

# A COMPARATIVE STUDY OF INTERVAL MANAGEMENT CONTROL LAW CAPABILITIES

*Bryan E. Barmore, Colin Smith and Susan O. Palmer, NASA Langley Research Center,  
Hampton, VA*

*Terence S. Abbott, Booz Allen Hamilton, McLean, Virginia*

## Abstract

This paper presents a new tool designed to allow for rapid development and testing of different control algorithms for airborne spacing. This tool, Interval Management Modeling and Spacing Tool (IM MAST), is a fast-time, low-fidelity tool created to model the approach of aircraft to a runway, with a focus on their interactions with each other. Errors can be induced between pairs of aircraft by varying initial positions, winds, speed profiles, and altitude profiles.

Results to-date show that only a few of the algorithms tested had poor behavior in the arrival and approach environment. The majority of the algorithms showed only minimal variation in performance under the test conditions. Trajectory-based algorithms showed high susceptibility to wind forecast errors, while performing marginally better than the other algorithms under other conditions. Trajectory-based algorithms have a sizable advantage, however, of being able to perform relative spacing operations between aircraft on different arrival routes and flight profiles without employing 'ghosting' methods. This comes at the higher cost of substantially increased complexity, however. Additionally, it was shown that earlier initiation of relative spacing operations provided more time for corrections to be made without any significant problems in the spacing operation itself. Initiating spacing farther out, however, would require more of the aircraft to begin spacing before they merge onto a common route.

## Introduction

Interval Management (IM) is an Automatic Dependent Surveillance-Broadcast (ADS-B)-enabled airspace operation that consists of extended metering, speed advisories for controllers and airborne spacing of aircraft. The FAA is in the process of an initial roll-out of IM capabilities including extended metering and speed advisories to deliver aircraft to the entry to the terminal area. These capabilities are

being coordinated through improvements to Time-Based Flow Management (TBFM) and En Route Automation Modernization (ERAM) activities. In parallel, the FAA is working through RTCA Special Committee-186 to develop Minimum Operational Performance Standards (MOPS) for the avionics systems to support the airborne spacing component of IM. The safety and performance requirements that support the MOPS were published in June 2011 as DO-328 [1].

For many years, several different research groups have been developing and refining the airborne spacing capabilities that will make up the airborne spacing avionics. The NASA Langley Research Center started with some early investigations of airborne spacing in the late 1970s in support of improved wake avoidance [2]-[5]. Work was revitalized in the late 90s as ADS-B became an expected surveillance source onboard the aircraft [6]. This work has continued through today with the focus of the NASA research being on improving runway throughput resulting from more precise inter-aircraft spacing [7]-[10]. To support this work, a prototype spacing tool was developed that used expected route information for both the ownship, also called the IM Aircraft, and the leading aircraft, called the Target Aircraft, to predict each aircraft's estimated time of arrival (ETA) to the runway threshold [11], [12]. Speed adjustments are calculated to make the difference in the ETAs match the controller-assigned goal, the Assigned Spacing Goal. Since routing information is available to the IM Aircraft, spacing can begin as soon as the Target Aircraft is within surveillance range. The current prototype is called ASTAR.

The Eurocontrol CoSpace project developed a different approach to provide airborne spacing capabilities [13]-[15]. Their tool does not require route information on the Target Aircraft; instead, the airborne spacing operation does not start until both the Target and IM Aircraft are in-trail or direct to a common, or merge, point. Prior to the merge point, a

simple dead-reckoning trajectory can be used to estimate the time of arrival at the merge point. Speeds are provided so that the IM Aircraft sequences the merge point after the Target Aircraft sequenced the point by a time specified by the controller, i.e., the Assigned Spacing Goal. Once in-trail, speeds are provided to position the IM Aircraft at the point where the Target Aircraft was at a time in the past equal to the Assigned Spacing Goal.

Both of these approaches have gone through extensive testing in simulation and flight trials and appear to perform sufficiently well. The main operational difference between the two approaches is the need for ASTAR to have access to route or flight path information for both aircraft to calculate ETAs and the CoSpace requirement that the aircraft be direct to a common point or in-trail before initiating the operation.

