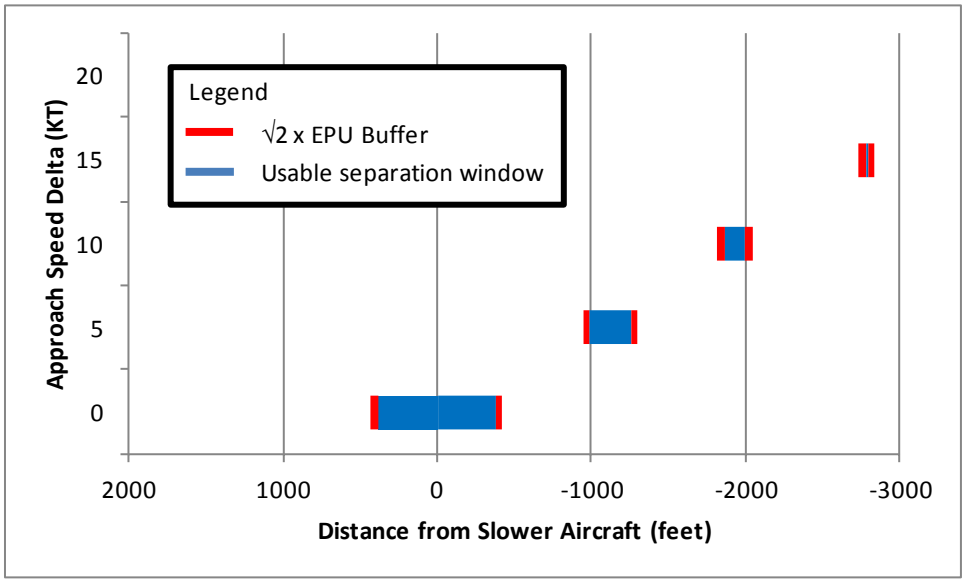


Figure 2-9: FAF Separation Window w/ Speed Management – ±1.45 KT Velocity Bias and 10 m EPU

2.4. Options for Loosening Separation Contraints

Though the conformance zone has a length of 6000 ft at the FAF, kinematic back-trace of each aircraft from touchdown to the FAF shows that the usable portion of this zone is much smaller, sometimes as little as about 100 ft. In addition, some of these separation windows place the aircraft pair in close proximity for the 10+ NM stretch from loss of vertical separation to the FAF. The constraint with the greatest influence on the separation window is the 1000 ft IGE wake-safe boundary. Here are some examples of how extending the IGE wake-safe boundary impacts results.

- An IGE wake-safe boundary of 1275 ft opens the procedure to aircraft with a velocity bias of ± 2 KT (and approach speed differences of up to 15 KT).

- An IGE wake-safe boundary of 1450 ft allows the trailing aircraft to begin with an along-track separation greater than 1000 ft when the aircraft pair has the same approach speed (and

velocity bias is within 1.45 KT).

- An IGE wake-safe boundary of 1650 ft opens the procedure to approach speed differences of 20 KT (and velocity bias of each aircraft is within 1.45 KT).

Adjustments other than IGE wake-safe boundary can be made to the procedure to make better use of the separation windows that result from the wake-safe boundaries. Listed below are options to extend the usable separation window or to increase the procedure's tolerance of velocity or position errors.

- Position aircraft based on predicted winds and customize separation as appropriate. As stated in the previous section, the kinematic back-trace computes the separation windows as if both aircraft experience adverse crosswind; this allows the procedure to be performed without adjusting for wind direction. However, the procedure could include the predicted winds into decision-making and tailor the separation appropriately. For example, controllers could be required to place the faster aircraft on the downwind runway or the SAPA avionics on the faster aircraft could compute separation based on predicted winds along the path. Placing the faster aircraft on the downwind runway will extend the forward edge of the FAF separation window. Though this could open the procedure to 20 KT approach speed differences or increase tolerance to a ±2 KT velocity bias, it does not allow the faster aircraft to position itself any further back in those cases where the rear edge of the separation window is less than 1000 ft. Placing the slower aircraft downwind would allow the faster aircraft to position itself further back but, unless the faster aircraft remains downwind from initiation to touchdown, the faster aircraft can still start no further back than the OGE wake-safe boundary of 3000 ft. Thus, placing the faster aircraft downwind can fail to open the procedure to higher velocity differences or significantly improve tolerance to velocity bias.

- Decrease allowable adverse wind speed. This option trades availability with respect to time against availability with respect to aircraft pairings. To assess this trade will require wake studies with different adverse winds.

- Add a segment prior to the FAF to maneuver into the preferred separation at the FAF. The trailing aircraft would initiate separation near the OGE wake-safe boundary and maintain this separation until some specified distance before the FAF. The faster aircraft would then accelerate to the preferred separation at the FAF. This would allow the faster aircraft to remain well behind the slower aircraft for much of the 10+ NM distance between procedure initiation and the FAF.

- When aircraft request the same approach speed, the controller directs the trailing aircraft to increase its planned approach speed by 5 KT. An approach speed difference of 0 KT becomes a troublesome case because the velocity bias causes the trailing aircraft to be the 'slow' aircraft. This forces the separation window to remain in close proximity to the abeam position. Requiring the trailing aircraft to have a greater planned approach speed should guarantee that the trailing aircraft will remain the faster aircraft in the presence of velocity bias.

3. The ALAS Alerting Algorithm

ALAS (Adjacent Landing Alerting System) is an alerting algorithm designed to detect intrusions on closely spaced parallel runways. It employs a mechanism for detecting imminent intrusions into a protection zone (analogous to AILS [Abbott2002]) and a mechanism for detecting lateral deviations from the runway centerline in a manner similar to the Precision Runway Monitor system [Shank1994]. The algorithm is highly configurable through a set of user-specifiable parameters.

3.1. Structure of the Intrusion Detection Algorithm

The intrusion detection algorithm in ALAS uses a trigger mechanism based on the rate of change of the track angle of the intruder to initiate a sweep of potential intrusions as shown in Figure 3-1. This mechanism is augmented with two additional tests. The first test is a simple conflict probe. The conflict probe detects if the current velocity vector will intersect the ownship's front or back buffer. For the second test, illustrated in Figure 3-2, the horizontal two-dimensional distance between the aircraft is checked to see if it is less than the distances for a red alert (absDistRed) and a yellow alert (absDistYellow). A red alert is issued if the distance is less than absDistRed. If the distance is between absDistYellow and absDistRed, a yellow alert is issued.

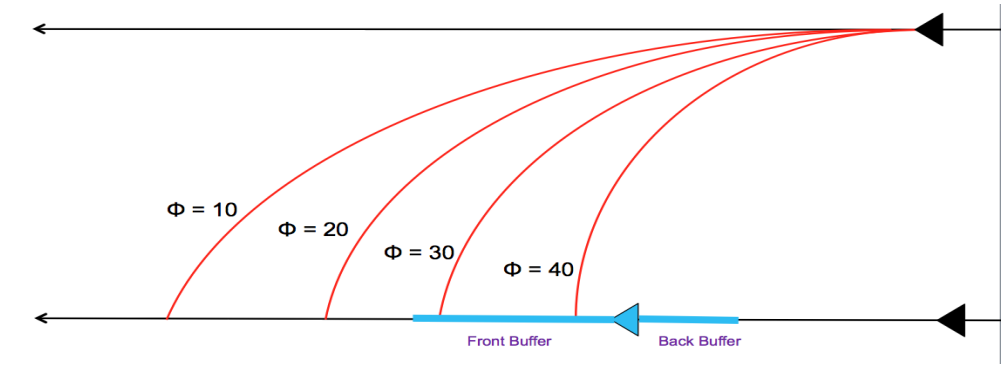


Figure 3-1: ALAS Algorithm Sweep

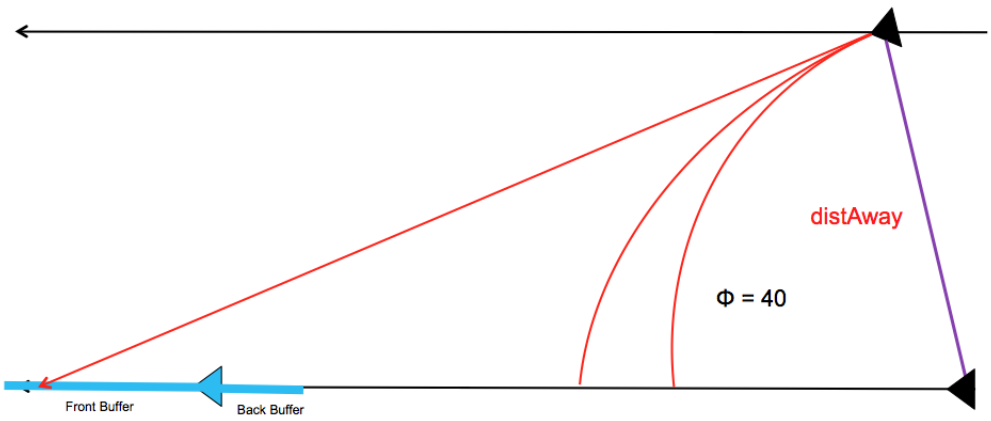


Figure 3-2: ALAS Algorithm distAway Check

The basic structure of the algorithm is as follows. The parameters of the algorithm are defined in Section 4.

```
if (Math.abs(so.z - si.z) < initHeight) {
    alertLevel_ lines = alas_lines(so, vo, si, vi,
                ln_T_red, ln_back_buffer_red, ln_front_buffer_red);
    omega = estimateOmega(traffic);
    double tau = tau(so - si,  vo,  vi)
    if (omega > trackRateThreshold && tau >= 0) {
        for (double phi = phiIncr; phi <= maxPhi; phi = phi + phiIncr) {
            double R = turnRadius(vi.groundSpeed(), phi);
            alertLevel_circle = alas_circle(so, vo, si, vi, R, ln_back_buffer_red,
                ln_front_buffer_red, ln_T_red);
        }
    }
    alertLevel_distAway = checkabsDistAway(so, si, to);
    alertLevel = max(alertLevel_ lines, alertLevel_circle, alertLevel_distAway)
}
```

The following subfunctions are used.

alas_lines	Projects the trajectory ln_T_red seconds into the future using current position and velocity vectors for the ownship (so, vo) and intruder (si, vi). Returns true if at the time that the intruder intersects the path of the ownship, it falls within the front and back buffers (ln_front_buffer_red and ln_back_buffer_red).
estimateOmega	Estimates the angular velocity (i.e. track rate) omega of the intruder based on the past numPtsTrkRateCalc data points.
tau	Calculates the time of closest approach (TCA). The function tau is negative if the trajectories are divergent.
turnRadius	Calculates the turn radius R given a ground speed and bank angle phi.
alas_circle	Projects the trajectory ln_T_red seconds into the future using a circular trajectory with radius R. Returns true if at the time that the intruder intersects the path of the ownship, it falls within the front and back buffers (ln_front_buffer_red and ln_back_buffer_red).
checkabsDistAway	Calculates the horizontal distance between the aircraft and checks if it is less than absDistRed and absDistYellow.

Note that the circular trajectory search is not performed unless omega > trackRateThreshold and the trajectories are convergent (tau > 0). This reveals that the performance of the algorithm is sensitive to the value of the trackRateThreshold parameter and the mechanism used to compute the angular velocity omega. This same technique was used in the AILS algorithm [Samanant2000]. Currently, we are using a simple averaging function to calculate omega. This provides some filtering of noise on the velocity vector. In the future, we would like to develop a filter based on real data obtained from parallel landings at several airports.

3.1.1. Mathematical Definitions of Key Components of Algorithm

In this section, the letter **s** is used to denote positions and **v** to denote velocities. The subscript

o indicates ownship and subscript **i** indicates traffic (i.e., intruder) vectors, e.g., $\mathbf{s_i}$ and $\mathbf{v_i}$. Vector variables are written in boldface and their components are referenced by subscript indices, e.g., v_x, v_y, and v_z.

$$
\begin{aligned}
&\texttt{alas\_lines}(\mathbf{s_o}, \mathbf{v_o}, \mathbf{s_i}, \mathbf{v_i}, \texttt{back\_buffer}, \texttt{front\_buffer}, T) = \\
&\quad (t_o, t_i) = \texttt{lines\_intersection}(\mathbf{s_o}, \mathbf{v_o}, \mathbf{s_i}, \mathbf{v_i}) \\
&\quad \texttt{sqd} = (t_o - t_i)^2 \, {\mathbf{v_o}}^2 \\
&\quad \texttt{IF } \det(\mathbf{v_o}, \mathbf{v_i}) = 0 \texttt{ THEN RETURN false} \\
&\quad \texttt{ELSE} \\
&\qquad \texttt{IF } t_o \leq t_i \texttt{ THEN sqd} < (\texttt{back\_buffer})^2 \\
&\qquad \texttt{ELSE sqd} < (\texttt{front\_buffer})^2 \\
&\qquad \texttt{ENDIF} \\
&\quad \texttt{ENDIF}
\end{aligned}
$$

$$
\begin{aligned}
&\texttt{lines\_intersection}(\mathbf{s_o}, \mathbf{v_o}, \mathbf{s_i}, \mathbf{v_i}, \texttt{back\_buffer}, \texttt{front\_buffer}, T) = \\
&\quad \texttt{IF } \det(\mathbf{v_o}, \mathbf{v_i}) = 0) \texttt{ THEN return } (-1, -1) \\
&\quad \texttt{ELSE} \\
&\qquad \mathbf{s} = \mathbf{s_i} - \mathbf{s_o} \\
&\qquad d = \det(\mathbf{v_o}, \mathbf{v_i}) \\
&\qquad t_o = \det(\mathbf{s}, \mathbf{v_i})/d \\
&\qquad t_i = \det(\mathbf{s}, \mathbf{v_o})/d \\
&\qquad \texttt{return } (t_o, t_i) \\
&\quad \texttt{ENDIF}
\end{aligned}
$$

$$\det(\mathbf{s}, \mathbf{v}) = \mathbf{s} \cdot \mathbf{v}^{\perp} = s_x v_y - s_y v_x$$

$$
\begin{aligned}
&\texttt{tau}(\mathbf{s}, \mathbf{v}) = \\
&\quad \texttt{IF } \mathbf{v} = \mathbf{0} \texttt{ THEN return } \infty \\
&\quad \texttt{ELSE return } -\frac{\mathbf{s} \cdot \mathbf{v}}{\mathbf{v} \cdot \mathbf{v}} \texttt{ ENDIF}
\end{aligned}
$$

$$\texttt{turnRadius}(s, \phi) = \left| \frac{s^2}{g \tan(\phi)} \right|$$

$$\texttt{alas\_circle}(\mathbf{s_o}, \mathbf{v_o}, \mathbf{s_i}, \mathbf{v_i}, \texttt{R}, \texttt{back\_buffer}, \texttt{front\_buffer}, T) =$$
$$\mathbf{C} = \texttt{centerOfTurn}(\mathbf{s_i}, \mathbf{v_i}, R, \texttt{sign}(\texttt{det}(\mathbf{s_o} - \mathbf{s_i}, \mathbf{v_i})));$$
$$t_o = \texttt{Theta\_D}(\mathbf{s_o} - \mathbf{C}, \mathbf{v_o}, 1, R)$$
$$\mathbf{s} = \mathbf{s_o} + t_o \mathbf{v_o};$$
$$t_i = \texttt{cross\_time\_intruder}(\mathbf{s_i}, \mathbf{v_i}, \mathbf{C}, R, \mathbf{s})$$
$$\texttt{sqd} = (t_o - t_i)^2 \mathbf{v_o}^2;$$
$$\texttt{IF } (\texttt{Delta}(\mathbf{s_o} - \mathbf{C}, \mathbf{vo}, R) \geq 0 \wedge t_i \geq 0 \wedge t_i \leq T \wedge$$
$$((t_o \leq t_i) \wedge \texttt{sqd} < (\texttt{back\_buffer})^2 \vee (t_i < t_o) \wedge \texttt{sqd} < (\texttt{front\_buffer})^2)$$
$$\texttt{return } t_i$$
$$\texttt{ELSE return } -1$$
$$\texttt{ENDIF}$$

$$\texttt{cross\_time\_intruder}(\mathbf{s_i}, \mathbf{v_i}, \mathbf{C}, R, \mathbf{s}) =$$
$$\epsilon = \texttt{sign}(\texttt{det}(\mathbf{s} - \mathbf{s_i}, \mathbf{v_i}))$$
$$\omega = \epsilon \frac{||\mathbf{v_i}||}{R}$$
$$\texttt{return } |(\frac{1}{\omega})\texttt{acos}(\frac{(\mathbf{s} - \mathbf{C}) \cdot (\mathbf{s_i} - \mathbf{C})}{R^2})|$$

$$\texttt{centerOfTurn}(\mathbf{s_o}, \mathbf{v_o}, R, \epsilon) =$$
$$\texttt{IF } (\epsilon > 0) \texttt{ THEN } \mathbf{v}^\perp = \texttt{PerpR}(\hat{\mathbf{v_o}})$$
$$\texttt{ELSE } \mathbf{v}^\perp = \texttt{PerpL}(\hat{\mathbf{v_o}})$$
$$\texttt{ENDIF}$$
$$\texttt{return } \mathbf{s_o} + R\mathbf{v}^\perp$$

$$\texttt{Theta\_D}(\mathbf{s}, \mathbf{v}, \epsilon, D) =$$
$$a = (\mathbf{v} \cdot \mathbf{v})$$
$$b = 2(\mathbf{s} \cdot \mathbf{v})$$
$$c = (\mathbf{s} \cdot \mathbf{s}) - D^2$$
$$\texttt{return } = \frac{-b + \epsilon\sqrt{b^2 - 4ac}}{2a}$$

$$\texttt{Delta}(\mathbf{s}, \mathbf{v}, D) = D^2\mathbf{v}^2 - \texttt{det}(\mathbf{s}, \mathbf{v})^2$$

$$\texttt{checkAbsDistAway}(\mathbf{s_o}, \mathbf{s_i}) = ||\mathbf{s_o} - \mathbf{s_i}||_{2D}$$

$$\texttt{distPerp}(\mathbf{s}, \mathbf{v}, \mathbf{q}) =$$
$$t = \frac{(\mathbf{q} - \mathbf{s}) \cdot \mathbf{v}}{\mathbf{v} \cdot \mathbf{v}}$$
$$\texttt{return } ||(\mathbf{s} + t\mathbf{v}) - \mathbf{q}||_{2D}$$

