ANALYSIS OF AN APPROACH SPACING APPLICATION[1]

Ganghuai Wang, MITRE Corporation, McLean, VA

Jonathan Hammer, MITRE Corporation, McLean, VA

Abstract

The focus of this paper is on an application of Airborne Separation Assurance for final approach spacing. The objective of this application is to improve throughput onto existing runways during periods of high, sustained demand. The benefit of this application is economic – increased runway throughput equates to increased efficiency, greater schedule reliability, and other benefits. This application is being developed as part of the FAA's Safe-Flight 21 Program.

In this paper we will explain an analysis technique using Monte-Carlo simulations, we will give examples of the performance of a potential cockpit approach spacing algorithm (It should be noted that other alternative algorithms are being explored to meet the needs of this application – we examine one alternative herein), and we will discuss an example of the methodology being developed by industry to analyze safety for this class of applications.

Monte-Carlo Simulation

We used Monte-Carlo simulation software developed at The MITRE Corporation for this study. Random factors are introduced to various system parameters through the simulation and the simulation produces statistics that indicate runway throughput and the variation in that throughput. The Monte-Carlo simulation is described here.

For this study, we analyzed results for a configuration with a single runway. This is accomplished by running a simulated stream of aircraft to the runway. The stream is re-run many times, enabling the gathering of statistics on the runway inter-arrival time and spacing as a function of arrival number.

Simulation Input

The input for the simulation software includes the following:

- The number of Monte-Carlo repetitions
- The number of arrivals for each Monte-Carlo Repetition
- The separation standard, dependent on the weight category of the aircraft pairs
- An Inter-delivery distribution that is dependent on the weight category of the aircraft.
- An assumed distribution of initial speed, intermediate speed, and final speed of the aircraft. Final speeds over the threshold are weight category dependent.
- An Assumed distribution of deceleration rates.

The separation standards that we used in the simulation are in Table 1 [1].

Table 1. Separation Standard

Weight Category	Separation Standard (NM)
Large following Large	2.5
Small following Large	4.0
Heavy following Heavy	4.0
Large following Heavy	5.0
Small following Heavy	6.0

For the results presented in this paper, the nominal speed profile consisted of an initial speed of 210 knots approximately 20 nm from the runway threshold, transitioning to an intermediate speed of 170 knots at 15 nm from the runway threshold, transitioning to the final

[1] The contents of this material reflect the views of the authors. RTCA, the Federal Aviation Administration, and the Department of Transportation makes no warranty or guarantee, or promise, expressed or implicit, concerning the content or accuracy of the views expressed herein.

approach speed at approximately 5 nm from the runway threshold (see Figure 1).

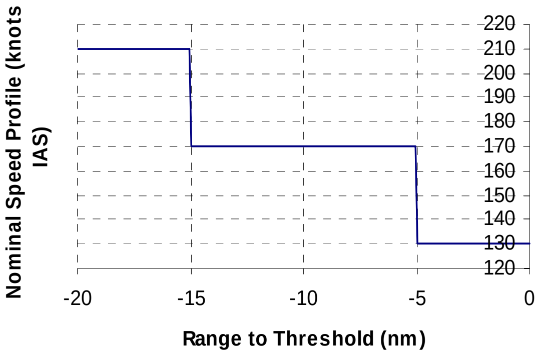

Figure 1. Nominal Speed Profile

Simulation Processing

For each run of the simulation, a stream of simulated aircraft is delivered onto the final approach course. As per [2], the inter-delivery distance between arrivals is a random variable that follows an extreme value distribution. Figure 2 shows a probability density function (PDF) of an extreme value distribution.

Figure 2. PDF of an Extreme Value Distribution

The mean value of the inter-delivery distance is dependent on the weight category of the aircraft pair. The rationale for using an extreme value distribution is that we expect air traffic controllers to act conservatively in delivering aircraft to final approach to ensure that the trailing aircraft avoids the wake vortex generated by the lead aircraft. Other probability distributions (e.g., Gaussian) can also be used to generate the inter-delivery distances.