In addition to the operational differences, the two approaches have different performance characteristics. The CoSpace algorithm is largely insensitive to winds, especially once in-trail, as both aircraft experience similar winds at the same point in space. As the algorithm can be thought of as attempting to match ground speeds, wind speeds are irrelevant. Since ASTAR is doing a trajectory prediction, it is sensitive to the difference between the forecast winds used to build the trajectory and the truth winds experienced during flight. Also, since ASTAR inherently knows the expected speed along the trajectory, which is necessary to calculate the ETA, it is able to bound the output speed to keep the commanded speed relatively close to the expected speed. This has helped keep the commanded speeds operationally reasonable and consistent between different aircraft conducting the same operation and leads to stability in long strings of spacing aircraft [12], [16], [17].

While both of these approaches work well, we were interested in developing a modeling tool that would allow for rapid prototyping and testing of alternate approaches. This new tool would also allow for focused investigations of an algorithms' behavior under different conditions. We therefore have developed such a model and named it Interval Management Modeling and Simulation Tool (IM MAST). This paper reports on the development of IM MAST and the early results comparing several different spacing algorithms.

## IM MAST Design

The goal for IM MAST is to have a light-weight, fast, and extendable framework to support rapid prototyping of speed control algorithms and assess the effects of various error sources. To model aircraft performance, a simple trajectory is used where the three-dimensional path of the aircraft is pre-defined with no coupling between speed, altitude and lateral path. Turns are made with instantaneous changing in heading. At each calculation cycle, the point representing the aircraft is integrated forward along the fixed trajectory based on the current speed. A new speed is calculated from the spacing algorithm and the cycle repeats. For those algorithms that use trajectory prediction, separate predicted and truth trajectories can be provided. Likewise, separate forecast and truth wind fields can be provided. Position and speed information is shared between aircraft using an ADS-B-like communication link.

IM MAST starts with a researcher-defined scenario file that defines the predicted and truth trajectory for each aircraft as a series of trajectory-change points. These points are then connected with straight lines in both lateral and vertical dimensions. Each aircraft can have its own trajectory. The scenario definition also includes the forecast and truth winds, the initial position and speed for each aircraft, the Target Aircraft identity, and the Assigned Spacing Goal. Several aircraft can be linked together to form a string of aircraft. The model is then integrated forward using fixed time steps until the last aircraft has finished spacing (i.e., landed)<sup>1</sup>.

The structure of a generic spacing algorithm has been decomposed into four fundamental steps. Each step can be defined in IM MAST independently and then mixed-and-matched to create new algorithms. These steps are: spacing error calculation, spacing error filtering, speed correction of spacing error, and speed bounding. Two types of command speed implementation have been constructed. The first is an instantaneous speed that is updated with each calculation cycle. This is how both ASTAR and CoSpace calculate commanded speeds. The alternative is to calculate a new speed profile for the remaining trajectory. This is similar to how current

---

<sup>1</sup> Current testing is limited to arrival and approach operations. Future work will investigate spacing in other operational environments, such as en route cruise and departure / climb-out.

day Flight Management Computers attempt to meet Required Times of Arrival (RTA) constraints.

The spacing error calculation refers to the identification of the desired ownship position, be it based on time, distance, or ETA, which is then compared to actual ownship position to calculate current error. This error will either be time-based ( $x$  s of error) or distance-based ( $x$  nmi of error). The dimension of the Assigned Spacing Goal can be different from how the speed control algorithm calculates the spacing error. For example, the time history algorithm, described below, calculates the spacing error in distance even though it converts all Assigned Spacing Goals to time to calculate the spacing error.

Spacing error filtering refers to filters and gains of various forms which are applied to the spacing error. These may be applied with an intent to limit the severity of speed changes or to distribute speed corrections across the flight plan. Filters refer to an amount of error which is ignored, so that only the error above the filter is fed to the speed correction algorithm. The filter value may be dynamically changed based on a time- or distance-to-go schedule. Gains refer to multipliers applied to the error (greater than one will increase perceived error, while less than one will decrease perceived error) which impact the control law's sensitivity to error.

The speed correction algorithm uses the filtered spacing error to calculate a speed correction. The speed correction is added to a base speed, the speed that would be flown if there was no spacing error. Thus, spacing commands are a sum of the speeds needed to maintain ideal spacing and the speed change needed to reach ideal spacing.