3.2. Runway Conformance Tests

In addition to the intrusion search algorithm, the Alas object provides a runway conformance test that measures the perpendicular distance from the centerline, as illustrated in Figure 3-3. This test can be performed on both the ownship and the intruder aircraft. The Boolean parameter runwayConformance is true if the test is to be applied to the ownship:

```
int ownConformance = alas.runwayConformance(true);          // test ownship
int trafConformance   = alas.runwayConformance(false);      // test intruder
```

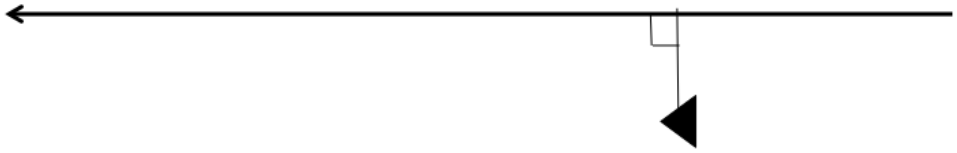


Figure 3-3: Runway Conformance Test Technique

3.3. ALAS Interface (The Programmer's API)

The interface to ALAS is simple. The user of ALAS first creates an Alas object. In Java:

```
Alas alas = new Alas();
```

In C++:

```
Alas alas = Alas();
```

Next, the aircraft id of the ownship is entered:

```
alas.setOwnship("Own");
```

The location of the runways and their orientation are entered as follows:

```
alas.setOwnRunway(37.61352, -122.35713, 13.1, 298.332);  // 28R
alas.setTrafRunway (37.61170, -122.35641, 12.7, 298.326);  // 28L
```

Then the following is called in the execution loop of the simulation:

```
alas.update("Own" , lat1, lon1, alt1, trk1, gs1, vs1, time);
alas.update("Traf1", lat2, lon2, alt2, trk2, gs2, vs2, time);
int alert = alas.alasAlert();
```

The alasAlert() function calls both the intrusion-detection sweep algorithm and the runway conformance check on the intruder aircraft.

The return value alert indicates the level of the alert:

0 = No Alert
1 = Yellow Alert
2 = Red Alert

4. ALAS Parameters

The ALAS parameters fall into three broad categories: (1) parameters that define the line-based conflict detection region, (2) parameters that control the algorithm, and (3) parameters used by the runway conformance tests.

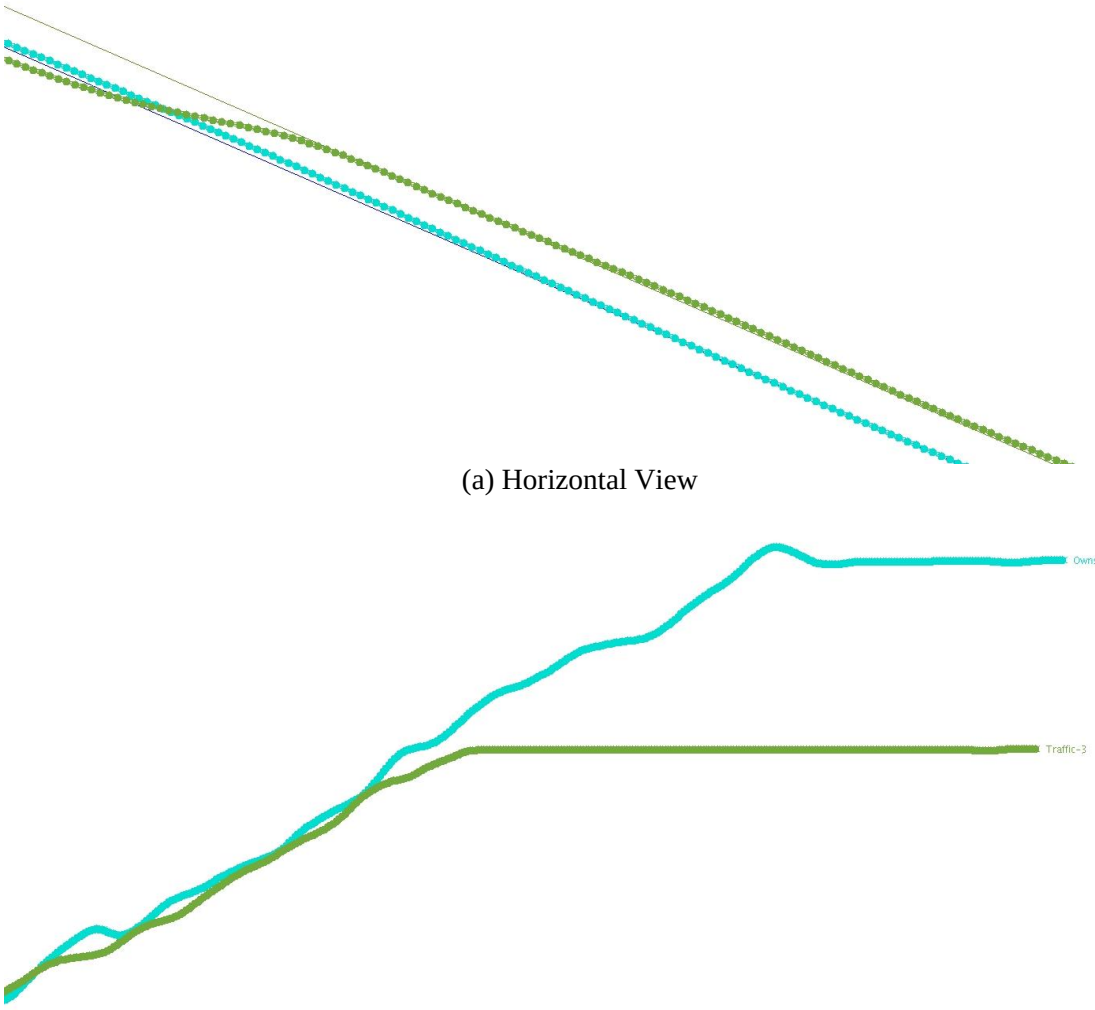
Line-based Detection Parameter	Meaning	Default Value
ln_front_buffer_red	Length of the red-alert buffer in front of aircraft	10,000 ft
ln_back_buffer_red	Length of red-alert buffer in back of aircraft	800 ft
ln_T_red	Lookahead time for red alert	15 s
ln_front_buffer_yellow	Length of yellow-alert buffer in front of aircraft	10000 ft
ln_back_buffer_yellow	Length of yellow-alert buffer in back of aircraft	1400 ft
ln_T_yellow	Lookahead time for yellow alert	35 s

Internal Parameters	Meaning	Default Value
useAbsDistAwayAlg	True if additional distance test is used	true
initHeight	Altitude difference where algorithm turns on	MAX_VALUE
numPtsTrkRateCalc	Number of data points used in track rate calculation	3
maxPhi	Highest bank angle used in search	40°
phiIncr	Bank angle increment in search	5°
trackRateThreshold	Track rate threshold that triggers the bank-angle sweeep search	1°/s
absDistRed	Mininimum horizontal distance that triggers a red alert	486.5 ft
absDistYellow	Mininimum horizontal distance that triggers a yellow alert	545.4 ft

Runway Conformance Parameter	Meaning	Default Value
redRunwayDist	Distance from centerline that triggers a red alert	170 ft
yellowRunwayDist	Distance from centerline that triggers a yellow alert	132 ft

5. Example Runs Using ALAS

The Cockpit Motion Facility (CMF) Desktop simulator was used to produce high-fidelity trajectories on SFO runways 28L and 28R. The trajectories were recorded in a comma-separated values file that contains geodesic positions and velocities for both aircraft every 0.5 s. The trace of a single run is illustrated in Figure 5-1. These trajectories were used to tune the ALAS parameters and configure the trajectory error model used in the tALAS simulator.



(a) Horizontal View

(b) Vertical View

Figure 5-1: High Fidelity Run from CMF Desktop Simulator

6. The tALAS Simulator

The test simulator for ALAS (tALAS) is based on simple kinematic models of the trajectory of the ownship and intruder aircraft. Figure 6-1 illustrates these models and identifies their key parameters.

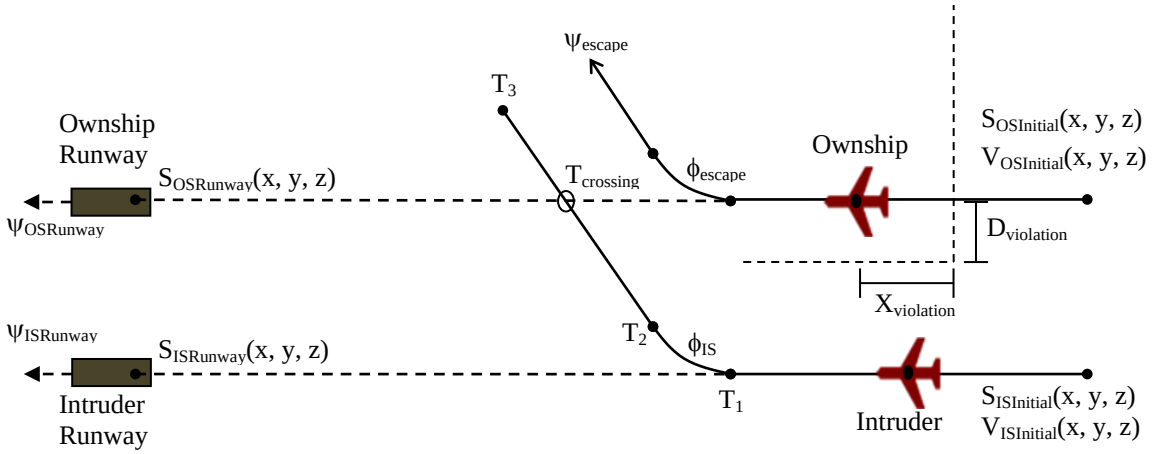


Figure 6-1: Top View of Test Scenario with Relevant Runway and Trajectory Parameters

The flight trajectory for the ownship is a straight-line descent with an optional escape maneuver. The trajectory for the intruder can be either a normal or a blundered landing approach. The trajectories are independently specified and the only interaction between the trajectories is when a red alert from ALAS triggers an escape maneuver by the ownship. The following paragraphs describe the trajectory parameters for the ownship and intruder aircraft.

A test trajectory has independent forward and lateral components. The forward trajectory component specifies the point-to-point desired path of travel. The lateral component simulates the tracking error with a simple sinusoidal oscillation model. As shown in Figure 6-2 this lateral oscillation is superposed on the forward trajectory and is specified by three parameters: peak amplitude, time period, and phase offset. The trajectories for the ownship and intruder have independently specified lateral oscillations. For the current version of tALAS, the oscillation parameters are constant for each test case. Vector addition is used to combine the forward and lateral components for a trajectory's position and velocity.

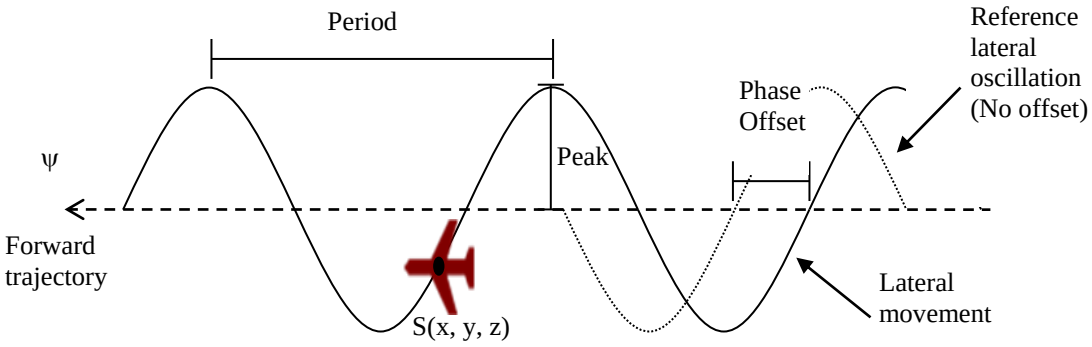


Figure 6-2: Lateral Oscillation Superposed on the Forward Trajectory

The ownship landing trajectory is specified relative to the position and heading of the runway. The position of the runway is specified by the touchdown point, denoted $\mathbf{S_{OSRunway}}$ in Figure 6-1. The runway heading is denoted $\boldsymbol{\psi}_{\mathbf{OSRunway}}$ in Figure 6-1. The landing trajectory is a straight line from the initial position $\mathbf{S_{OSInitial}}$ to the touchdown point following a specified descent angle flown

at a constant ground speed. The escape maneuver is an optional feature of the ownship flight trajectory consisting of a climbing turn away from the intruder with bank angle ϕ_{escape} and constant vertical acceleration. The turn continues until a specified heading ψ_{escape} is reached, and the vertical acceleration is sustained until the vertical speed reaches a specified value. After completing the turn and vertical acceleration, the ownship continues in a straight line. The escape maneuver is triggered by a red alert from ALAS with a specified "pilot delay" from the time of the alert to the beginning of the maneuver. Since the position and velocity components of the lateral oscillation are superposed onto the forward landing trajectory, the actual initial position and velocity are the result of the vector addition of the forward and lateral trajectory components.

The intruder trajectory is a straight-line descent with an optional blunder. In Figure 6-1, the runway touchdown point and heading are denoted $\mathbf{S}_{ISRunway}$ and $\psi_{ISRunway}$. The initial position $\mathbf{S}_{OSInitial}$ and velocity $V_{OSInitial}$ are specified to match the desired descent profile with a specified glideslope angle. The intruder ground speed remains constant throughout the descent. A blunder consists of a constant bank angle ϕ_{IS} turn from time T1 to T2, after which the intruder continues in a straight line. The turn can be to either the left or the right of the forward direction of travel. The blunder may also include leveling out to a constant altitude at or after T1.

All the parameters for the ownship and intruder trajectories are real valued, except for the binary discrete variables specifying whether the intruder will execute a normal or blundered descent, whether the intruder will level out after T1, the turn direction for a blunder, and whether the ownship is allowed to execute the escape maneuver. A run on tALAS consists of a series of test cases, each with specific parameter values. During a run, the discrete parameters are constant and each real-valued parameter is assigned evenly spaced values over a specified range starting with the minimum value and continuing with constant increments as long as the value is within the specified range. A run generates test cases until all the sweeps of all the real-valued parameters are complete or until a specified number of trials have been completed.