The trajectory of each aircraft is generated based on aircraft characteristics and speed commands calculated by the simulated onboard approach spacing algorithm. The approach spacing algorithm is activated 20 nautical miles from the runway threshold. The approach spacing algorithm calculates a speed command based on the speed profiles of the lead aircraft and own ship (the trail aircraft). The current and predicted separation between the lead aircraft and the trail aircraft is calculated each second. When the projected separation is less than the separation standard, own ship is instructed to decrease its speed. When the current separation is less than the separation standard, a breakout command, that would instruct the crew to break off the approach, is issued by the algorithm. To simplify the simulation, the aircraft is allowed to fly to the airport runway threshold even if a breakout command has been issued.

Winds are accounted for in the projection calculations. Wind error, especially the along runway centerline component, has an effect of the predictions and consequently algorithm performance. Wind error is modeled by a colored recursive stochastic process using the weighted sum of the previous wind error and an new white noise component at each time epoch (every one second). To simplify the simulation, only the longitudinal position of the aircraft trajectory is affected by the wind error. An example of the wind error included in the simulation is shown in Figure 3.

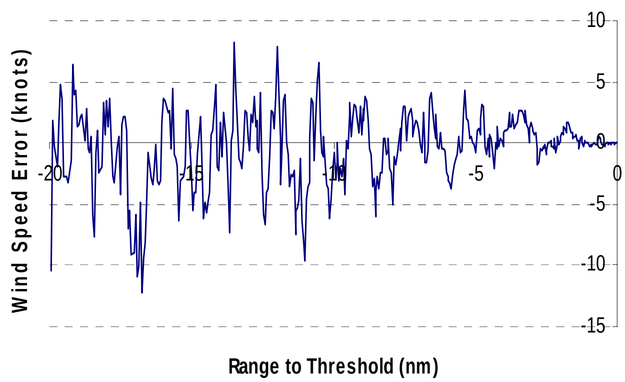

Figure 3. Wind Speed Error

Simulation Output

Statistics output by the simulation include the mean, standard deviation, minimum and maximum of the threshold inter-arrival time, and the inter-delivery time as output of the simulation. Histograms of the minimum separation for each aircraft pair are also produced for analysis. In a detailed mode of the simulation, the position, speed and acceleration of the lead aircraft and own ship are also recorded for each arrival pair.

The average inter-delivery rate and the average threshold inter-arrival rate can be calculated from the above data. A key modeling assumption is that if an aircraft pair breaks the required separation standard anywhere during the approach, the trailing aircraft must break out and go-around. One key metric of performance is the number of go-arounds per 1000 flights; our presumption is that for operational acceptability, the number of breakouts should be less than approximately 1 per thousand arrivals.

Characteristics of an Example Approach Spacing Algorithm

An approach spacing algorithm developed at MITRE is described here. In instrument meteorological conditions (IMC), the algorithm is based on the notion of not violating a minimum distance and the predicted separation is compared with the applicable separation standard. The algorithm can also be set up to use a minimum inter-arrival time over the threshold. This might be used, for example, in visual meteorological conditions (VMC). In this paper, we discuss results with a distance criteria and IMC since runway throughput improvements in IMC are considered to be of high interest. The goal of the algorithm, the performance criteria, and the performance are described below.

The Goal of the Algorithm and Performance Criteria

The goal of the approach spacing algorithm is to minimize the number of violations of the separation standard (i.e., the number of missed approach go-arounds) while maximizing the runway threshold throughput. Our metrics include the following.

(a) the number of violations of the separation standard per 1000 aircraft,

(b) the number of violations of the separation standard with minimum separations less than 14,000 feet per 1000 flights, and

(c) the average hourly throughput at the runway threshold.

(d) The number of speed changes commanded during the approach

Criterion (a) and criterion (c) are traded-off since conservative measures taken to reduce (a) can reduce (c) and aggressive measures taken to increase (c) can increase (a). The challenge is to develop an approach spacing algorithm that produces good performance in all criteria.

Sample Speed Profiles

Figures 4 and 5 illustrate sample speed profiles commanded by the algorithm. Commanded speed changes are indicated by a step in the speed profile. It is notable that the figures show a small number of commanded speeds outside those changes expected due to the nominal speed change profile.