Speed bounding refers to a process which restricts speed commands, either to prevent them from violating regulations or requirements, or to prevent them from issuing unreasonable commands. Regulation/requirement-based speed restrictions are usually hard-set, such as an absolute minimum speed of 100 kt or a restriction of a maximum speed of 250 kt when below 10,000 ft. Unreasonable speed commands are generally prevented by use of a profile-based speed restriction, where the aircraft is only permitted to deviate from an ideal profile by a certain percentage.

A subset of the spacing control algorithms developed to date is described in this paper. They include: station keeping, velocity-relative station keeping, time history, time prediction and relative ETA. Each is described in more detail in the following section.

### ***Spacing Control Algorithm Descriptions***

The following algorithms were selected from the literature or are variations on existing algorithms and form a range of ideas on how to calculate the spacing error and produce speeds to resolve the spacing error. Ref. [18] provides an overview of several different speed control algorithms that have been proposed and tested for use in airborne spacing operations.

#### **Error Correction Parameter**

All of the presented algorithms use a parameter to convert the spacing error into a speed correction term. For the relative ETA algorithms, the same parameters were used as for the NASA Langley ASTAR algorithm [11]. For consistency, the error correction parameters for each of the other algorithms were calculated to give the same speed correction for the same error input.

#### **Station Keeping**

Station keeping is perhaps the easiest airborne spacing approach to visualize [4], [19]. The goal is to have the spacing aircraft maintain a constant, along-track distance behind the Target Aircraft. The Target aircraft's path is recorded from the ADS-B messages. To calculate the current spacing error, the distance from the Target Aircraft to the ownship is measured along the Target Aircraft's path history. This distance is compared to the Assigned Spacing Goal and the difference is the spacing error. No filtering is applied to the spacing error. The new commanded speed is based on the Target Aircraft's current ground speed plus the speed correction term.

Station keeping is only viable when trying to maintain a distance-based spacing. Operationally, station keeping is typically limited to constant-speed, level-flight operations as changes in the Target Aircraft's ground speed immediately results in changes to the ownship's commanded speed. This makes a string of spacing aircraft behave like a train; as soon as the first aircraft changes speed then the entire string of aircraft must match that speed.

### **Velocity-Relative Station Keeping**

In an attempt to overcome some of the difficulties with station keeping operations, it has been suggested to adjust the distance-based Assigned Spacing Goal based on current ground speed. Our velocity-relative station keeping algorithm uses the Target Aircraft's current ground speed to convert the time-based Assigned Spacing Goal to a distance-based Assigned Spacing Goal. The standard station keeping algorithm is used with that new distance-based Assigned Spacing Goal. The base speed is the Target Aircraft's ground speed at the ownship's current position.

This algorithm smoothes out the transition when the Target Aircraft changes its ground speed. Velocity-relative station keeping only works when trying to maintain a time-based spacing.

### **Time History**

The time history, or constant time delay, algorithm requires the ownship to match the position where the Target Aircraft was at a point in the past given by the Assigned Spacing Goal [13], [20], [21]. The position of the Target Aircraft at the delay time is determined by interpolating between historic ADS-B position reports. The difference between the Target Aircraft's historic position and the ownship's current position is the spacing error.

The base speed for the algorithm's commanded speed is the Target Aircraft's ground speed at the ownship's current position. The speed correction term is added to the base speed to form the commanded speed.

In the time history operation, the ownship will attempt to follow the Target Aircraft's ground speed profile. Therefore, when the Target Aircraft makes a speed change, the ownship will make that same change at the same place in space instead of the same time as is done in station keeping.

Time history can be used with either a distance-based or a time-based Assigned Spacing Goal. In the former case the distance is converted to time using the ownship's current ground speed. In all of the analysis shown in a subsequent section, there was no difference in behavior between a time history algorithm given a time-based spacing goal versus a distance-based spacing goal; because of this, all figures show only the time-based time history calculations.