The simulator tALAS is instrumented to collect data on false alarms, missed alerts, and the distance of closest approach. A safety protection zone defined around the ownship is used to assess false alarms and missed alerts. Figure 6-1 illustrates the protection zone, which is defined as a moving open quadrant with the corner point X ft behind the current position of the ownship along its runway centerline, and D ft inward toward the opposite runway measured relative to the ownship's runway centerline. For each test case, tALAS determines: the time of closest approach and the corresponding distance in 3D space, as well as the horizontal and vertical distances; whether there was an intrusion into the safety violation zone; whether the intruder crossed the ownship's centerline and the corresponding time of crossing; the time of the first yellow (i.e., level 1) alert; the time of first red (i.e., level 2) alert; whether there was a loss of separation, which is determined with respect to dedicated intrusion envelopes around the aircraft; whether a given red alert was not preceded by a yellow alert; the elapsed time from the yellow alert to the red alert; the elapsed time from the red alert to the time the intruder crossed the ownship's centerline; whether there was a red alert without a blunder (i.e., a false alarm); and whether there was a violation of the protection zone without a red alert (i.e., a missed alert). For a set of test cases, tALAS can also identify the case with the overall minimum approach distance and present a complete analysis for it. The simulator tALAS generates the test trajectories as time-indexed state sequences. These sequences are processed by the instrumentation, and they can also be written to output files for post-run visualization and analysis.

7. Low-Fidelity Simulation Results

The tALAS simulator was used to evaluate the performance of the ALAS algorithm and software implementation. Each data point in the tables below was obtained from 1,889,568 simulated landings. Different trajectories for the ownship and intruder were obtained by varying the parameters listed in Table 7-1.

Table 7-1: tALAS Trajectory Parameters

Parameter	Meaning	Min Value	Max Value	Step Size
T1	Start Time of Intrusion (s)	10	20	5
T2	Duration of Intrusion Turn (s)	2	10	1
T3	Duration after turn (s)	20	20	20
bankAngle	Bank Angle of Intrusion (°)	5	30	5
Peak	Max Trajectory error (ft)	131	131	10
Period	Period of Trajectory error (s)	60	70	10
Phase	Phase of Trajectory error (°)	-180	+180	45
ownshipInitialSx	Distance from runway (NM)	5.0	5.4	0.2
intruderInitialSx	Distance from runway (NM)	5.0	5.4	0.2
ownshipInitialGs	Ground speed (KT)	160	170	10
intruderInitialGs	Ground speed (KT)	160	170	10

The horizontal profile of a blunder trajectory is illustrated in Figure 7-1.

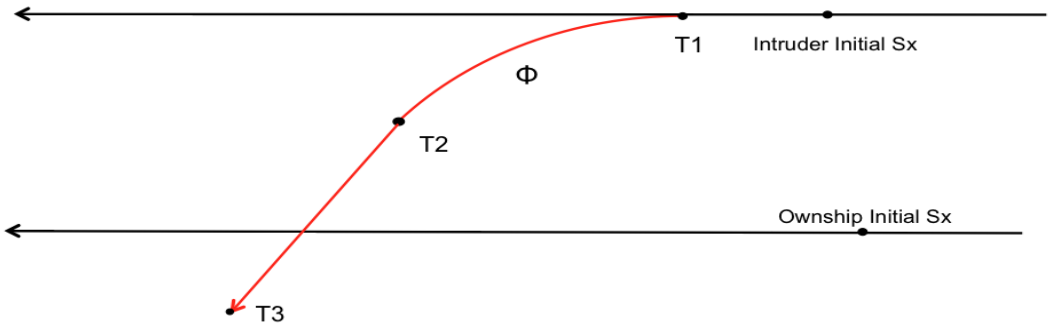


Figure 7-1: Blunder Trajectory (Horizontal View)

The program tALAS can create blunders where the intruder's altitude levels out at some point or where it continues to follow its normal vertical profile. If a vertical level-out is specified then the vertical profile is as shown in Figure 7-2.

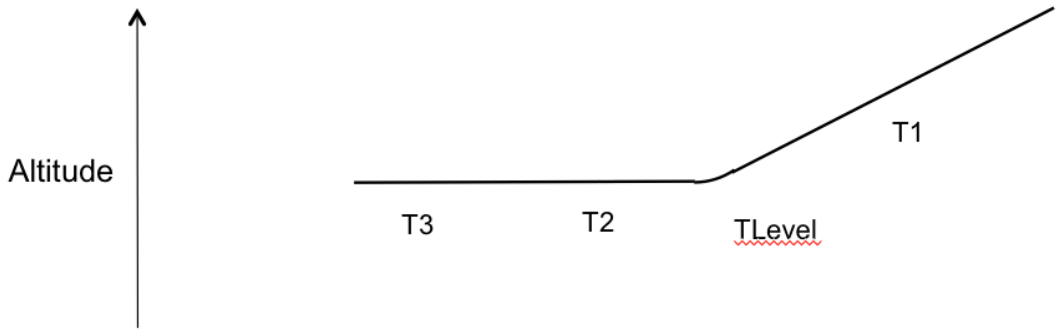


Figure 7-2: Vertical Profile for a Blunder Trajectory with a Level-out Component

The TLevel parameter, which specifies the time the level-out begins, can appear anytime after T1, the beginning of the intrusion. If a vertical level-out is not specified, then the vertical profile is as shown in Figure 7-3.

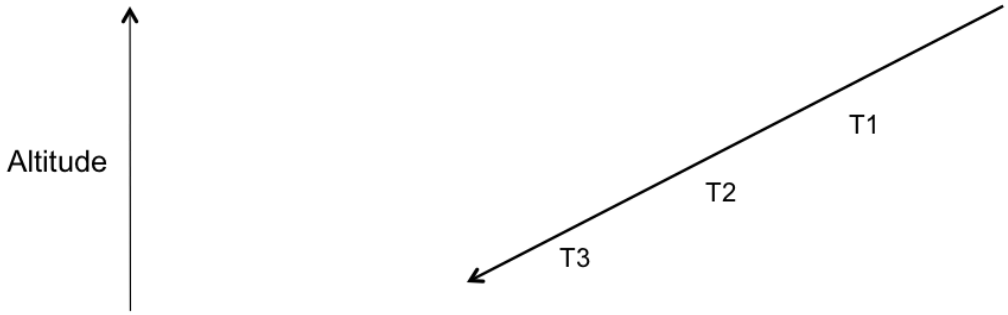


Figure 7-3: Vertical Profile for a Blunder Trajectory without a Level-out Component

The parameters in Table 7-2 characterize the escape maneuver that was used.

Table 7-2: Escape Maneuver Parameters

Parameter	Meaning	Nominal Value
escapePilotDelay	Time for pilot to react (s)	0
escapeTrack	Target track delta (°)	45
escapeBankAngle	Bank Angle of Escape Turn (°)	30
escapeGoalVs	Target vertical speed (fpm)	2000
escapeVsAccel	The vertical acceleration (m/s^2)	2.0

We will first present some of the major results obtained while tuning the ALAS algorithm. Then, we present the results for the two types of intrusions: blunder without a vertical level-out and blunder with an altitude level-out. In the millions of test cases generated by tALAS there were no cases where the ALAS algorithm failed to issue an alert.

7.1. Tuning of ALAS algorithm parameters

We present in this section the results of experimentation to evaluate the performance of ALAS as a function of algorithm parameters and procedural characteristics.

7.1.1. Performance as a Function of the Escape Pilot Delay

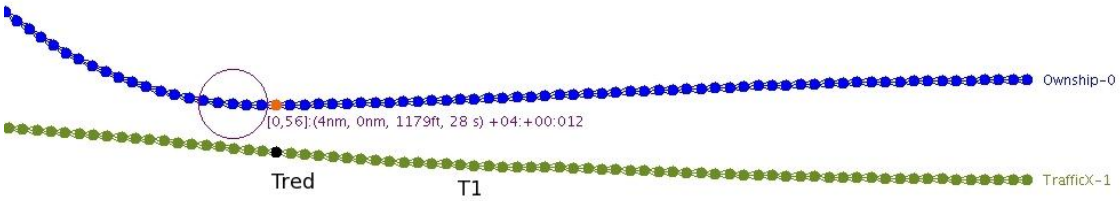
The parameter escapePilotDelay is the time between first red Alert and the initiation of the escape maneuver. It has a very significant impact on the minimum distance obtained. The results in Table 7-3 were obtained without an altitude level-out. TCA stands for Time of Closest Approach.

Table 7-3: Performance as a Function of Pilot Delay

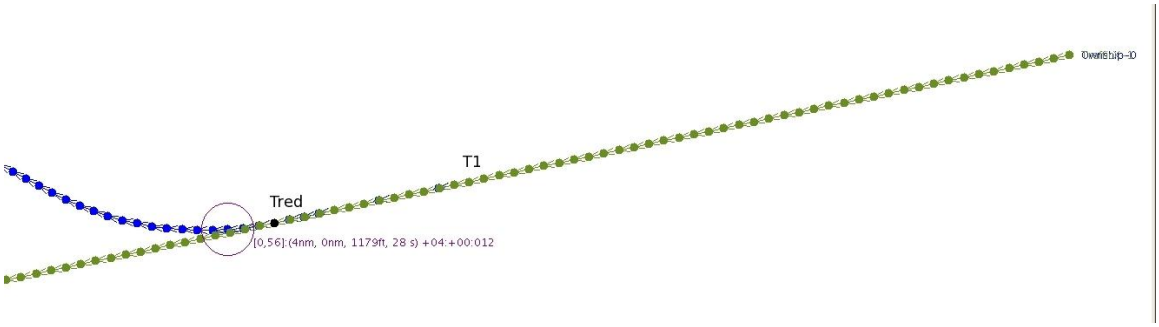
escape PilotDelay (s)	Worst-Case Minimum Distance (ft)	Horizontal Distance at TCA (ft)	Vertical Distance at TCA (ft)	T1 (s)	Time of Red Alert (s)	TCA (s)
0	448	448	7	20.0	26.5	28.0
1	379	370	82	20.0	20.5	26.5
2	260	208	155	10.0	10.5	19.0
3	163	114	116	10.0	10.5	19.0
4	122	89	83	10.0	10.5	19.0

The worst-case run for escapePilotDelay = 0 is shown in Figure 7-4. The Euclidean distance (i.e. 3D) at closest approach was 448.28 ft, which occurred at 28.0 s, or 8 s after the start of the intrusion. The horizontal distance at closest approach (indicated by a circle) was 448.22 ft and the vertical distance was 7.38 ft. The first red alert was issued at 26.5 s, or 1.5 s before the time of closest approach (TCA) and 6.5 s after the start of the intrusion.

The worst-case run for escapePilotDelay = 1 results in a minimum distance of 379.24 ft at 26.5 s. The horizontal distance at closest approach was 370.27 ft and the vertical distance was 82.02 ft.



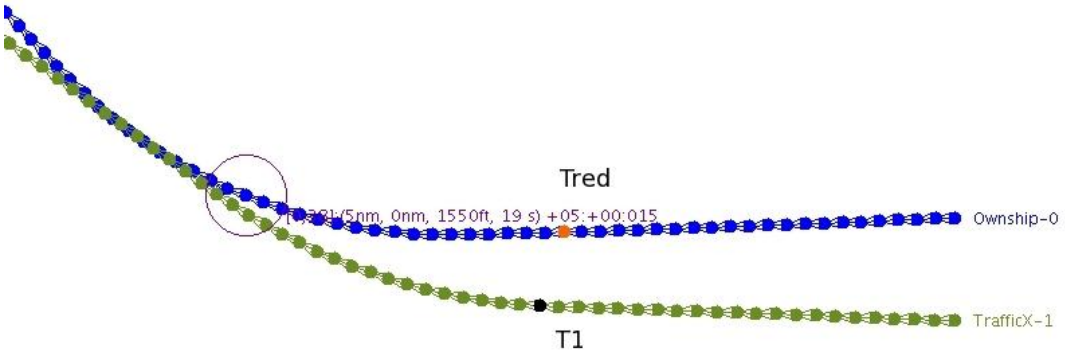
(a) Horizontal View



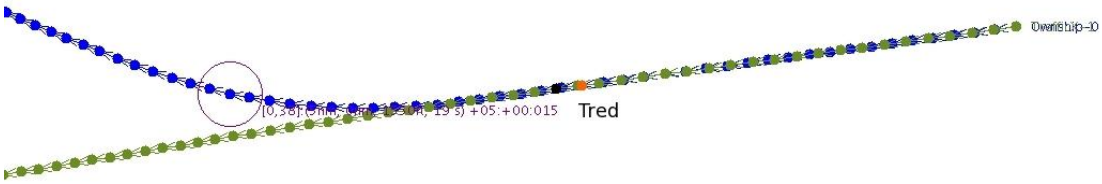
(b) Vertical View

Figure 7-4: Worst Case for escapePilotDelay = 0

The worst case run for escapePilotDelay = 3 is shown in Figure 7-5. The closest approach, indicated by a circle, is 163.15 ft at 19.0 s. The horizontal distance at TCA was 114.67 ft. The vertical distance at TCA was 116.05 ft. This example clearly shows that even with an alert that occurs 0.5 s after the start of the intrusion, a pilot delay of 3 s leads to an unacceptable minimum distance. Interestingly, as the escape pilot delay is increased, the vertical separation at TCA increases, but the horizontal distance is much closer. The circle indicates the point of closest approach and the orange dot indicates the point of the first red alert.



(a) Horizontal View



(b) Vertical View

Figure 7-5: Worst Case for escapePilotDelay = 3

7.1.2. Performance as a Function of Algorithm Parameter ln_T_red

The parameter ln_T_red is the look-ahead time for the red alert. All projected conflicts that will occur later than this parameter are ignored. If this parameter is too large, there can be nuisance alarms from normal trajectories. The data in Table 7-4 shows that this is not a problem for ln_T_red < 20 s. The % False Alarms column was calculated using only normal trajectories (i.e., without a blunder).

Table 7-4: Performance as a Function of ln_T_red

ln_T_red (s)	Worst-Case Minimum Distance (ft)	Horizontal Distance at TCA (ft)	Vertical Distance at TCA (ft)	% False Alarms
10	417	416	20	0.00
15	448	448	7	0.00
18	455	455	7	0.00
20	461	461	3	0.02
30	465	464	11	21.48

7.1.3. Performance as a Function of Algorithm Parameter absDistRed

The performance of the ALAS algorithm is strongly dependent upon the absDistRed parameter. This parameter sets the threshold for the red alert that is issued based on the horizontal distance between the aircraft. A peak value of 131 ft was used for the trajectory error. Therefore, two "normal" trajectories can get as close as 488 ft (i.e., 750 - 2*131) when they are abeam and 180° out of phase with each other. The absDistRed parameter is currently set at 486.5 ft, so normal trajectories never invoke this part of the algorithm. However, for values larger than 488 ft, an alert will sometimes occur for normal trajectories. This can be thought of as a preemptive alert. The minimum distance over all the runs can be increased by increasing this parameter. However, this comes at the cost of false alarms being issued for "normal" trajectories. The results in Table 7-5 were obtained for intrusions without an altitude level-out.

Table 7-5: Performance as a Function of absDistRed

absDistRed (ft)	Worst-Case Minimum Distance (ft)	% False Alarms
0.00 (i.e. Off)	419	0.0
300 .00	419	0.0
450.00	419	0.0
486.50	448	0.0
500.00	458	3.6
525.00	471	7.8
550.00	477	10.7

In this work, we have explored the idea of false alarms in the presence of abnormal trajectories. This is discussed in Appendix A, but we do not have any solid statistical results at this time.

We note that for absDistRed = 0, the horizontal distance check in the algorithm is effectively turned off. For that case, the time of the first red alert was 27.5 s, or 7.5 s after the beginning of a 5° bank angle intrusion. With the horizontal distance check active (using default 486.5 ft), the alert occurs at 26 s, which is 1.5 s earlier. The importance of this part of the algorithm can be seen by examining the horizontal distance for 5° bank angle intrusion as a function of time as shown in Table 7-6.