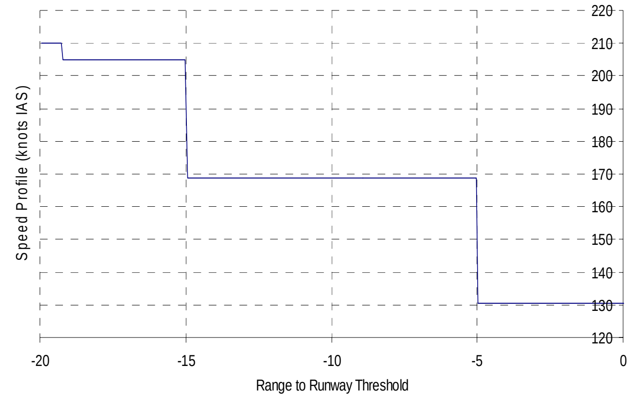

Figure 4: Sample Speed Profile

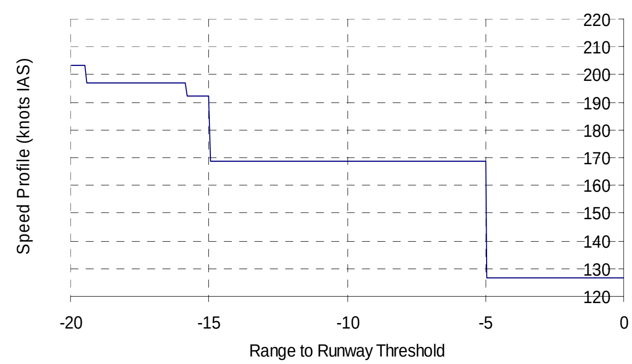

Figure 5: Sample Speed Profile

Performance of the Algorithm

To gather results, our simulations were run 2000 times with 50 arrivals per run, generating 10,000 simulated aircraft arrivals. The average rate of delivery onto the approach was 37.9 aircraft per hour unless specified otherwise. We ran simulations under different conditions that include the following:

(a) with and without wind speed error,

(b) with different nominal speed profiles,

(c) with different separation standards for large aircraft behind large aircraft

Results With the Addition of Wind Speed Error

Wind speed error is modeled as a function of altitude of the aircraft. A sample wind speed error for a single approach is shown in Figure 3. The performance of the algorithm, with and without the addition of the wind speed error model, is shown in Table 2.

Table 2. Performance with and without Wind Speed Error

	Without Wind Speed Error	With Wind Speed Error
Throughput (aircraft/hour)	36.76	36.79
Total number of violations of separation standard per 1,000 aircraft	0.4	0.9
Total number of violations of separation standard per 1,000 aircraft with separation no greater than 14,000 feet	0.18	0.4

Figure 6 shows the probability distribution of the minimum separation.

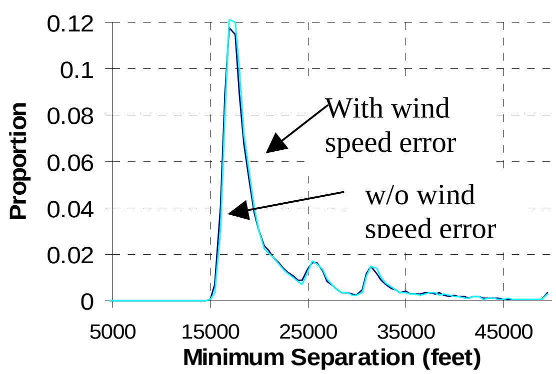

Figure 6. Probability Distribution of the Minimum Separation

Sensitivity to different Nominal Speed Profiles

To test the sensitivity of the algorithm to changes in the nominal speed profile, we examined performance with a second nominal speed profile. The second profile consisted of a speed of 180 knots from 20 miles from the threshold to the final deceleration point.

The performance with the profile change is shown in Table 3.

Table 3. Performance of the Algorithm when initial speed and intermediate speed are both 180 knots (when average inter-delivery rate is 36.14 aircraft per hour)

	Without Wind Speed Error	With Wind Speed Error
Throughput (aircraft/hour)	35.38	35.38
Total number of violations of separation standard per 1,000 aircraft	0.72	0.9
Total number of violations of separation standard per 1,000 aircraft with separation no greater than 14,000 feet	0.36	0.28

Sensitivity to Different Separation Standard

We examined results for a separation standard of 3 nmi for large following large. This is the separation standard at many US airports. Only those with high speed turnoffs normally operate with a 2.5 nmi in-trail standard for large aircraft following large aircraft.

The results are shown in Table 4.