### **Time Prediction**

The time prediction algorithm can be thought of as the inverse of the time history algorithm [5], [22]. Instead of comparing the Target Aircraft's position in the past to the ownship's current position, the time prediction algorithm predicts where the ownship will be in the future and compares it to the Target Aircraft's current position. The ownship's predicted position is calculated using the ownship's current ground speed. The difference between the Target Aircraft's current position and the ownship's predicted position is the spacing error (measured in distance). The base speed for the speed command is the Target Aircraft's ground speed at the ownship's current location.

### **Profile-based Time Prediction**

One complication of how the prediction works in the time prediction algorithm is that it assumes the ownship will fly a constant speed for the near future. This will always put the ownship ahead of where it should be when the Target Aircraft is slowing down. A solution to this is to predict the ownship's position assuming it will follow the Target Aircraft's speed profile. This becomes a short-term trajectory prediction. If the prediction time, determined by the Assigned Spacing Goal, would place the ownship's position estimate beyond the Target Aircraft's current position then the Target Aircraft's current velocity and acceleration is used to extrapolate the trajectory.

The commanded speed is then calculated the same as in the normal time prediction method. The difference between the ownship's predicted position and the Target Aircraft's current position is the spacing error. The commanded speed takes the Target Aircraft's ground speed at the ownship's current location as the base and then adds the speed correction term.

### **Relative Estimated Time of Arrival**

The relative ETA algorithm is a fully trajectory-based algorithm [10], [11]. Instead of collecting ADS-B position and speed history for the Target Aircraft, the algorithm requires access to the ownship's and Target Aircraft's expected trajectories. These trajectories are used, along with current position of each aircraft, to calculate the ETA at a common reference point (called the Achieve-by Point). The difference in these ETAs is compared to the Assigned Spacing Goal to give the spacing error. The commanded speed is the speed correction added

to the nominal speed for the current leg from the ownship's trajectory.

An advantage of the relative ETA approach is that the aircraft do not need to be in-trail to provide spacing guidance. For all of the preceding algorithms, the spacing aircraft had to be following the Target Aircraft. However, this flexibility comes at the cost of additional information requirements. The ownship's trajectory is needed as well as information to construct the Target Aircraft's trajectory. Data requirements also include the need for reasonably accurate wind forecast information along both trajectories. The Target Aircraft's trajectory information could be replaced by a current ETA obtained from the Target Aircraft.

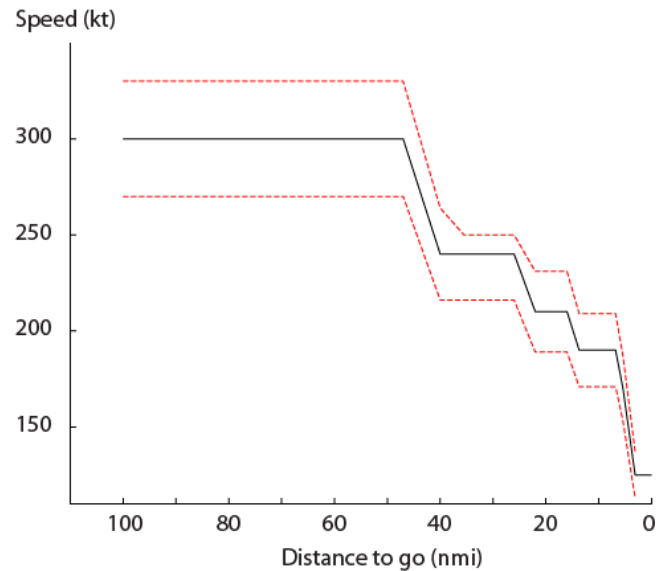
### Impact of Errors

There are many sources of errors or uncertainties during an airborne spacing operation that the control law must counter. Some examples include the initial scheduling time; winds and/or wind forecast errors; tracking of the expected trajectory, especially during turns and descent; and conformance to the commanded speed. Our first goal is to characterize the behavior of the above algorithms to these sources of error in isolation. Future work will look at additional error sources and the interplay of multiple error sources.