Table 7-6: Horizontal Distance as a Function of Time for a 5° Bank Angle Intrusion

Time (s)	Horizontal Distance (ft)
23.5	538
24.0	526
24.5	514
25.0	503
25.5	490
26.0	478
26.5	465
27.0	452
27.5	438

Obtaining the alert 1.5 s earlier has a large impact on the minimum distance at the point when the escape maneuver begins. This data is for a very gradual intrusion caused by a 5° bank turn. In a sharp turn, the distances drop at a much faster rate.

7.1.4. Performance as a Function of Algorithm Parameter numPtsTrkRate

The ALAS algorithm's bank-angle sweep is guarded by a function estimateOmega that estimates the track rate. If this estimate is smaller than the parameter trackRateThreshold, then this sweep is not performed. The function estimateOmega performs a simple averaging using the latest numPtsTrkRate of data. This parameter influences the detection time and hence the minimum distance as shown in Table 7-7.

Table 7-7: Performance as a Function of numPtsTrkRate

numPtsTrkRate (s)	Worst-Case Minimum Distance (ft)	Horizontal Distance at TCA (ft)	Vertical Distance at TCA (ft)	T1 (s)	Time of Red Alert (s)	TCA (s)
2	451	451	7	20.0	43.5	47.0
3	448	448	7	20.0	26.5	28.0
4	446	446	7	20.0	36.5	38.0
5	437	425	99	15.0	16.0	21.5
6	427	419	82	20.0	21.0	26.0
7	427	419	82	20.0	21.0	26.0
10	379	370	82	20.0	21.5	26.5

7.2. Blunder Trajectory Without Vertical Level-Out

The tests analyzed in this section were run with the intruder aircraft following a descending vertical profile without an altitude level-out.

7.2.1. Performance as a Function of Escape Vertical Acceleration

The minimum distance results in Table 7-8 were obtained for escapePilotDelay = 0.

Table 7-8: Performance as a Function of Escape Vertical Acceleration

Vertical Acceleration (m/s^2)	Worst-Case Minimum Distance (ft)	Vertical Distance at TCA (ft)
1.0	448	3.69
2.0	448	7.38
3.0	448	11.07
5.0	448	18.45

Surprisingly, the vertical acceleration has almost no effect. But for escapePilotDelay = 2, the following results in Table 7-9 were obtained.

Table 7-9: Performance as a Function of Escape Vertical Acceleration

Vertical Acceleration (m/s^2)	Worst-Case Minimum Distance (ft)	Horizontal Distance at TCA (ft)	Vertical Distance at TCA (ft)
1.0	199	144	137
2.0	260	208	155
3.0	293	226	186
5.0	317	292	122

The vertical acceleration has a major effect on the minimum distance when escapePilotDelay is greater than 0. When escapePilotDelay is 0, the horizontal turn is initiated soon enough to achieve adequate separation.

7.2.2. Performance as a Function of Peak Trajectory Error

As Table 7-10 shows, the worst-case minimum distance decreases as the peak value of the trajectory error increases.

Table 7-10: Performance as a Function of Peak Trajectory Error

Peak Trajectory Error (ft)	Worst-Case Minimum Distance (ft)
10	533
40	500
80	461
90	453
121	448
131	448

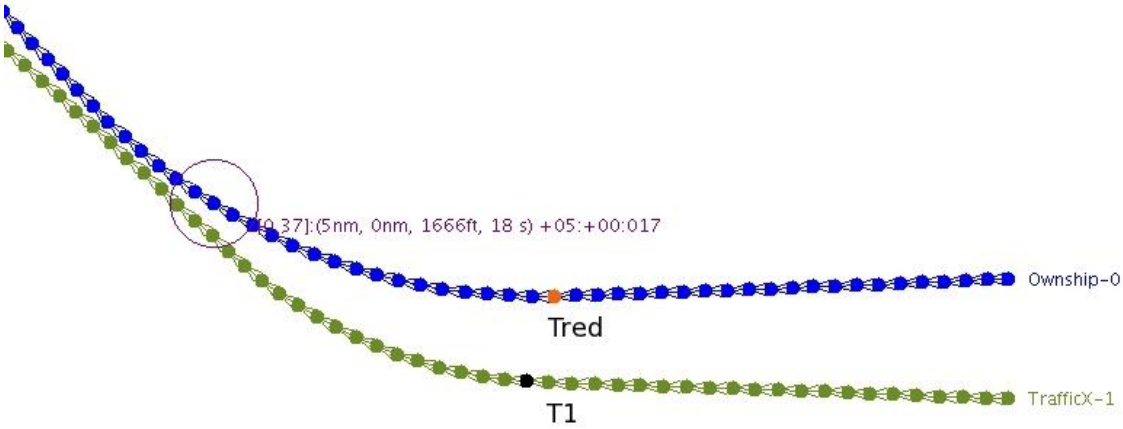
7.2.3. Performance as a Function of Maximum Bank Angle of Intrusion

The results are shown in Table 7-11. Notice that an increase in the maximum bank angle allowed on an intrusion has a significant effect.

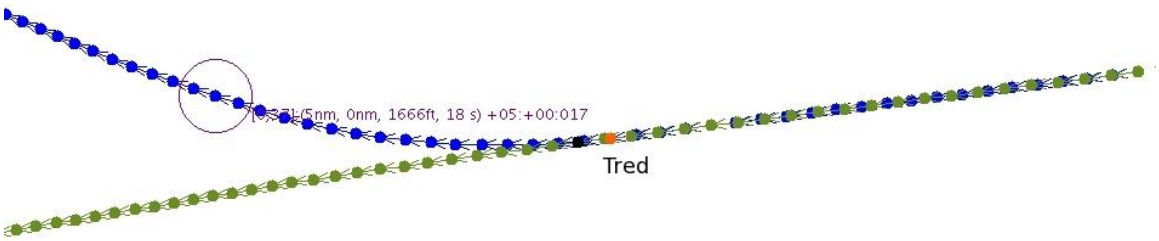
Table 7-11: Performance as a Function of Maximum Bank Angle of Intrusion

Maximum Bank Angle (°)	Worst-Case Minimum Distance (ft)
10	448.28
20	448.28
30	448.00
35	410.90
40	301.67

The worst-case run with maxBankAngle = 40° resulted in an unacceptably close encounter. This run is shown in Figure 7-6.



(a) Horizontal View



(b) Vertical View

Figure 7-6: Worst-Case Run Using High Bank Angle Intrusion

The closest approach, indicated by a circle, is 301.67 ft at 18.5 s. The horizontal distance at TCA was 201.58 ft and the vertical distance was 224.43 ft. In this case, the vertical separation was much more important than in the lower bank angle intrusions. This shows the high sensitivity of the minimum distance to the basic assumptions about the trajectory of the intrusion. Low bank angle intrusions lead to later alerts, but because the closure rate horizontally is slower, the horizontal turn of the escape maneuver is effective. If the intrusion has a high bank angle, then the horizontal closure rate is faster. Fortunately, the ALAS algorithm detects these earlier than the 30° blunder, so adequate separation is maintained. But if the blunder has a bank angle of 40° or higher, the protection zone can be penetrated.

7.3. Blunder Trajectory With Vertical Level-Out

In this section, we look at the effect of allowing the blunder trajectory to level out vertically at some arbitrary point during the intrusion. The level-out is controlled by two parameters listed in Table 7-12: TLevel and blunderVSAccel. To keep the sample size manageable, we set stepT2 = 2 s for the level-out runs.

Table 7-12: Parameters for Blunder Trajectory with Vertical Level-Out

Parameter	Meaning	Min value	Max Value	Step size
TLevel	Time of Level Out (added to T1) (s)	0 .0	15.0	1.0
blunderVsAccel	Vertical Acceleration [m/s^2]	2.0	2.0	2.0

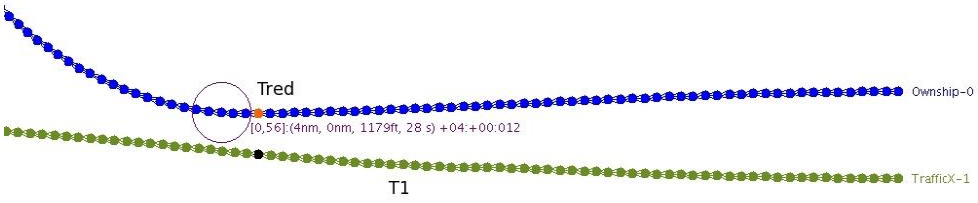
7.3.1. Performance as a Function of Escape Pilot Delay

The parameter escapePilotDelay is the time between first red alert and the initiation of the escape maneuver. The results are given in Table 7-13.

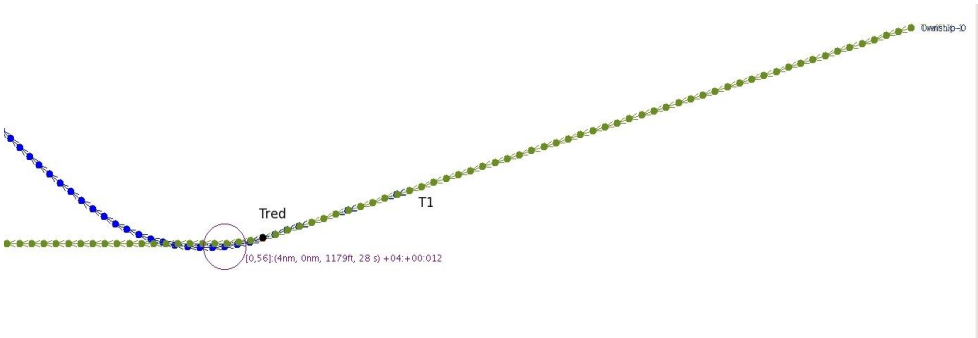
Table 7-13: Performance as a Function of Escape Pilot Delay

Escape Pilot Delay (s)	Worst-Case Minimum Distance (ft)	Horizontal Distance at TCA (ft)	Vertical Distance at TCA (ft)	Time (intrusion, alert, TCA) (s)
0	448	448	5	20.0, 26.5, 28.0
1	340	313	133	10.0, 10.5, 20.5
2	177	144	102	10.0, 10.5, 21.0
3	68	56	38	10.0, 10.5, 20.5

The worst case run with escapePilotDelay = 0 is shown in Figure 7-7.



(a) Horizontal View



(b) Vertical View

Figure 7-7: Worst-Case Run with Level-out Blunder

The TCA occurs at 28.0 s and the Euclidean distance at that time is 448.26 ft. The horizontal distance at TCA is 448.22 ft. The vertical distance at TCA is 5.74 ft. Interestingly, the level-out aspect of the intrusion has no significant effect on the distance at closest approach. This shows that the horizontal component of the escape maneuver is what is providing the needed separation. However, as the pilot delay increases, the minimum distance decreases faster in the level-out case compared with the blunder without a level-out in altitude.

7.3.2. Performance as a Function of Vertical Acceleration

The results are shown in Table 7-14. The minimum distance at TCA was not improved by increasing the escape vertical speed for the level-out blunder. The reason for this is that the horizontal turn is providing all of the separation.

Table 7-14: Performance as a Function of Vertical Acceleration

escapeVSAccel (m/s^2)	Worst-Case Minimum Distance (ft)
1.0	448
2.0	448
3.0	448
5.0	448

7.3.3. Performance as a Function of Maximum Bank Angle of Intrusion

Table 7-15 shows the effect of varying the bank angle (and hence the turn radius) of the intrusion.

Table 7-15: Performance as a Function of Maximum Bank Angle of Intrusion

MaxBank (°)	Worst-Case Minimum Distance (ft)
10	448
20	448
30	448
35	336
40	208

There is no effect up to the standard blunder bank angle (i.e. 30°), but the effect is significant for very sharp turns.

8. Performance of Runway Conformance Test

The runway performance test measures the perpendicular distance from the runway centerline. In this section, we analyze the effectiveness of this test when it is used alone.

Table 8-1: Performance as a Function of redRunwayDist

redRunwayDist (s)	Worst-Case Minimum Distance (ft)	Horizontal Distance at TCA (ft)	Vertical Distance at TCA (ft)	Times (T1, Alert, minDist) (s)
132	142	131	52	(20, 25.5, 29.5)
140	142	131	52	(20, 25.5, 29.5)
150	140	129	52	(20, 26.5, 30.5)
180	125	117	40	(20, 26.0, 29.5)
200	118	115	29	(15, 22.0, 25.0)

As Table 8-1 shows, this test alone does not provide an alert early enough for the escape maneuver to maintain adequate separation. Even for the most aggressive case, redRunwayDist = 132, the alert occurs 5.5 s after the beginning of the intrusion. Nevertheless, when used in conjunction with the ALAS triggering algorithm, it helps for very slow intrusions.

8.1. ALAS Algorithm Without Runway Conformance Test

If the ALAS algorithm is run without the runway conformance test, the minimum separation at TCA drops from 448 ft to 421 ft for escapePilotDelay = 0. The worst-case run is shown in Figure 8-1. The intrusion started at time 20 s, but the red alert was not issued until 53.5 s. The intrusion was so gradual that it did not trigger the omega threshold until 33.5 s after the slow intrusion began. The TCA is 56.5 s. When the runway conformance test is run, a red alert is issued at time 26.0 s, only 6 s after the start of the intrusion. The horizontal separation at TCA was 420 ft and the vertical separation was 16 ft.

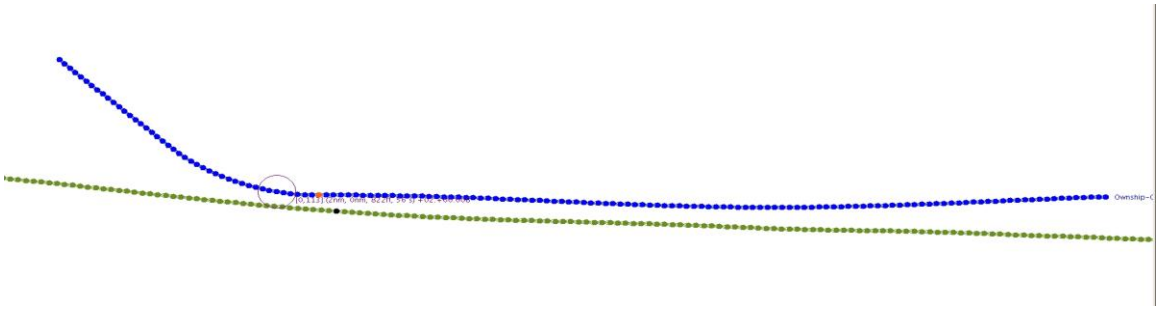


Figure 8-1: ALAS Algorithm without Runway Conformance Test (Horizontal View)

9. Performance of Yellow Alerting

In phase I of this work, we have only performed a rudimentary analysis of yellow alerting. In fact, we have still not optimized the values of the algorithm parameters associated with yellow alerting. We have obtained some statistics on the timing of the yellow alerts relative to the red alert. A histogram of the time between yellow and red alerts is shown in Figure 9-1. In this histogram, we can see that there are a high percentage of cases where there is no time between the yellow and red alert. We also see that there are a large percentage of cases with 10 or more seconds between the alerts. This suggests that there is much room for improving the yellow alerting in the future.

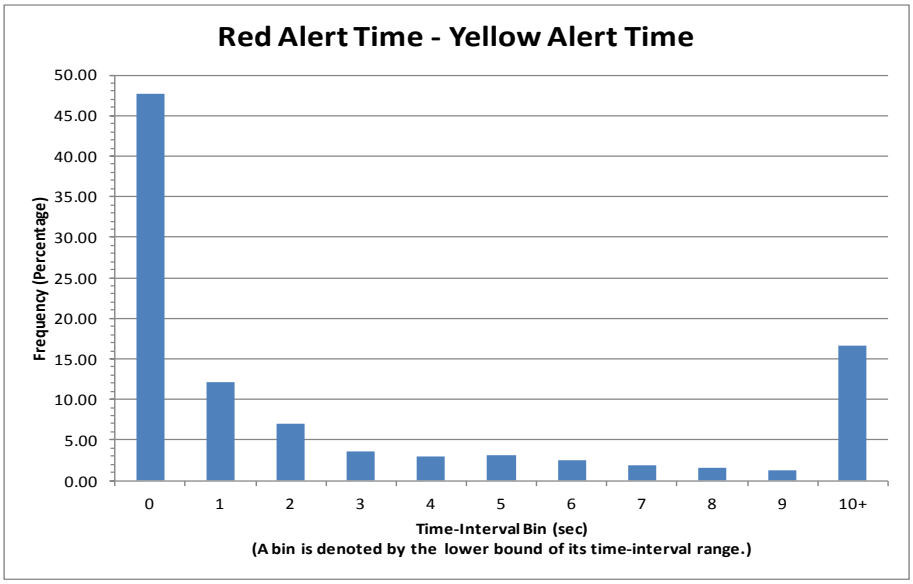


Figure 9-1: Histogram of Red Alert Times - Yellow Alert Times

10. Performance of the Tangent Fan Algorithm

A new implementation of the Tangent Fan algorithm, which is similar to the AILS algorithm [Abbott2002], has been developed. The ALAS software provides easy access to several variants of the Tangent Fan algorithm. Each of the variants uses a different conflict probe. There are three choices: a circular protection zone, an elliptical protection zone, and a linear protection zone. The worst-case minimum distances shown in Table 10-1 were measured for the case where the Tangent Fan algorithm was not augmented by a runway conformance monitor.