Table 4. Results with 3 nmi Separation Standard

	Without Wind Speed Error	With Wind Speed Error
Throughput (aircraft/hour)	33.34	33.36
Total number of violations of separation standard per 1,000 aircraft	0.54	1
Total number of violations of separation standard per 1,000 aircraft with separation no greater than 17,000 feet	0.18	0.48

Distribution of Frequency of Speed Commands

We are interested in the number of new speed commands issued by the algorithm per approach. The probability mass function of that number for each aircraft in the simulation was calculated and recorded. A summary of the descriptive statistics is in Table 5. The mode is the point with the largest probability. In each of the three columns labeled Mode/Probability, Minimum/Probability, and Maximum/Probability, the first number is the mode, minimum, and maximum, respectively. The second number is the probability that the number of new speed command equals that number.

Table 5. Descriptive Statistics of the Number of New Speed Commands Per Approach

Aircraft #	Number of new speed command		
	Mode /Probability	Minimum /Probability	Maximum /Probability
1	1/0.493	0/0.066	8/0.002
2	1/0.311	0/0.126	9/0.003
3	1/0.265	0/0.166	9/0.006
4	1/0.242	0/0.188	9/0.008
5	1/0.257	0/0.194	11/0.001
6	1/0.253	0/0.201	10/0.001
7	1/0.225	0/0.229	10/0.002
8	1/0.258	0/0.218	10/0.001
9	0/0.227	0/0.227	9/0.009
10	1/0.238	0/0.196	10/0.001

The probability mass function (PMF) and the cumulative distribution function (CDF) of the number of new speed commands per approach for aircraft 1, 30, and 50 are shown in Figures 7, 8, and 9, respectively.

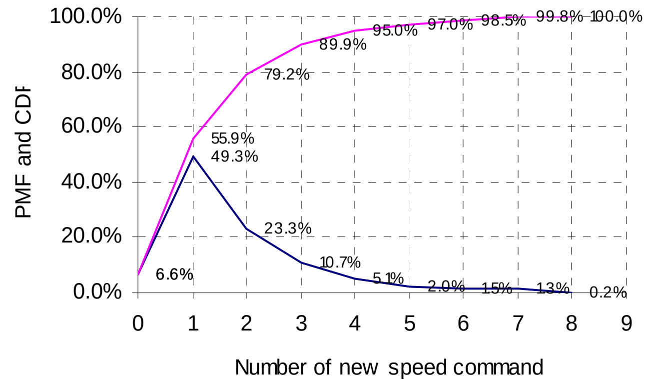

Figure 7. Probability Distribution of number of new speed command per approach for the second arrival aircraft

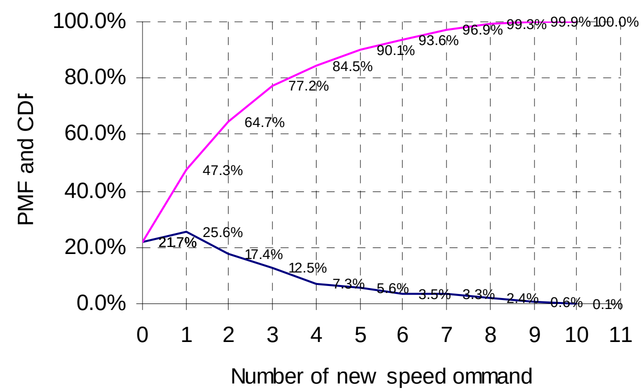

Figure 8. Probability Distribution of number of new speed commands per approach for the 30th arrival aircraft

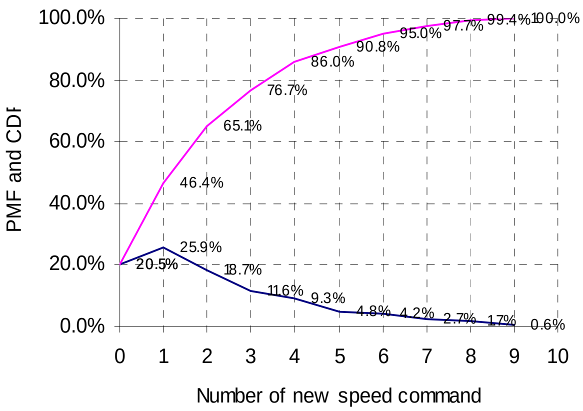

Figure 9. Probability Distribution of number of new speed command per approach for the 50^{th} arrival aircraft

Preliminary Analysis of Safety and Required Performance

This section will examine system safety considerations related to data integrity using the approach spacing algorithms described above. The section will illustrate some of the techniques being developed by industry for safety analysis of ADS-B applications as specifically applied to the approach spacing application.