### Baseline Scenario

For all of the analysis provided in this paper, we use the same baseline arrival scenario. The aircraft are initiated at 100 nmi from the runway at 30,000 ft and 300 kt calibrated airspeed<sup>2</sup> on a straight-in approach. A straight line path is used to eliminate any turn effects including changes to the perceived wind direction. No Mach dynamics are modeled so cruise flight is at constant airspeed. The aircraft performs a near constant descent from 30,000 ft to the runway height of 660 ft. The speed profile is based on a modified Optimized Profile Descent arrival. The commanded speed is bounded by  $\pm 10\%$  of the profile speed, less than 250 kt below 10,000 ft and must exactly match the final approach speed (taken to be 125 kt for all conditions) by 3.1 nmi to the runway. This corresponds to having a stable approach by 1000 ft above the runway. The speed profile bounds

are shown in Figure 1. When a speed change is needed the aircraft is limited to an acceleration rate of  $\pm 1.2$  kt/s.



**Figure 1: Nominal Speed Profile along with Upper and Lower Bounds for the Commanded Speed.**

This study focused on single aircraft behavior so only two aircraft were in each traffic scenario. The Target Aircraft was initiated at the starting location. The ownship is initiated 120 s later at the same location. The Assigned Spacing Goal is 120 s.

For those algorithms requiring a distance-based Assigned Spacing Goal, the distance-based relative ETA algorithm was used to determine what distance would result in zero error at the runway and that distance, 4.4 nmi, was used as the Assigned Spacing Goal for all of the distance-based algorithms.

### Reaction to Target Aircraft's Speed Change

The initial study was to examine how each algorithm reacts to an unscheduled speed change by the Target Aircraft. Figure 2 shows the Target Aircraft flying at 250 kt and then approximately 17 nmi into the run slowing to 225 kt and maintaining that speed for the rest of the run. As expected, the station keeping algorithm slows at exactly the same time which corresponds to about 10 nmi in-trail. The speed profiles are identical except for this offset. The velocity-relative station keeping shows the undesirable effect of speeding up as the Target Aircraft begins its deceleration before finally decelerating to match the Target Aircraft's lower speed. This acceleration is due to the spacing

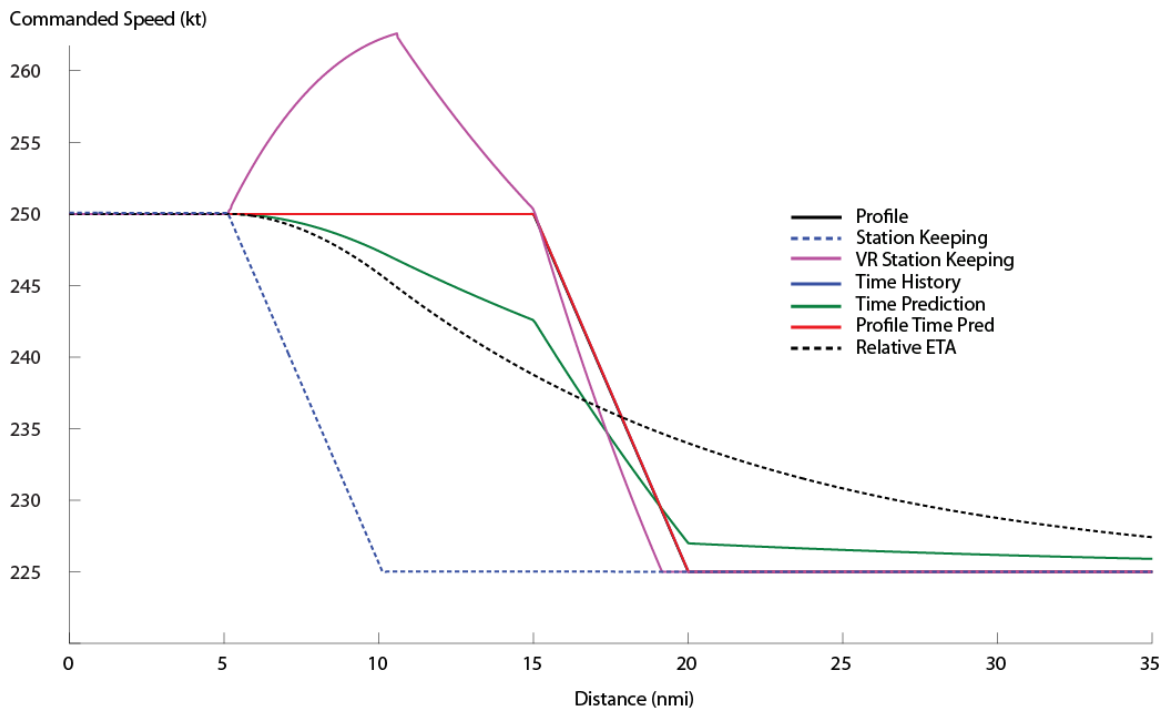
<sup>2</sup> All speeds are assumed to be calibrated airspeed unless otherwise explicitly stated.