Table 10-1: Performance of the Tangent Fan Algorithm

Algorithm	Worst-Case Minimum Distance (ft)	Horizontal Distance at TCA (ft)	Vertical Distance at TCA (ft)
1 (circle)	136	19	135
2 (ellipse)	201	174	99
3 (lines)	96	96	1

We present these numbers with some caution because the parameters used in the algorithm have not been tuned as carefully and fully as the ALAS parameters. In addition, the elliptical conflict probe is different than the one explored in earlier AILS studies. The probe used in the ALAS object allows for a front buffer larger than the back buffer.

Table 10-2 shows that a significant improvement is seen if we add a runway conformance monitor to the Tangent Fan algorithm.

Table 10-2: Tangent Fan Performance with Runway Conformance Test

Algorithm	Worst-Case Minimum Distance (ft)	Horizontal Distance at TCA (ft)	Vertical Distance at TCA (ft)
1 (circle)	279	249	125
2 (ellipse)	201	174	99
3 (lines)	315	283	138

11. Preliminary Performance in a High-Fidelity Simulation

The ALAS algorithm with runway conformance test was exercised under limited scenarios in a high-fidelity simulation of a large transport-class aircraft that is used in Langley's CMF. This CMF simulation provides an environment that can subject ALAS to system latencies and errors and can assess the effectiveness of the escape maneuver with high-fidelity dynamics.

11.1. Modeling the SAPA Procedure

In the CMF simulation, the SAPA procedure was automated using mode control panel (MCP) settings and event-driven pilot actions. The CMF simulation does not have the capability to automate dependent operations, and each participating aircraft performed the SAPA procedure independently from initiation to touchdown. However, since the implementation is automated and dependent operation occurs under constant speed conditions, the independent operation does approximate the outcome of a dependent operation under the same condition. The CMF simulation did require an adjustment to the initial separation of the two vehicles to account for the larger true airspeed of the higher vehicle (for the given constant calibrated airspeed) that exists until the high aircraft descends along the glidepath to a near co-altitude position with the low aircraft. The transport aircraft model in CMF simulation is designed to use ILS for coupled approaches and does not have RNAV capability. This does not influence the planned path of the vehicle in the SAPA procedure, but does introduce differences in modeling total system error as described in Section 11.5.1. In the CMF simulation, the speed schedule of the procedure is primarily automated using the autothrottle speed (SPD) mode. Horizontal position is controlled using localizer (LOC) mode. Vertical position is initially automated with altitude hold (ALT

HLD) until near the glideslope where the MCP is switched to approach (APP) mode. The simulated aircraft requires pilot action to drop the landing gear, set the flaps, and deploy the speed brake (if needed). These actions were scheduled based on events with at least 2 s between consecutive actions. Deceleration of the lead aircraft was scheduled to occur at the FAF. Deceleration of the trailing aircraft was programmed to start 3 s after the lead aircraft began deceleration or upon reaching the FAF, whichever came first. The autoland feature of the autopilot performed flare and touchdown.

11.2. Modeling Blunders

The CMF simulation was modified to simulate three types of blunders: a side-step, a 30° heading change while descending, and a 30° heading change while leveling-off altitude. The latter two are consistent with FAA guidance for modeling blunders [Massimini2006]. In the side-step, the blundering aircraft switches runways, crossing to the approach path of the ownship runway. The 30° heading change was automated using heading select (HDG SEL) mode. The heading change while descending used vertical speed mode (VERT SPD) to continue the descent as dictated by the glidepath and speed schedule. The heading change while leveling-off used altitude hold (ALT HLD) mode to maintain altitude. To provide results comparable with the tALAS simulation, only results from the 30° heading change blunders (descending and level) are presented here.

11.3. Modeling the Evasive Manuever

The escape maneuver was modeled as a go-around with a heading change. Prior to initiating the go around, target speed on the mode control panel is changed to 250 KT and target altitude is set to the SFO go-around altitude of 3000 ft for runways 28L and 28R or to 1700 ft of climb whichever is higher. The 1700 ft is double the vertical collision zone for TCAS II and allows the aircraft to switch from ALAS to TCAS for collision avoidance after the SAPA pair separate vertically by more than 800 ft. The go-around is initiated by automated toggling of the go-around switch. The heading change follows and is executed as a 60° heading change away from the intruding aircraft, using heading select (HDG SEL) mode. A change of 60° elicits the most aggressive turn from the autopilot. Section 7.1.1. shows that pilot delays of 1 s or more in executing the escape maneuver can lead to intrusions of the collision zone. Therefore, the escape maneuver is auto-executed after an ALAS alert with the latency described in Section 11.5.3.

11.4. Scenarios

The ALAS algorithm was run in the CMF simulation using aircraft pairs with 7 to 8 KT differences in approach speed which are in the midrange of what the SAPA procedure can support as shown in Section 2.3. The paired approach speeds were 114 KT / 122 KT, 122 KT / 130 KT, 130 KT / 138 KT, 138 KT / 145 KT, 145 KT / 153 KT, and 153 KT /160 KT. Each step of approach speed in the above pairs corresponds to a 20,000-lb step in aircraft weight. The aircraft weight ranged from 140,000 lb to 260,000 lb to approximate the variety of aircraft sizes that may participate in the procedure. A run without blunders (not presented) was used to verify that the initial conditions would produce a successful SAPA approach and adjust those conditions as needed. Then, for each pair, four blunders were run. One pair of blunders was executed 10 NM from the threshold in the constant speed segment. A second pair of blunders was executed at 2.5 NM from the threshold in the final approach segment. In each blunder scenario, the ownship would conduct an escape maneuver when ALAS issued an alert. For each blunder, two runs were

made, an "ideal" case and a case with the "modeled avionics". The "ideal" case runs the scenario without noise, latency, or precision applied to the ALAS input data; however, the total system error in following the approach path is still modeled through induced ILS noise. The "modeled avionics" case models GPS noise, end-to-end latency, and GPS and ADS-B OUT output precision. These are applied along with latency compensation to the ownship and traffic states that are input into ALAS.

11.5. Errors and Latencies

Next, we consider the total system error, the uncertainty in the position and velocity inputs to ALAS, and the latency between the red alert and the beginning of the escape maneuver.

11.5.1. Total System Error

There are multiple avionics contributors to total system error. To adjust total system error with one control variable, each for horizontal error and vertical error, the standard deviations of the localizer and glideslope receiver noise were used. The noise in both receivers is modeled using a first-order Gauss-Markov process. The noise causes the autopilot to meander about the ILS or glideslope beam with an error on the order of the standard deviation of the noise. The standard deviation of the localizer noise was set to 0.036° which corresponds to a 2σ navigation error of 40 m (131 ft) at the distance where the ALAS algorithm activates. The standard deviation of the glideslope noise was left at its default setting of 0.035° and thus produces a near-equal error in the vertical. Because the standard deviation is expressed as a small angle, the corresponding position error decreases linearly as the aircraft approach the ILS transmitters. At 200 ft AGL (when ALAS deactivates), the corresponding position error has a 2σ value of 18 ft (5.5 m). This differs from GPS-based RNAV, which is expected to exhibit a constant error with distance from the runway threshold. However, the ILS-induced total system error, as modeled above, is adequate to model worst-case performance for the aircraft fleet in the 5 to 10+ year deployment horizon for the procedure. A side effect of injecting glideslope error for this simulated aircraft is that the meandering climbs and descents also cause deviations away from the commanded airspeed. These deviations contributed to an uncertainty in velocity difference of less than ±3 KT in the constant speed segment and of less than ±3.9 KT in the approach. This is larger than the estimated maximum uncertainty for the SAPA procedure that was computed in Section 2.3. The default noise for the glideslope is indicative of older ILS receivers and could be reduced. However, since all aircraft pairs can be positioned to successfully execute the SAPA procedure with these uncertainties, establishing more realistic vertical deviations was left to future work.

11.5.2. Uncertainty of Position and Velocity Inputs to ALAS

ALAS is expected to receive updates of the traffic aircraft state via ADS-B IN. The uncertainty of the input is a function of the uncertainty in the traffic aircraft's GPS-sensed position, the truncated precision of the GPS and ADS-B transmissions, the end-to-end latency from measurement to ALAS input, and the compensated latency. Ownship state updates are expected to be received directly from the on-board GPS unit. Both data paths begin with the on-board GPS unit and end with the avionics executing ALAS. The full data path for the ownship state is simply GPS → ALAS. The full path for the traffic data is GPS → ADS-B OUT → ADSB-IN → ALAS.

This study assumes that WAAS GPS will be the minimum equipage for aircraft participating in the SAPA procedure. RTCA DO-229D defines minimum operations standards for WAAS GPS receivers. However, the allowable position uncertainty of 32 m and update rate of 1 Hz are inadequate to support the SAPA requirement for a NACp of 10 or better [RTCA2006]. RTCA DO-242, the MASPS for ADS-B, defines, in Appendix J, the expected 1 σ position error of WAAS GPS to be 1.8 m (2σ = 3.6 m) in steady flight [RTCA2002].[2] Moreover, Garmin advertises WAAS GPS units for aviation with a 1 m RMS position error and 5 Hz update rate [Garmin2012]. For velocity, RTCA DO-260B, the MOPS for ADS-B OUT, identifies the expected 95% velocity accuracy of GPS units to be 0.2 m/s per axis (or 0.5 m/s for ground speed magnitude assuming a Rayleigh distribution) in stable flight [RTCA2009]. For GPS unit latency, RTCA DO-242 Appendix K defines the expected latency from measurement to data transmission to be 0.3 s for high NACp applications [RTCA2002]. This latency is divided into 0.1 s for receiver operation and 0.2 s for state data transmission to destination. GPS units transmit the UTC time of applicability for the state data, which enables compensation of latency by the receiving unit; the time of applicability coincides with the time mark pulse emitted by the GPS unit, which is normally also the start of transmission. RTCA DO-260B implies that in high NACp applications, the ADS-B avionics must be synchronized on the time mark of the GPS unit and the GPS unit must be configured to emit time marks at a UTC sub-epoch (a 0.2 UTC sub-second) [RTCA2009]. Based on the information above, the GPS units were modeled with a 3 m position noise, a 0.5 m/s ground-speed noise, a 5 Hz update rate on the UTC sub-epoch, and a 300-ms latency from truth state through transmission. In addition, the data transmitted has numerical precision as defined in RTCA DO-229D Appendix H: 8.38E-8° for latitude and longitude, 0.125 ft for altitude, 0.125 KT for ground speed, and 0.0055° for true track [RTCA2006].

For ADS-B OUT, the FAA rule permits a 2.0 s latency from state measurement to ADS-B OUT transmission; 0.6 s of the latency can be uncompensated [FAA2010]. These minimum requirements are compatible with the FAA minimum NACp of 8 (92.6 m) but not the SAPA minimum NACp of 10 (10 m). RTCA DO-242 Appendix K, again, provides guidance for the end-to-end latency of ADS-B OUT in high NACp applications. The latency contribution from the GPS unit is as defined in the previous paragraph. The latency expected for the ADS-B avionics from reception of GPS data to transmission of the ADS-B report is 0.4 s [RTCA2002]. ADS-B units synchronized to the GPS time mark, as defined in DO-260B, should be able to apply compensation for the full latency and can assign a UTC sub-epoch as the time of applicability for the ADS-B OUT report. This will allow the receiving aircraft to further compensate for latencies downstream. ADS-B OUT reports also exhibit the precision defined in RTCA DO-260B. The encoding algorithm for position maintains an approximately 5 m precision in the North and East axes. Velocity precision is 1 KT in the North and East axes [RTCA2009].

Latency for ADS-B IN was taken from RTCA DO-317A, the Minimum Operational Performance Standards for Aircraft Surveillance Applications. This RTCA document allocates 0.5 s to ADS-B IN latency [RTCA2011]. To define the latency for the ALAS avionics, it was assumed that the ALAS avionics would run at 5 Hz but would operate asynchronously to both the ownship GPS unit and the ADS-B IN avionics. Therefore, ALAS can introduce 200 ms of additional latency to the ownship and traffic data.

---

[2] In the SAPA procedures, ownship and traffic accelerations are expect to remain below 0.1 g prior to an alert; therefore, the SAPA scenarios qualify as steady flight.

Because the state data is closely associated with a UTC sub-epoch under both paths, nearly all of the latency can be compensated. However, there are some allowances to decrease the compensated latency. RTCA DO-260B allows up to a 5-ms deviation in the GPS time mark from the actual UTC sub-epoch [RTCA2009]. This deviation plus an additional 1 ms of uncaptured time delays is subtracted from the compensated latency for paths within each aircraft. The 5-ms GPS time mark deviation is also used to establish a worst-case clock difference between aircraft of 10 ms to which an additional 1 ms is added for other uncaptured time delays. This 11 ms uncompensated latency is assigned to the ADS-B IN processing of the traffic data. The resulting end-to-end latency of aircraft equipped for the SAPA procedure is summarized below. In this summary, the latency of the traffic data is divided between ADS-B OUT and ADS-B IN because of the change in precision of the state data in the ADS-B OUT report.

- Ownship State
  - Total Latency (Measurement to ALAS input): 500 ms
  - Compensated Latency: 494 ms
- Traffic State
  - ADS-B OUT
    - Total Latency (Measurement to Transmission): 700 ms
    - Compensated Latency: 694 ms
  - ADS-B IN
    - Total Latency (Reception to ALAS): 700 ms
    - Compensated Latency: 689 ms

One side effect of relying on ADS-B synchronized to the GPS time mark is that the 2-per-second transmission of state data cannot occur in equal intervals. The transmission is now associated with a 0.2 s sub-epoch. Therefore, the interval alternates between 0.4 and 0.6 s. The ALAS algorithm has not been verified and validated for variable intervals. Therefore, the CMF simulation calls ALAS at a constant 0.5 s rate and increases total and compensated latency by 100 ms for both the ownship data and the ADS-B in path of the traffic data.

11.5.3. Latency of Evasive Maneuver

The latency between alert and command of the escape maneuver was set at 260 ms. Two hundred ms was the assumed worst-case compute time of the ALAS avionics. The autopilot was assumed to run at a rate of 50 Hz. A one-cycle delay in receiving the alert and a two-cycle compute delay in issuing the escape maneuver command were assumed for the autopilot.

11.6. Increasing Bank Rate of Automated Manuevers

Initial runs showed that turns using the autopilot are slow to develop because the autopilot takes more than 15 s to bank the aircraft to 30°. This is almost three times as long as what is achievable in manual flight. Therefore, an option was added to augment the roll rate using roll stick commands. A simple feedback loop moves the roll control based on three criteria. At the start of the turn, the roll control chases a 10°/s roll rate. When the aircraft nears a bank angle of 30°, the roll control is moved toward a normalized value of 1/3, which maintains a bank of ~30°. As the aircraft nears the desired heading, the roll control is returned to a normalized value of zero and augmentation of the autopilot control ceases. The feedback loop was tuned to ensure that the blundering aircraft would not intercept the ownship's approach path by more than 30°. The tuning effectively capped the length of time that the blundering aircraft could maintain a high turn

rate corresponding to bank angles between 25° and 30°. The ownship, targeting a heading change of 60°, could sustain a high turn rate longer. However, this advantage had little influence on minimum separation after the alert because the minimum occurs within the first 13.5 s following the alert. When activated, this roll augmentation was applied to both the blunder and the escape maneuver since participating aircraft turning at the same rate should represent a worst case. Results with and without this roll augmentation are presented in Table 11-1 through Table 11-4.