The industry work is being developed in a consensus process by RTCA Special Committee 186 and Eurocae WG51. These groups are developing standards related to Automatic Dependent Surveillance Broadcast (ADS-B). More specifically, standards describing requirements for "Airborne Separation Assurance" are being written. Approach spacing is one airborne separation assurance application of ADS-B.

The primary motivation for developing such a safety analysis as part of the standards process is the recognition that operational applications such as approach spacing have dependencies on multiple sub-systems. The operational use of equipment for these applications, additionally, involves significant change to existing procedures. In order to properly specify sub-system requirements and provide assurance that equipment can safely support the desired procedures it is necessary to use a safety analysis to help develop proper requirements. It is felt that such a requirements safety analysis should be performed as the systems are developed, rather than as an after-the-fact analysis of equipment ready to be fielded.

The methodology for developing the standards involves the use of safety and fault-tree analysis. The methodology follows that outlined in RTCA DO-264 [3] and follows three basic steps:

1. Completion of an operational service and environment description (OSED)
2. Operational hazards are analyzed in the Operational Hazard Assessment (OHA)
3. Performance of an allocation of safety objectives and requirements (ASOR).

The operational service and environment description contains a detailed description of the application, operational procedures, and operational environment. The operational hazard assessment details the potential hazards involved in the operation, and the allocation of safety requirements and objectives analyzes each hazard, its resulting consequence, and criticality, then determines the system requirements that must be allocated to achieve the criticality for each operational consequence.

The allocated requirements will specify the required integrity, accuracy, availability, and continuity for each supporting subsystem. The approach spacing application, as envisioned using surveillance information provided by the Automatic Dependent Surveillance (ADS-B) system is supported by several subsystems including navigation, ADS-B, processing, and displays.

This section will provide an illustration of a part of the safety analysis and requirements allocation process using the approach spacing application as an example. We specifically focus on a fault-tree analysis of the approach spacing system. We investigate the required data integrity to support the operation without incurring failures resulting in a loss of separation or a near mid-air collision (NMAC).

The Monte-Carlo simulation and the approach spacing algorithm described in the previous sections provide an analytic framework on which the results in this section are based.

Examination of Faults Resulting In an NMAC

Figure 10 below depicts a fault tree that illustrates the potential paths that could result in a near mid-air collision (NMAC). (Note that all the fault trees presented have values at each node indicated. These are for illustrative purposes only, and do not indicate a final result). At the first sub-level of the fault tree, we observe three events that could, in combination, lead to an NMAC. An NMAC will occur if the guidance provided by the approach spacing system were in such error that the trailing aircraft were actually guided into an NMAC, if simultaneously air-traffic control (ATC) failed to resolve the problem, and if the Traffic Alert and Collision Avoidance System (TCAS) failed to resolve the problem.

For the purposes of this analysis, we will concentrate on the path delineated by the line through the fault tree, which involves the use of the ADS-B data and the related automation processing systems on-board the aircraft.

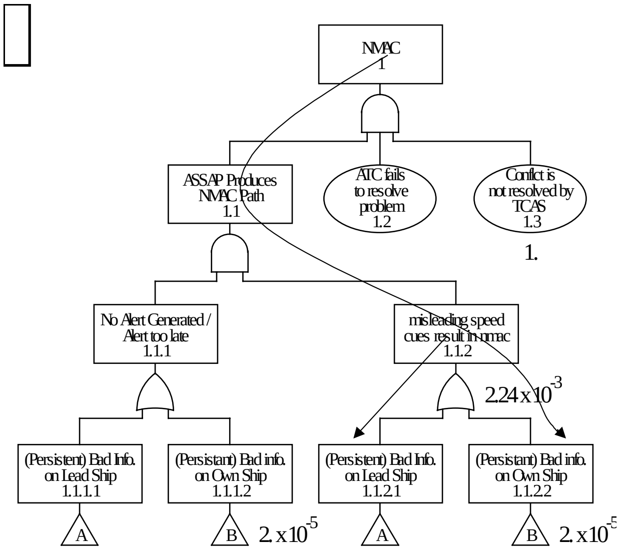

Figure 10: Top Level Fault Tree for Near Mid Air Collision Event

The left branch in the figure has the event that "ASSAP produces an NMAC path." Here ASSAP refers to the airborne surveillance and separation assurance processing that will make use of ADS-B and own-ship navigation inputs to derive guidance and alerts for display to the flight crew.