goal conversion from time to distance uses the Target Aircraft's current ground speed. So, as soon as the Target Aircraft slows down, the effective spacing distance decreases requiring the ownship to increase speed. De Groot et al. [23], [24] have suggested an improved version of velocity-relative station keeping that includes having the ownship hold speed while the Target Aircraft is changing speeds. This prevents the increase in speed seen here.

The time-history exactly matches the Target Aircraft's speed profile as it was designed to do. The Profile-based Time Prediction algorithm also exactly matches the Target Aircraft's profile. The normal time prediction algorithm starts slowing down immediately after the Target Aircraft slows, albeit at a slower rate. The time prediction algorithm is assuming the Target Aircraft will maintain the same

speed as the ownship is flying so any change is seen as a spacing error and is corrected. The kink at 17 nmi is because the Target Aircraft has reached 225 kt and the spacing error is no longer growing. The time prediction algorithm takes a long time to settle out as seen by the speed slowly converging on the Target Aircraft's final speed.

The relative ETA algorithm also reacts to the Target Aircraft's by gently slowing down and converging slowly on the Target Aircraft's final speed. The ETA calculation point was approximately 100 nmi from the start so the speed change by the Target Aircraft had only a small initial effect on the Target Aircraft's ETA. And since the ETA is calculated with the nominal speed and not the current speed, the size of the spacing error is greatly reduced relative to the other algorithms.



**Figure 2: Speed Response to a Speed Change by the Target Aircraft.**

If the speed change by the Target Aircraft in Figure 2 had been a planned speed change on the profile, the relative ETA algorithm would have exactly matched the Target Aircraft's speed profile. The behavior of all of the other algorithms discussed would remain unchanged as none of them make use of the Target Aircraft's planned speed profile.

### Initial Scheduling Error

An IM Operation begins with the controller issuing a clearance to the flight crew to perform the IM Operation. It is expected that the controller has been provided with information that allows them to determine if the IM Operation would be feasible prior to issuing the clearance to the flight crew. Ideally, when the crew initiated the operation the spacing algorithm would calculate zero spacing error. Any deviation from this is called the initial scheduling error.

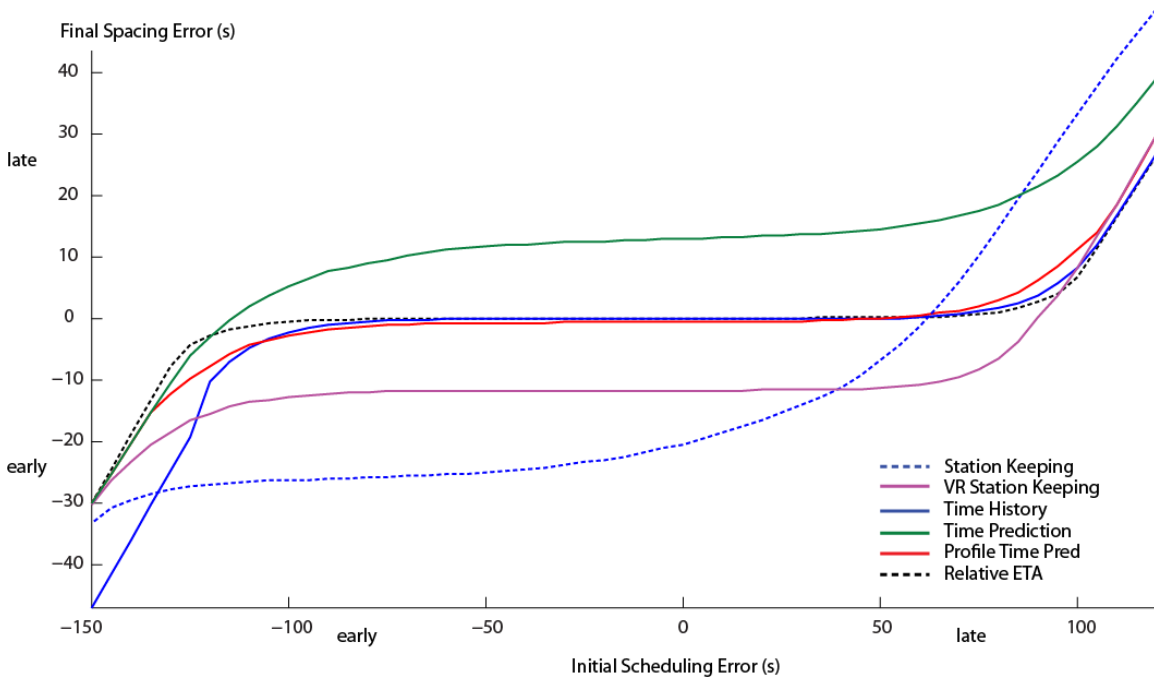
error. This deviation arises from the delivery of the aircraft not matching the desired scheduling of the aircraft. This is probably the simplest error for a control law to compensate for since it is a one-time source of error at the very beginning of the operation. The aircraft has the entire flight to overcome the error and any corrective action should only reduce the spacing error.