11.7. Measuring Time to Alert

One aspect of ALAS performance is how quickly it alerts the ownship of the blundering traffic. Of particular interest is the effect of avionics latencies and error in the time to alert. In this paper, the time to alert is defined as the time between vehicle departure from the approach path and the issuing of the alert. ALAS uses rate of turn as the metric to identify suspected departures from path. However, aircraft normally perform small turns on approach due to navigation and flight technical error. In the CMF simulation, the aircraft exhibit a maximum turn rate of 0.36°/s under blunder-free conditions. Therefore, the time when the turn rate first exceeds 0.5°/s or greater is the reference used to count time-to-alert.

11.8. Results

The outcome of the high-fidelity CMF simulation runs are shown in Table 11-1 through Table 11-4. Each table shows the time to alert, the magnitude of the 3D distance vector between aircraft at the time of the alert, and the 3D distance at closest separation. This data set enables analysis of blunder type, rate of turn, and avionics performance on collision avoidance.

11.8.1. Blunder Type

Whether the aircraft descends or levels out while blundering does not make a significant difference to alert times. Of the 48 case-pairs that differ by blunder type, only one-third (16) differ by greater than the 0.5 s time step for the ALAS algorithm; the median difference is 0.02 s. However, where the case-pairs do differ, the alert times for level-out blunders are lower (14 out of 16). Though blunder type has little influence on alert time, this is not true for efficacy of the escape maneuver. The closest approach is lower for the level-out blunder in 46 of the 48 case-pairs and the median decrease is 125 ft. The level-out blunder eliminates one advantage of the escape maneuver, the climb. In the level-out blunder, the trailing aircraft continues to descend on the glidepath while the blundering aircraft remains level. By the time an alert is issued, the trailing aircraft is often near or below the altitude of the blundering aircraft. For the level-out blunder, the median vertical separation at the alert is -8 ft; for the descending blunder the median is +79 ft. Furthermore, for the descending blunder, the blundering aircraft continues to descend during the escape maneuver. That leads to a greater vertical separation at closest approach. For the descending blunder, the median vertical separation at closest approach is 293 ft; for the level-out blunder, the median is 63 ft.

11.9. Rate of Turn

Rate of Turn does not significantly influence time to alert. Of the 48 case-pairs, only 18 show a difference greater than the 0.5 s time step of the algorithm. Nevertheless, all 18 are lower for the augmented turn and the median difference over all 48 cases is +0.2 s (normal turn higher). Increased rate of turn also does not offer consistent improvement in separation. Of the 48 case

pairs, the increased rate of turn produces improved separation greater than 33 ft (10 m) in 23 cases but also produces reduced separation greater than -33 ft in 12 cases. Overall, little more than half (27) of the cases show any improvement in separation from increased turn. The median difference between the augmented and normal turns is +30 ft. The primary reason that rate of turn has little influence on separation is that the blundering aircraft and ownship use the same turn performance, separated only by time. Nevertheless, significant trends do emerge when the case pairs are subdivided between constant speed and final approach segments. In the constant speed segment, there emerges a tilt toward better separation under the normal turn; however, the advantage is not significant. The normal turn produces greater separation in 16 of the 24 cases but the median improvement is only 18 ft. On the other hand, the augmented turn produces significantly higher separations against blunders in the final approach segment. Separation is improved in 19 of 24 cases and the median improvement is 60 ft. What favors the augmented turn during final approach is the difference in speed between the two aircraft. The faster turn rate applies more of the trailing aircraft's faster groundspeed toward cross-track separation more quickly; in fact, the median increase in cross-track separation over the normal turn at minimum separation is 190 ft.

11.9.1. Modeled Avionics

The end-to-end latency of the avionics, though compensated, has a direct effect on time to alert. When avionics are modeled, the time-to-alert grows by a median value of 1.5 s, the same as the end-to-end latency of the traffic data. Avionics noise and precision almost evenly tilts half (23 out of 48) of the case-pairs toward a 1 s (13 cases) or 2 s (10 cases) increase in time-to-alert. This delay in time-to-alert directly leads to a decrease in minimum separation during the escape maneuver.[3] The median loss of separation is 113 ft, almost a half-wingspan for a Boeing 747-8.

11.10. Indicated Collisions

A collision is indicated when the minimum separation between the participating aircraft falls below 400 ft. Within the 96 runs presented in Table 11-1 through Table 11-4 there are nine collision indications. However, these 96 runs are not all independent; they are variations of 12 uncorrelated initial conditions. Of these twelve, a collision indication appears for three. All but one of the collisions occur with avionics modeled. The one collision in the ideal case occurs for a normal turn, which may not be indicative of the escape performance of the aircraft; the augmented turn for the same case produces a separation well outside 400-foot collision zone. All of the collision indications occur when 3D separation at alert is less than 880 ft. But not all alerts at less than 880 ft separation lead to a collision; there are 17 of these runs without a collision indication. Nevertheless, the data suggests that an unsafe probability of collision may exist at separations of less than 900 ft.

[3] The avionics latency and noise modeled here are only applied to ALAS inputs and are not applied to the autopilot or other systems that drive vehicle performance. Therefore, changes in seperation between the ideal and modeled avionics cases can be attributed solely to delays in ALAS alerts.

Table 11-1: High-Fidelity Simulation Results: 30° Blunder While Descending During Constant Speed Segment

	Ideal Input to ALAS					
	Normal Turn			Augmented Turn		
Approach Speeds of Fast/Slow (KT)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)
122 / 114	1.52	855	587	1.34	862	613
130 / 122	1.74	1016	833	1.12	1019	861
138 / 130	2.16	915	773	2.14	889	700
145 / 138	1.34	1133	988	1.66	1123	908
153 / 145	1.16	1214	1072	1.50	1203	1039
160 / 153	2.56	967	704	1.88	938	736

	Modeled Avionics Input to ALAS					
	Normal Turn			Augmented Turn		
Approach Speeds of Fast/Slow (KT)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)
122 / 114	3.02	836	500	2.84	841	448
130 / 122	3.24	1000	732	2.62	1007	730
138 / 130	3.66	907	678	3.14	878	615
145 / 138	3.34	1120	853	2.66	1114	828
153 / 145	3.16	1205	962	3.00	1190	907
160 / 153	3.56	963	677	3.38	921	552

Table 11-2: High-Fidelity Simulation Results: 30° Blunder While Level During Constant Speed Segment

	Ideal Input to ALAS					
	Normal Turn			Augmented Turn		
Approach Speeds of Fast/Slow (KT)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)
122 / 114	1.50	853	410	1.36	859	539
130 / 122	1.70	996	661	1.10	1010	749
138 / 130	1.62	875	441	1.10	866	620
145 / 138	1.30	1108	804	1.14	1107	787
153 / 145	1.58	1186	923	0.94	1187	904
160 / 153	2.46	940	515	1.30	919	624

	Modeled Avionics Input to ALAS					
	Normal Turn			Augmented Turn		
Approach Speeds of Fast/Slow (KT)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)
122 / 114	3.00	838	353	2.86	835	349
130 / 122	2.70	985	634	2.60	991	601
138 / 130	2.62	868	383	2.60	852	441
145 / 138	2.80	1099	717	2.64	1092	661
153 / 145	3.08	1176	810	1.94	1180	807
160 / 153	3.96	933	456	2.80	904	418

Table 11-3: High-Fidelity Simulation Results: 30° Blunder While Descending During Final Approach Segment

	Ideal Input to ALAS					
	Normal Turn			Augmented Turn		
Approach Speeds of Fast/Slow (KT)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)
122 / 114	1.20	1369	1056	1.00	1375	1157
130 / 122	1.40	1587	1277	1.14	1609	1319
138 / 130	1.24	1592	1289	1.14	1613	1427
145 / 138	1.56	814	500	1.36	828	558
153 / 145	1.56	874	431	1.44	903	491
160 / 153	1.26	1405	1074	1.22	1440	1074

	Modeled Avionics Input to ALAS					
	Normal Turn			Augmented Turn		
Approach Speeds of Fast/Slow (KT)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)
122 / 114	3.20	1345	883	3.00	1357	1013
130 / 122	2.90	1563	1125	2.64	1589	1213
138 / 130	3.24	1564	1115	2.64	1591	1332
145 / 138	3.06	792	422	2.36	812	463
153 / 145	3.56	830	267	2.44	880	327
160 / 153	2.26	1386	988	2.72	1408	958

Table 11-4: High-Fidelity Simulation Results: 30° Blunder While Level During Final Approach Segment

	Ideal Input to ALAS					
	Normal Turn			Augmented Turn		
Approach Speeds of Fast/Slow (KT)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)
122 / 114	1.38	1355	906	1.00	1364	970
130 / 122	0.94	1581	1169	1.12	1604	1098
138 / 130	1.24	1580	1180	1.16	1612	1358
145 / 138	1.60	822	454	0.90	835	582
153 / 145	1.62	884	241	0.98	919	517
160 / 153	1.34	1411	952	1.28	1449	1033

	Modeled Avionics Input to ALAS					
	Normal Turn			Augmented Turn		
Approach Speeds of Fast/Slow (KT)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)
122 / 114	2.38	1344	840	2.50	1351	806
130 / 122	2.94	1544	1023	2.62	1578	986
138 / 130	3.24	1545	956	2.16	1594	1294
145 / 138	2.60	810	394	2.40	816	462
153 / 145	2.62	866	177	2.98	871	235
160 / 153	3.34	1369	854	2.78	1413	879

11.11. Worst Case

In the above discussion, three modeling parameters are shown to have the greatest influence on minimum separation after alert. They are a level-turn blunder, avionics latency, and separation at the alert below 900 ft. These aspects were combined in two additional runs, for the constant speed and final approach segments, to elicit the worst-case performance of the SAPA procedure. For the constant speed segment, starting separation was adjusted to produce an along-track separation of 130 ft at the time of alert. For the final approach segment, an along track separation of 260 ft was targeted for the alert. The increased separation for the final approach segment is intended to counter the faster speed of the trailing aircraft. In both cases, adjustment of starting separation used the augmented turn as a reference. As a result, the along-track separation at alert for the normal turns are smaller, 130 ft and 260 ft, respectively. The results are presented in Table 11-5 and Table 11-6. Only one case in the table stays outside of the collision zone. Moreover, the minimum separation in half of the cases is less than the wingspan of a Boeing 747-8. Such a deficit is unlikely to be overcome by adjustments to the algorithm or the SAPA procedure. In fact, the tALAS simulation results point to an inherent obstacle to overcoming this deficit.

As discussed in Section 7.1.1. the tALAS simulation results show that collisions cannot be prevented if pilot initiation of the escape maneuver is delayed by 1 s or more after the alert. The

tALAS simulator uses true state data for inputs and, therefore, is equivalent to the ideal cases of the CMF simulation. The alert delay in the modeled avionics cases becomes comparable to a delay in pilot action. This delay has a median of 1.5 s but is also followed by a 260-ms delay for the autopilot to initiate the maneuver. Therefore, the worst-case results of the high-fidelity simulation should approach the worst-case result from tALAS with a 2-second delay. Section 7.3.1. shows the results of pilot delay for the level-out blunder. The program tALAS estimates a worst case separation of 177 ft for a 2 s delay. In the CMF simulation, the augmented turn comes closest to matching the turn performance that tALAS uses for the ownship and traffic. Under the augmented turn, the minimum separation is as low as 146 ft, which is near the 177 ft result from tALAS. Therefore, a collision-free procedure is not possible under the latency introduced by real-world avionics. The total latency of the traffic state would need to be reduced below 1 s for a collision-free procedure at 750-ft runway separation. This is unlikely to be accomplished using ADS-B as currently defined.

Table 11-5: Small Along-Track Separation at Alert: 30° Blunder While Level During Constant-Speed Segment

	Modeled Avionics Input to ALAS					
	Standard Turn			Augmented Turn		
Approach Speeds of Fast/Slow (KT)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)
122 / 114	2.78	718	51	2.68	700	216
130 / 122	2.56	737	58	2.54	717	241
138 / 130	2.46	754	114	2.44	731	294
145 / 138	2.34	752	119	2.72	722	146
153 / 145	2.90	767	118	2.78	737	174
160 / 153	4.26	802	115	3.06	761	181

Table 11-6: Small Along-Track Separation at Alert: 30° Blunder While Level During Final-Approach Segment

	Modeled Avionics Input to ALAS					
	Standard Turn			Augmented Turn		
Approach Speeds of Fast/Slow (KT)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)	Time to Alert (s)	3D Distance at Alert (ft)	Closest 3D Distance (ft)
122 / 114	2.36	852	348	2.58	853	418
130 / 122	2.66	828	302	2.36	850	354
138 / 130	3.28	776	114	2.60	818	374
145 / 138	3.16	833	167	2.52	856	342
153 / 145	3.22	811	88	2.58	841	274
160 / 153	3.02	755	65	2.44	800	245

12. Comparing the Low Fidelity and High Fidelity Results

The conclusions from the low-fidelity simulations and the high fidelity simulations are different. The low fidelity simulations show that with a zero pilot delay, a minimum separation distance greater than 400 ft is achievable. The high-fidelity simulations found several cases where the separation was significantly less than 400 ft. A comparison of the results show that the difference is caused by the escape maneuver used in these simulations. The low-fidelity simulation assumed that a 30° turn immediately begins with a 3.94°/s turn rate. In the high-fidelity simulation, the automated system produced a very sluggish turn. In fact, it took 5.5 s just to reach a turn rate of 1°/s. A typical turn is shown in Figure 12-1.

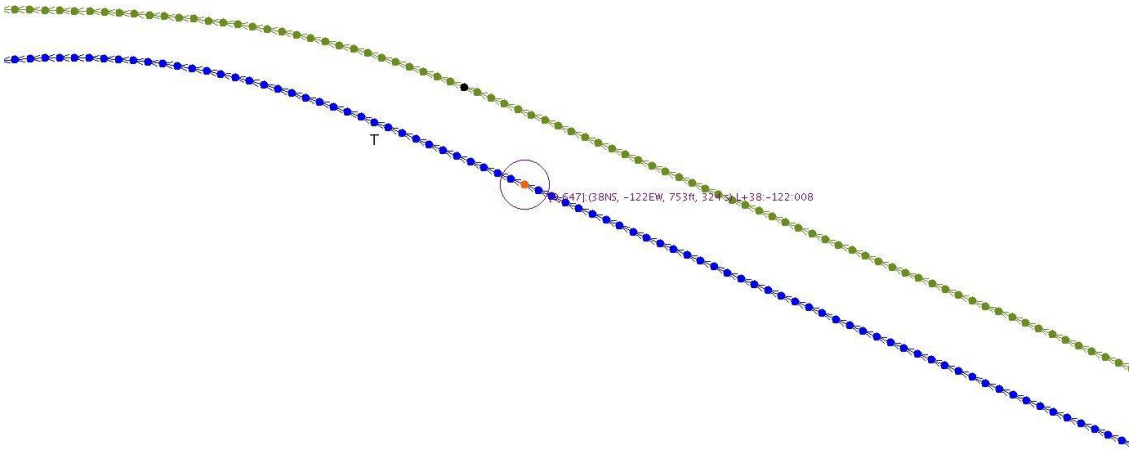


Figure 12-1: Horizontal Perspective for Typical Large Class Aircraft Turn

The orange dot indicates the point where the escape maneuver was initiated (e.g. time 323.5 s). But the rate of change of the aircraft track does not reach 1°/s until 5.5 s later at time 329 s. This is indicated by the T on the diagram. The vertical perspective is shown in Figure 12-2.