Two possibilities exist for the event of an NMAC path. These are that no alert is generated that indicates to the flight crew that there is a problem, or that misleading guidance (speed cues) will guide the crew into an NMAC. Bad guidance can result from bad data on own-ship or bad data on the lead ship. The assumption is that the data must be persistently bad for the fault to continue and the NMAC to result.

Figure JH2 traces the fault tree to a lower level for the case where there is bad data on own ship. Bad data on own ship can result from a navigation integrity failure, or a hardware or internal communications error. Our interest in this paper is in examining a navigation integrity failure.

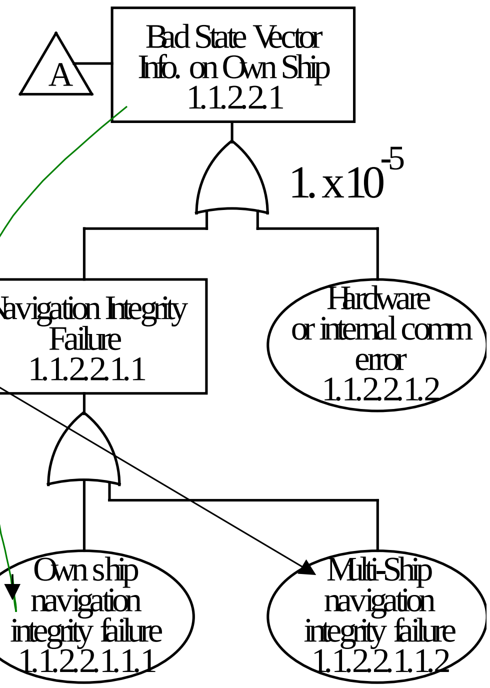

Figure 11: Bad State Data for Own Ship

Figure 12 illustrates a fault tree leading to bad data on the lead ship. There is additional complexity in the figure as compared with Figure JH2 because of the introduction of ADS-B communications to get information from the lead ship to own ship.

Although the primary focus of our discussion will be on the path through the fault tree indicated by the line, it is instructive to touch on the other paths as well.

The path labeled "excessive latency" indicates that the data that is being received by the trailing aircraft is too old. This would result from undetected failures in the transmitter in the lead aircraft and/or the receiver in the trailing aircraft. Another possibility for corrupted data is if there is a failure in the correlation and tracking mechanisms in the receiver. This can be the result of a track swap, or corrupted track identifiers.

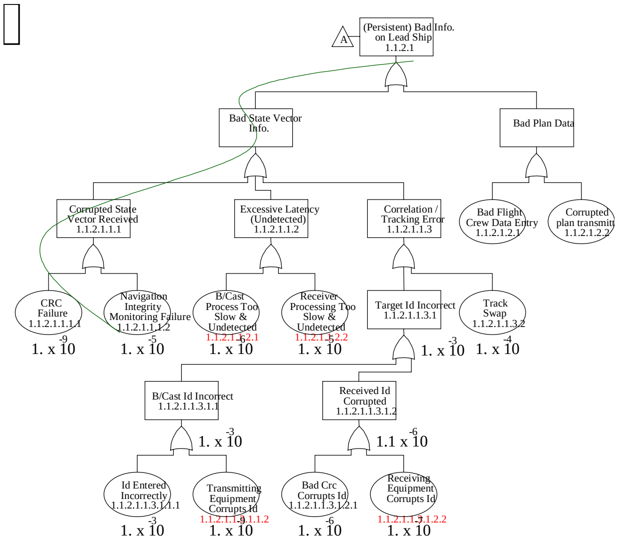

Figure 12: Bad State Data for Lead Ship

The path that we are most interested in for the analysis of this paper is the one indicated by the curved line through the fault tree. In this path, corrupted state data is received from the lead ship. The data is corrupted in a way that is persistent, undetected, and undetectable. An undetected integrity failure in GPS that results in a biased navigation position would be one example of such a failure.

In fact, we hypothesize that a bias error in the data would be the most likely form of such an error. We now determine the size of the bias errors that would result in enough error to allow guidance that would result in an NMAC.

We look at this problem from two perspectives: first, we examine the necessary bias error on one ship's navigation solution that would cause guidance leading to an NMAC. Next we look at the necessary errors on both ships that would cause guidance that would lead to an NMAC.

Figure 13 illustrates the effects of a undetected positional bias error. Note that for the purposes of this analysis, the biases were applied in the longitudinal (along runway centerline) direction. The bias in the cross-runway direction does not play a major role in the failure we are examining. Aircraft with bias errors had a single bias error value added to their position reports through the approach.

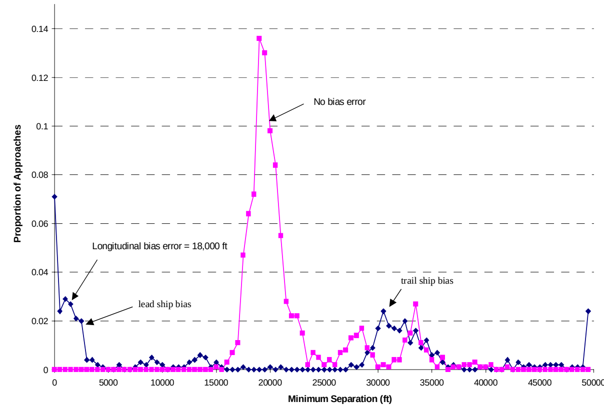

Figure 13: Histogram of Minimum Separations for Single Ship Positional Bias Error

The line in Figure 13 labeled "No bias error" shows the distribution of minimum separations from the Monte-Carlo simulation when there are no bias errors applied to either ship. This line shows there were no NMACs under this condition for the simulation sample paths that were run.

We applied increasing bias errors to the position measurement of one ship in further Monte-Carlo studies. The smallest bias error where a statistically significant proportion of NMACs occurred was when the bias was increased to 18,000 feet. This is illustrated in the figure by the distribution labeled "longitudinal bias error = 18,000 ft."

In our Monte-Carlo simulations, we use multiple arrivals in a stream to help determine algorithm performance. In order to model a single ship failure, we added the bias error to every other ship in the stream. The trail ship in one arrival pair is the lead ship for the next arrival pair. The bias error has a different implication for the lead ship as opposed to the trail ship. In one case, the bias error leads to reduced separation; in the other case, the bias error leads to increased separation. This is indicated in the figure with the annotations labeled "lead ship bias" and "trail ship bias."

Figure 14 shows achieved minimum separations for cases in which both lead and own-ship have undetected bias errors in their along-runway-centerline position measurements.

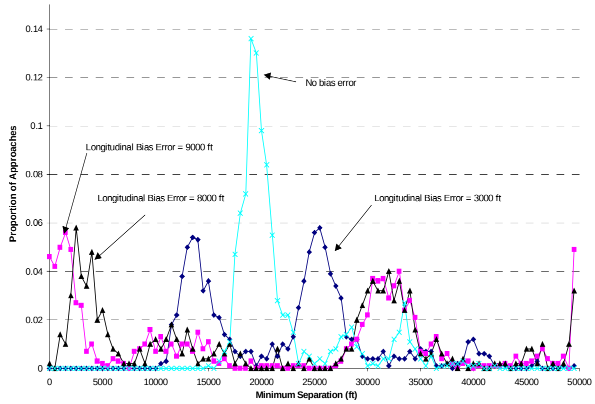

Figure 14: Histogram of Minimum Separations for Dual Ship Positional Bias Error

It is assumed that the bias errors for the lead and trail aircraft have the opposite sign. Thus the errors are additive, rather than canceling. The opposite signs result in the opposite condition when the trail aircraft for one approach pair is considered as the lead aircraft for the next pair. In consecutive approaches, the bias errors cause reduced, then increased separation.