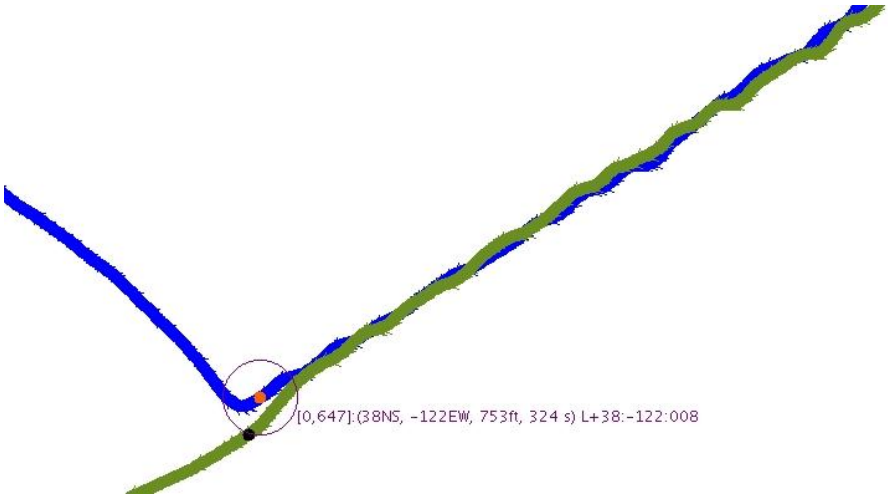


Figure 12-2: Vertical Perspective For Typical Large Aircraft

By the time the aircraft has reached a robust turn rate (e.g. 2.8°/s), 12 s has elapsed and the intruder aircraft has covered most of the 750-ft separation. This sluggish turn rate is likely the result of tuning for passenger comfort; transport autopilots are not designed with the expectation that they will be used to evade potential collisions. When the high-fidelity simulation models avionics, communication latencies add between 1 and 1.5 s to the time to alert over what was seen in the low-fidelity simulations.

Late in the study, some additional high-fidelity runs were performed using automated movement of the pilot roll control in addition to the autopilot to increase the rate of turn; these are referred to previously as the "augmented turn." The augmented escape maneuver reduced the time to reach 1°/s to about 3 s, but this is still not fast enough to have a guarantee that there is no violation of the protection zone for the worst possible scenarios.

It appears that the fully general SAPA procedure will not be possible unless an escape maneuver can be created that initiates an aggressive turn almost immediately after the alert is received. We believe that the false alarm rate can be reduced with further improvements and that the yellow alerting can be improved, but in our simulations, the ALAS algorithm detected the intrusions as quickly as could be expected.

We have performed some preliminary studies of the 1000-ft parallel runway case and the results are more promising. A pilot delay of 2 s or less resulted in no near collisions. These results were obtained without tuning the ALAS parameters for the 1000-ft separation. Table 12-1 provides the results of these simulations.

Table 12-1: Low Fidelity Kinematic Simulation of 1000 ft Parallel Runway

escape PilotDelay (s)	Worst-Case Minimum Distance (ft)	Horizontal Distance at TCA (ft)	Vertical Distance at TCA (ft)	Start of Blunder T1 (s)	Time of Red Alert (s)	TCA (s)
0	649	649	13	15.0	24.0	26.0
1	607	581	177	10.0	10.5	18.5
2	454	393	226	10.0	10.5	20.5
3	306	230	202	10.0	10.5	21.0
4	185	99	157	10.0	10.5	21.0

The blunder was an immediate 30° bank turn with no level out. The worst-case run for escapePilotDelay = 0 shows the Euclidean distance (i.e. 3D) at closest approach was 649 ft, which occurred 11 s after the start of the intrusion. The horizontal distance at closest approach was 649 ft and the vertical distance was 13 ft. Once again the need for a very rapid escape maneuver is seen in the results. A pilot delay of 3 s resulted in a near miss while a pilot delay of 2 s was acceptable. The extra 250 ft enables an extra 2 s to respond.

The 750 ft parallel spacing problem was especially challenging because the tracking error of a normal aircraft could be as large as 131 ft. With this maximum tracking error for both aircraft, the paired aircraft can get as close as 488 ft before the intrusion begins. Repeatedly, the fast-time kinematic simulator would find that the worst case involved trajectories where the aircraft get close to the tracking limits and then the intruder turns abruptly into the ownship.

13. Future Work

There are several directions that can be pursued to improve the ALAS algorithm and the safety analysis for the SAPA procedure.

13.1. ALAS Trigger Function

The ALAS algorithm uses a sweep of trajectories with different bank angles for the intruder aircraft and checks to see if any of these cross the ownship's runway centerline too close to the ownship. This sweep is guarded by a simple trigger that filters noise on the trajectories. The current design relies on a simple averaging function. If real data can be obtained for aircraft approaches on parallel runways, then a filter function tailored to this domain can be designed. This potentially could improve both the minimum distance and the false alarm rate.

13.2. Kinematic Analysis in the Presence of ADS-B Latency and Position Errors

Although we performed some simulations in the CMF laboratory that included ADS-B latency and position errors, this occurred late in the study and we were not able to introduce these errors into our tALAS low-fidelity simulation. The effect of these latencies on the minimum distances could be significant. Future work could improve the low-fidelity simulation to include ADS-B latencies and position errors.

13.3. A Kinematic Study Using Double-Turn Blunder Model

Appendix B presents our initial simulation results for the double-turn blunder model. The double-turn blunder model introduces several new degrees of freedom into the simulation and hence the number of test cases that are needed for even a coarse exploration of the input space leads to very large test times (e.g. 4+ days). In phase II, we can explore this state space more fully. We suspect that a more complex escape maneuver will be needed if the double-turn model is selected as the "standard blunder model" for use in the safety analysis. However, the more fundamental question that must be answered is what is the appropriate blunder model?

13.4. A Better False Alarm Analysis

To accurately forecast a false-alarm rate, one must have an accurate population distribution of blunder trajectories. Unfortunately, given the rarity of real blunders, there is very little statistical information available. Future work could compare the three different approaches presented in Appendix A.

13.5. Develop a Better Yellow Alert

Because there are heuristic components to the ALAS algorithm, it is difficult to develop an analytical basis for yellow alerting. We would like to have a yellow alert a few seconds before a red alert, but it is difficult to predict when a red alert is going to happen because of the uncertainty in the future trajectory. Therefore, the only strategy that we know how to pursue is to vary the parameters that influence the yellow alert until the statistics improve. Unfortunately, in phase I, we have not had adequate time to do this well. In future work, a yellow alerting methodology for closely spaced parallel runways could be developed and justified.

13.6. Tune/Test Algorithm for Other Runaway Spacings

In this study we have limited our experiments to approaches for parallel runways that are 750 ft apart. We could expand our studies to include other runway spacings. It is likely that the optimal values for the algorithm parameters vary with the runway separation.

13.7. Enhanced Automated Flight Modes

The escape maneuvers we studied were insufficient for the 750 ft parallel runway case. In order for the SAPA procedure to be used at this spacing, a more aggressive escape maneuver is required. One could envision a new automated flight mode in the flight guidance system, that performs a very aggressive escape maneuver, but this would not be available on existing aircraft and would depend upon a costly development life cycle of a new avionics suite. Alternatively, modifications to the SAPA procedure may be pursued that limit exposure to the fully abeam positioning of the aircraft.

13.8. An Intelligent Evasive Maneuver

The escape maneuver used in this study is a simple climb-turn and it is performed regardless of the relative position and velocity of the traffic at the time of the alert. In some of the CMF simulation cases, the ownship is below the traffic when the alert is issued. This most often occurs for the level-turn blunders. In this situation, the climb may only achieve a decrease in vertical separation, and it may be best for the ownship to delay the climb until the turn achieves adequate horizontal separation. There are also limited cases where an immediate turn or an increase in speed will narrow rather than increase separation. An intelligent escape maneuver, could improve separation for these select cases.

13.9. New Wake Studies and Trades

The current wake assumptions produce a highly constrained procedure. Options to trade procedure complexity and availability for increased approach speed differences or robustness to system errors will require more wake studies to determine the new wake-free boundaries for each trade.

13.10. Procedure Modifications to Improve Safety

Some procedure modifications with the potential to improve safety are using an offset approach for one or both aircraft and/or using an offset glidepath. The former provides greater horizontal separation; the latter provides greater vertical separation. However, under both options, the aircraft must still eventually converge on their respective runway centerlines, returning to a 750-ft cross-track separation and to an insignificant vertical separation. Therefore, these modifications represent solutions only if a blunder-free, along-track separation can be achieved prior to the convergence and maintained afterward. This may not be feasible with the current IGE wake-safe boundary of 1000 ft, which constrains the along-track separation from the centerline convergence to touchdown. Data in this study suggests that the collision-free boundary using ALAS may be in the neighborhood of 1000 ft and other options to expand the IGE wake-safe boundary may also be needed.

14. Conclusions

In this report, we have documented the progress in developing a practical SAPA procedure usable in closely spaced parallel approaches (as close as 750 ft apart) along with a tailored alerting algorithm. A new algorithm named ALAS was developed that combines features of a runway precision monitor and a conflict probe. The ALAS algorithm is highly configurable using a set of user-definable parameters.

A custom fast-time, low-fidelity kinematic simulator (tALAS) was developed to test the performance of the algorithm in conjunction with a proposed escape maneuver. The tALAS simulator uses kinematic models for the aircraft trajectories. This includes a simple turn model and constant ground speed and vertical speed accelerations. In this simulator, the ALAS algorithm was tested in the presence of millions of approaches with varying spacings and intrusion characteristics. Four different blunder models were pursued: (1) single-turn blunder, (2) single-turn blunder with altitude level-out, (3) double-turn blunder (see Appendix B), and (4) double-turn blunder with altitude level-out. We never encountered a case where the ALAS algorithm failed to issue an alert for a blunder. However, the distance at the closest point of approach varied significantly depending upon which blunder model was used. For the single-turn blunder, a minimum distance of 448 ft was obtained using the default values of the algorithm parameters. Using the double-turn model with altitude level-out, a minimum distance of 212 ft was obtained (see Appendix B). We expect that this could be improved with future work on the algorithm, but a satisfactory solution may require a more sophisticated escape maneuver that, for example, takes advantage of knowledge about the altitude of the blundering aircraft. However, a more fundamental question needs to be answered: exactly what type of blunder model should be used in the safety analysis for parallel runway studies? We are not sure at this time whether such a sophisticated blunder model is warranted.

The high-fidelity simulation returned a more negative result than the low-fidelity studies. The primary cause of this result was the sluggish turn response of the high-fidelity aircraft when executing the escape maneuver. The high-fidelity aircraft automated turn requires at least 6 s to achieve a 1°/s turn rate. This corresponds to a 5 s escape pilot delay in the kinematic simulations, which also had a very negative result. A more aggressive escape maneuver (the augmented maneuver) was explored, that improved the response of the high-fidelity aircraft significantly by reducing the latency to about 3 s. Nevertheless, this corresponds to a 3 s pilot delay in the kinematic studies that was found to result in collisions as well. This problem was further compounded when 1 to 2 s additional delay was added due to end-to-end latency of the traffic state in the ADS-B implementation.

The SAPA procedure for 750-ft parallel runways does not appear to be feasible at this juncture. Although we explored several options for an automated escape maneuver using the existing capabilities in a modern flight deck, the results indicate that the escape maneuver must be more aggressive than what we were able to achieve in the high fidelity simulation. However, additional turn performance may require development of a new autoflight mode. Using the SAPA procedure for 1000 ft or larger spacings looks more promising, but this has not yet been studied in sufficient detail to make any definitive recommendations.

15. References

[Abbott2002] Abbott, T.S: Flight Test Evaluation of the Airborne Information for Lateral Spacing (AILS) Concept. NASA Langley Technical Paper, NASA/TM-2002-211639, 2002.

[FAA2010] FAA. "Automatic Dependent Surveillance—Broadcast (ADS–B) Out Performance Requirements To Support Air Traffic Control (ATC) Service; Final Rule." Federal Register (National Archive and Records Administration) 75, no. 103 (May 2010): 30160 - 30195.

[FAA2012] FAA. digital - Terminal Procedures Search. http://aeronav.faa.gov/digital_tpp.asp?ver=1209&eff=08-23-2012&end=09-20-2012. Washington, D.C., August 23, 2012.

[Garmin2012] Garmin. "Datasheet for GPS 500W, GPS 400W, and GNC 420." https://buy.garmin.com/shop/store/assets/pdfs/specs/gps400w_spec.pdf (accessed September 4, 2012)

[Johnson2010] Johnson, Sally C., Terrance S. Abbot, Nelson M. G. Guerreiro, Gary W. Lohr, Paul Volk, and Burnell T. McKissick. Simplified Aircraft-based Paired Approach: Concept Definition and Initial Analysis. Hampton: NASA Langley Research Center, 2010 (under review).

[Massimini2006] Massimini, Vincent S. "Simultaneous Independent and Dependent Parallel Instrument Approaches: Assumptions, Analysis, and Rationale", MITRE, McLean, VA, 2006.

[Mohleji2010] Mohleji, Satish C., and Ganghuai Wang. Modeling ADS-B Position and Velocity Errors for Airborne Merging and Spacing in Interval Management Application. McLean: Mitre, 2010.

[RTCA2002] RTCA, Inc: "Minimum Aviation System Performance Standards for Automatic Dependent Surveillance Broadcast (ADS-B)", DO-242A, Washington, D.C., June 25, 2002.

[RTCA2006] RTCA, Inc: "Minimum Operational Performance Standards for Global Positioning System / Wide Area Augmentation System Airborne Equipment", DO-229D, Washington, D.C., December 13, 2006.

[RTCA2009] RTCA, Inc: "Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance – Broadcast (ADS-B) and Traffic Information Services – Broadcast (TIS-B)", DO-260B, Washington, D.C., December 2, 2009.

[RTCA2011] RTCA, Inc: "Minimum Operational Performance Standards (MOPS) for Aircraft Surveillance Applications (ASA) System", DO-317A, Washington, D.C., December 13, 2011.

[Samanant2000] Samanant, Paul and Mike Jackson: "Description of the AILS Alerting Algorithm", NASA/CR-2000-210109, May 2000.

[Shank1994] Shank, Eric M. and Katherine M. Hollister: "Precision Runway Monitor", The Lincoln Laboratory Journal. Vol. 7. No. 2. pp. 329-353, 1994.

[Winder2001] Winder, L. W. and James K. Kuchar: "Generalized Philosophy of Alerting with Application to Parallel Approach Collision Prevention", AIAA Guidance, Navigation, and Control Conference, Montreal, Canada, August 6-9, 2001.

Appendix A. Estimating False Alarm Rate

Intuitively, a false alarm occurs when an alert is issued on a non-blundering intruder trajectory. However, a rigorous definition of "false alarm" necessitates a mathematical definition of exactly which trajectories are blundering trajectories and which are not. This is surprisingly difficult to do. The problem is that there are many approaches and there is no a-priori means of deciding which approach is preferable to another. Three options for defining blundering are defined below.

A.1. Option 1: Any Deviation From Normal

In this approach, no attempt is made to delineate a class of blundering trajectories. Instead, normal trajectories are defined and any trajectory outside of this class is a blundering trajectory. The simplicity of this approach is appealing but it requires real data from actual landings for this approach to have any practical value. In [Winder2001] trajectories are described using several two-dimensional plots.

A.2. Option 2: Protection Zone

In this approach, a protection zone is defined. Only trajectories that carry the intruder into the protection zone should be alerted. All trajectories that do not carry the intruder into the protection zone yet cause ALAS to issue an alert are false alarms. This approach does provide a rigorous definition of a false alarm, but the definition of the violation zone can be non-trivial. The use of a circular disk around aircraft is problematic for closely spaced parallel runways. Using 400 ft diameter protection zones, there is an immediate loss of separation when the runway centerlines are 750 ft apart and the aircraft are abeam. Elliptical zones can help but the choice of the major and minor axes is non-trivial. Neither circular nor elliptical zones take the wake vortex risk into consideration. There is good reason to never allow the intruder aircraft to pass *in front of* the ownship. The risk of a catastrophic wake encounter or that the intruder might crash on the runway in front of the ownship argues for an infinite buffer size in front of the ownship. For these reasons we propose a strawman definition. The protection zone is a *moving* quadrant that is D distance away from the ownship's centerline (in the direction of the intruder) and X units behind the ownship. This is illustrated in Figure A-1.

We tentatively propose the following values of X and D: X = 400 ft and D = 230 ft. A violation occurs if the intruder enters the protection zone within the SAPA time interval. The SAPA time interval begins when the vertical separation becomes less than 800 ft and ends at decision height.

To estimate the false-alarm rate, one must construct trajectories that do not violate the protection zone. However, without any knowledge of the distribution of these trajectories in the real world, we do not know how to assign relative weights or probabilities to particular trajectories. Certain trajectories, (e.g. a turn right followed by a turn left) are probably more rare than a simple turn right. But in the absence of real blunder data, we have no credible way to assign these probabilities.

We will pursue this approach in future work.

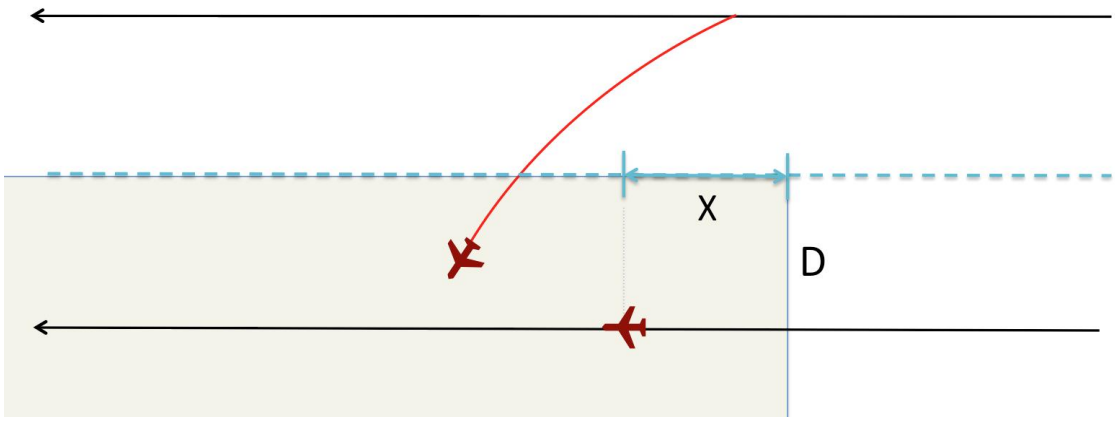


Figure A-1: A Strawman Definition of the Protection Zone for Parallel Runways

A.3. Option 3: Parametric Family of Blunder/Non-Blunder Trajectories

In this option, we do not seek to define a protection zone. Instead, we divide trajectories into two classes: blundering trajectories and non-blundering trajectories. The non-blundering trajectories can be defined parametrically in many ways. One attractive definition is all trajectories that stay within the runway conformance zone are non-blundering. This definition allows some of the non-blundering trajectories to have a short duration, high-bank angle turn followed by a corrective turn that never leaves the conformance zone. This type of trajectory stresses the trigger function (track-rate estimation) of the alerting algorithm and can be used to insure that the trigger is not too sensitive. However, if one seeks to calculate a percentage of false alarm statistics using this method, one is confronted with the reality that the statistic is highly dependent upon the definition of the parametric families. It is interesting that this approach can allow a third class of trajectories that are neither blundering nor normal.

Appendix B. Simulation Results Using Double-Turn Blunder

Towards the end of our study, we constructed a double-turn blunder trajectory in tALAS. The components of this trajectory are shown in Figure B-1. This blunder has two turn components defined by two different bank angles Φ_1 and Φ_2. The beginning and end times of these turn components are all variables.

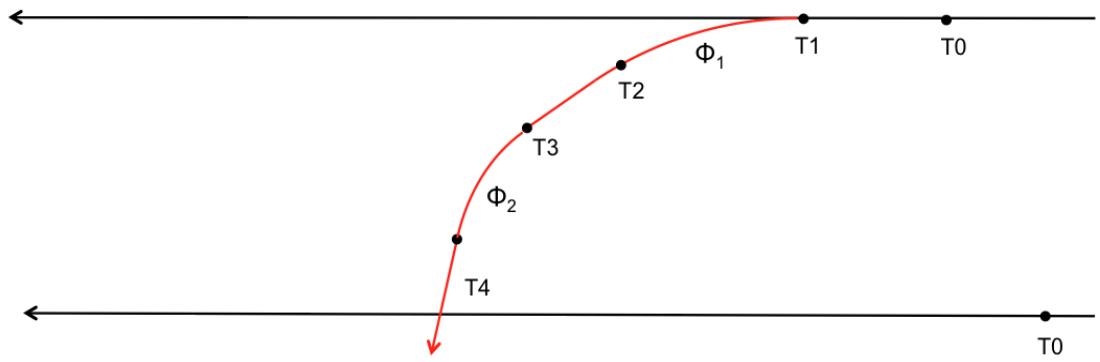


Figure B-1: Components of a Double-Turn Blunder

Unfortunately, adding this additional degree of freedom leads to enormous simulations times (e.g. multiple days), so we have had to be strategic in our experimentation. All of the results here are preliminary.

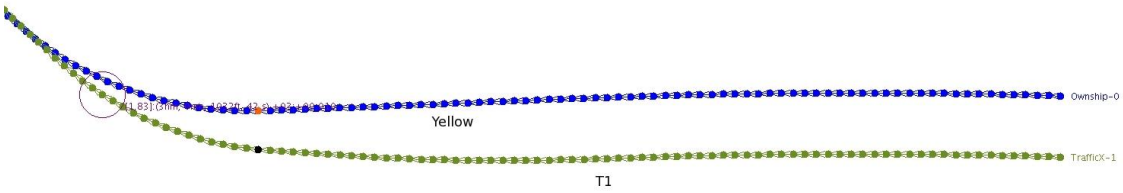
Table B-1 shows the parameter values for the double-turn blunder scenarios. We first ran a series of experiments with a coarse step size for the parameters (e.g. T4 step size of 5 s) and noticed that the worst case occurred when the first bank angle was 5° and the second bank angle was 30°. We then fixed these values and varied the other parameters using a smaller step size.

Table B-1: Trajectory Parameters for Double-Turn Blunder Scenarios

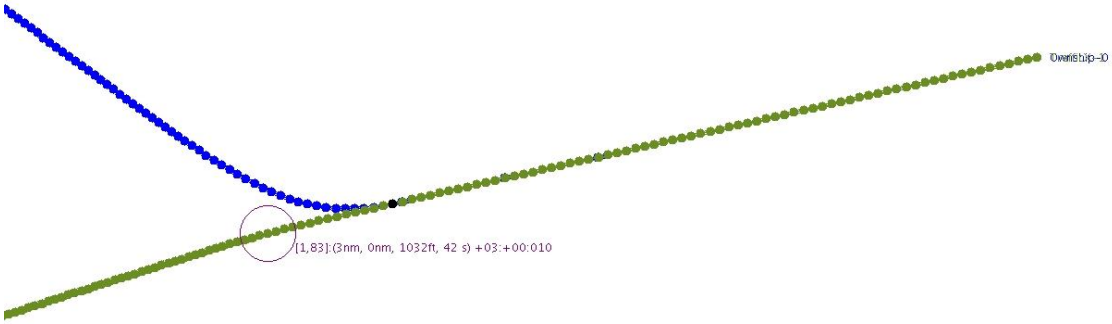
Parameter	Meaning	Min Value	Max Value	Step Size
T1	Start Time of Intrusion (s)	10	20	5
T2	Duration of Intrusion Turn 1 (s)	2	10	2
T3	Duration of straight segment after turn 1 (s)	0	6	2
T4	Duration of Intrusion Turn 2 (s)	2	10	2
bankAngle 1	Bank Angle of Intrusion (°)	5	5	5
bankAngle 2	Bank Angle of Intrusion (°)	30	30	5
Peak	Max Trajectory error (ft)	131	131	10
Period	Period of Trajectory error (s)	60	70	10
Phase	Phase of Trajectory error (°)	-180	+180	45
ownshipInitialSx	Distance from runway (NM)	5.0	5.4	0.2
intruderInitialSx	Distance from runway (NM)	5.0	5.4	0.2
ownshipInitialGs	Ground speed (KT)	160	170	10
intruderInitialGs	Ground speed (KT)	160	170	10

B.1. Double-Turn Blunder Without Altitude Level-out

First we ran without a vertical level-out and obtained a worst-case minimum distance of 284 ft. The horizontal distance was 234 ft and the vertical distance was 160 ft at TCA. This run is shown in Figure B-2.



(a) Horizontal View



(b) Vertical View

Figure B-2: Worst-Case Run Using Double-Turn Blunder

This is a particularly difficult scenario. The ownship's tracking error is bringing it closer and closer to the intruder before the blunder begins. The blunder begins with a very slow and gradual turn towards the ownship because the first turn has a bank angle of only 5°. Then when the aircraft are as close as possible without tripping the runway conformance test, the sharp turn begins. The first red alert occurred at time 34.5 s, or 14.5 s after the beginning of the intrusion. Interestingly, the yellow alert occurred at 27 s, a full 7.5 s earlier. The reason that the red alert was delayed was because of the parameter `ln_T_red` = 15. Because the first bank angle was 5°, the estimated time to cross the centerline was greater than 15 s for quite a while. This allowed the intruder to slowly get closer to the ownship. Then the intruder trajectory switched to a sharp 30° turn. Table B-2 shows details about the four different subcomponents of the ALAS detection algorithm

Table B-2: Performance of the ALAS Component Tests for Worst-Case No Level-Out, Double-Turn Blunder Scenario

Time (s)	Alert Level	Track-rate > Threshold (°/s)	alas_lines test (time-in) (s)	distAway Test (ft)	Runway-Conformance Test (ft)
27.0	1	No [0.84]	Yes [34.5]	No	No
27.5	1	No [0.86]	Yes [31.7]	No	No
28.0	1	No [0.87]	Yes [29.9]	No	No
28.5	1	No [0.89]	Yes [27.1]	No	No
29.0	1	No [0.90]	Yes [25.1]	No	No
29.5	1	No [0.91]	Yes [23.3]	No	No
30.0	1	No [0.93]	Yes [21.6]	No	No
30.5	1	No [0.66]	Yes [21.1]	No	No
31.0	1	No [0.39]	Yes [20.6]	No	No
31.5	1	No [0.12]	Yes [20.1]	No	No
32.0	1	No [0.14]	Yes [19.6]	No	Yes [132]
32.5	1	No [0.12]	Yes [19.1]	Yes [539]	Yes [141]
33.0	1	No [0.16]	Yes [18.7]	Yes [525]	Yes [151]
33.5	1	No [0.17]	Yes [18.1]	Yes [511]	Yes [169]
34.0	1	No [0.19]	Yes [17.6]	Yes [496]	Yes [182]
34.5	2	Yes [2.05]	Yes [12.8]	Yes [480]	Yes [199]
35.0	2	Yes [3.92]	Yes [12.3]	Yes [461]	Yes [212]

We note that the track-rate threshold test prevents the alas_circle probe from executing on turns with a small angular velocity. From this table we can see that the first yellow alert (level 1) is caused by the straight-line projection probe. This does not result in a red alert because the projected time to cross the centerline is 32.6 s, which is below the yellow threshold of 35 s but above the red threshold of 15.0 s. This suggests that a way to improve the minimum distance is by using a higher value of ln_T_red. But we also notice that the track rate threshold only reaches 0.89°/s for the initial turn. This suggests that a lower value of trackRateThreshold could also help, but this could also increase the false alarm rate.

Increasing the value of ln_T_red to 20 s improved the situation somewhat with a resulting minimum distance of 323 ft. However, the yellow alert was still 8 s ahead of the red alert.

So next we set trackRateThreshold = 0.75, which improved the situation significantly with a resulting minimum distance of 364.46 ft.

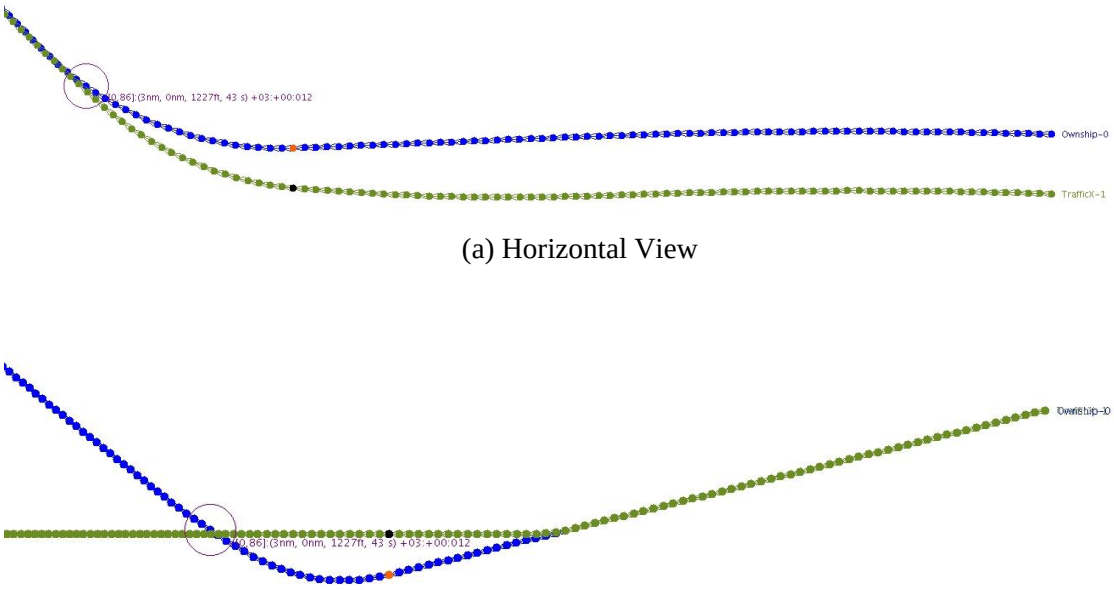
Finally, we tried (trackRateThreshold = 0.60, ln_T_red = 20, and redRunwayDist = 150) and we were able to improve the minimum distance to a value of 406.04 ft (397 ft horizontal, 82 ft vertical) These values resulted in a false alarm rate of 0.13%.

These results with the two-turn blunder model were obtained late in the study, so there was insufficient time to investigate the full effect of reducing the trackRateThreshold parameter. A value of 1.0°/s was more than adequate to detect the single turn blunder used in the body of this paper. In future work we will explore the reduction of this parameter to adequately detect the more sophisticated double-turn blunder including a level-out in altitude. It may also be necessary to design a better track-rate filter than was used in phase I of this study. The availability of real

data for aircraft approaches to parallel runways would greatly help this design.

B.2. Double-Turn Blunder With Altitude Level-out

We have only performed a few experiments with a double-turn blunder and an altitude level-out. The minimum distance at TCA was 212 ft, which is deep in the protection zone. The worst-case run is illustrated in Figure B-3. The circle indicates the point of closest approach and the orange dot indicates the point of the first red alert.



(a) Horizontal View

(b) Vertical View

Figure B-3: Worst-Case Run with Double Turn and Altitude Level-out

The altitude at TCA (43 s) was 1214 ft while it was 1088 ft at the time of the red alert at 33.5 s. The aircraft decelerates and reaches a minimum altitude of 1071 ft. The aircraft then climbs back up into the altitude of the level-out blunder. In this case, it would clearly have been better to stay at the lower altitude for a short while. In this particularly difficult scenario, the blundering aircraft also manages to closely follow the escape path in the horizontal dimension. We believe that decision logic could be designed which would select between two different kinds of escape maneuvers. In particular, a low altitude escape and a fast climb escape could be chosen depending upon the observed characteristics of the intrusion.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01- 02 - 2013	Technical Memorandum	

4. TITLE AND SUBTITLE

The Simplified Aircraft-Based Paired Approach with the ALAS Alerting Algorithm

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Perry, Raleigh B.; Madden, Michael M.; Torres-Pomales, Wilfredo; Butler, Ricky W.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

411931.02.07.01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

L-20197

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)

NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

NASA/TM-2013-217804

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 03
Availability: NASA CASI (443) 757-5802

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This paper presents the results of an investigation of a proposed concept for closely spaced parallel runways called the Simplified Aircraft-based Paired Approach (SAPA). This procedure depends upon a new alerting algorithm called the Adjacent Landing Alerting System (ALAS). This study used both low fidelity and high fidelity simulations to validate the SAPA procedure and test the performance of the new alerting algorithm. The low fidelity simulation enabled a determination of minimum approach distance for the worst case over millions of scenarios. The high fidelity simulation enabled an accurate determination of timings and minimum approach distance in the presence of realistic trajectories, communication latencies, and total system error for 108 test cases. The SAPA procedure and the ALAS alerting algorithm were applied to the 750-ft parallel spacing (e.g., SFO 28L/28R) approach problem. With the SAPA procedure as defined in this paper, this study concludes that a 750-ft application does not appear to be feasible, but preliminary results for 1000-ft parallel runways look promising.

15. SUBJECT TERMS

algorithms; landing aids; runways; simulations

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	70	(443) 757-5802