Four sets of results are illustrated in Figure 14. The first curve, labeled "no bias error," shows a histogram of the separations achieved when no bias errors were added to either aircraft. Again, as in the single aircraft bias case, the bias errors were increased in separate simulation runs until a set of samples with some NMACs were observed. In this case, the minimum bias error that resulted in NMACs was 9000 ft. The other two curves show the results for a 3000, and 8000 ft error bias. This result is consistent with the previous result for the single ship case, where we found NMACs will occur with twice the dual ship error (18,000 ft).

Figures 13 and 14 indicate that NMACs due to bad guidance based are unlikely without extreme bias errors in position information feeding the guidance algorithms. Reasonable integrity values for position information should result in a very low, acceptable likelihood of NMACs.

Faults Leading to Separation Violations

A fault tree that will lead to separation violations is essentially the same as that leading to near mid-air collisions. The major difference is that the top level event in Figure JH1 is replaced by an event labeled "separation violation." The criticality of the separation violation event is much lower than that of NMAC.

We again examine the positional data bias errors that will result in a significant probability of a separation violation. Figure 15 shows the results of a Monte-Carlo simulation run with a single ship positional bias error of 1500 ft. In this example, we used 2.5 nmi, or 15190 as the separation standard. This represents the standard for like aircraft following like aircraft (e.g., large behind large) for a runway with a demonstrated runway occupancy time of 50 seconds or less [ref 70.65].

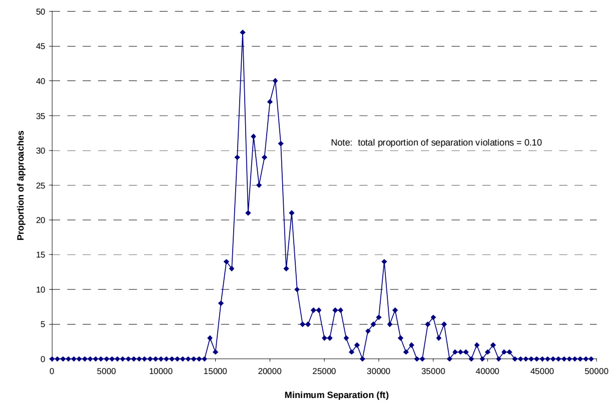

Figure 15: Single Ship Positional Bias Error Causes Separation Violation

The figure illustrates that a significant increase in separation violations results from a modest bias error (approximately 1/4 nmi). In the example of Figure 15, we found a separation violation rate of 1 per every 10 approaches. This represents an increase of two orders of magnitude over the unbiased results, presented earlier, of about 1 violation per 1000 approaches. As the figure illustrates, the violations are not extreme, and represent mostly technical violations of a few hundred feet.

Operational judgement will need to be applied to determine what constitutes a significant event from the standpoint of a separation violation.

Conclusions

We have presented an example methodology for achieving improved runway throughput via use of an approach spacing tool in the terminal area. The algorithm demonstrated here minimizes the number of speed changes issued to flight crews, and may be suitable for manual flight.

We have also analyzed the approach spacing system from a safety and failure mode perspective. We have found that the system is robust with respect to critical NMACs, but that it is sensitive to separation violations with small bias errors. Additional work will need to be done to determine requirements at both levels of criticality.

References

[1] US Federal Aviation Administration, Order 70.65M, Air Traffic Control.

[2] Warren, Anthony, Michael Ulrey, and Yaghoob Ebrahimi, 2000, "Final Approach Throughput Analysis for Conventional and Enhanced Air Traffic Management," DASC-2000

[3] RTCA, DO-264, Guidelines for the Approval of the Provision and Use of Air Traffic Services as Supported by Data Communications, RTCA Inc., Washington, DC, December 2000.

Biographies

Ganghuai Wang is a Lead Engineer with the MITRE Corporation's Center for Advanced Aviation System Development. Mr. Wang has more than four years experience in airport capacity simulation and modeling. He has a Ph.D. degree in Systems Engineering from the University of Virginia. Dr. Wang is a member of INFORMS and Tau Beta Pi.

Jonathan Hammer is a Principal Engineer with the MITRE Corporation's Center for Advanced Aviation System Development. Mr. Hammer has been developing surveillance systems for avionics and air-traffic control for the past 18 years, specifically working on air-traffic control radar tracking, TCAS, and ADS-B. He is the plenary secretary and a working group co-chair of RTCA special committee 186, developing standards for ADS-B. Mr. Hammer holds three US patents in the surveillance area.