The initial scheduling error was implemented by changing the initiation time of the aircraft in the scenario. An aircraft with an initial scheduling error of 10 s early would be initiated 110 second after the Target Aircraft instead of the “nominal” 120 s. Test conditions ranged from 150 s early to 120 s late, [-150, +120]. Note that the 150 s early condition causes the spacing aircraft to be initiated 30 s *prior* to Target Aircraft’s initiation in the simulation.

Figure 3 shows the final spacing error at the runway threshold as a function of the initial scheduling error. The majority of algorithms are able to overcome initial scheduling error between [-100, +70] s. Beyond these limits, there is not enough speed authority to correct the initial error. However, even this simple case reveals some of the difficulties with the station keeping and velocity-relative station keeping algorithms. The shape of the velocity-relative station keeping curve is similar to the other but is displaced about 12 s early. The station keeping algorithm shows a similar but more pronounced shift early as well as moving the flat part of the curve to

the left. The reason for the bias towards arriving early is that when the Target Aircraft starts its final deceleration the spacing aircraft’s speed is being bounded by the 10% constraint. Referring back to Figure 1, the Target Aircraft reaches 125 kt at 3.1 nmi to go. To maintain zero spacing error, the spacing aircraft should be at the same speed. However, the zero spacing error position for station keeping is at 7.5 nmi which has a profile speed of 190 kt and a lower bound of 171 kt. So the spacing aircraft’s speed, due to these speed limits, is above the desired speed and it arrives early. To avoid this offset, the spacing aircraft would have to slow to 125 kt at a location where it would normally be flying 190 kt. This would be expected to cause operational difficulties as well as be unacceptable to the crew and aircraft operator. The exact values for the velocity-relative station keeping are different, but the same effect is seen to a lesser degree.

The time prediction algorithm shows the opposite bias, and is late by approximately 10 s. As seen in Figure 2, the time prediction algorithm will start slowing the ownship when the Target Aircraft begins its deceleration. But since the slowdown is less than in station keeping, it is not speed limited. This early deceleration then causes the ownship to be slower than the Target Aircraft over the last several miles prior to the final approach fix and thus arrives late.



**Figure 3: Final Spacing Error for each of the algorithm types as a function of Initial Scheduling Error.**

The upward curve to the right and the downward curve to the left are where the algorithms run out of speed authority. Due to the speed bounding, there is a finite amount of error than can be corrected before reaching the runway. The reason the flat part of the curves extend further to the left is that by slowing down, the aircraft has more time to correct the error. Flying the standard speed profile takes 1225 s. Flying the max speed profile, defined as accelerating to the upper bound and remaining there until reaching the final approach speed, takes 1132 s. This is a gain of 93 s. Flying the minimum speed profile takes 1350 s allowing an aircraft to lose 125 s.

So, for most algorithms, if the initial scheduling error is within the control authority available, they are able to resolve the error. The three algorithms that show a bias would have to be modified to remove the bias to make them operationally acceptable. For

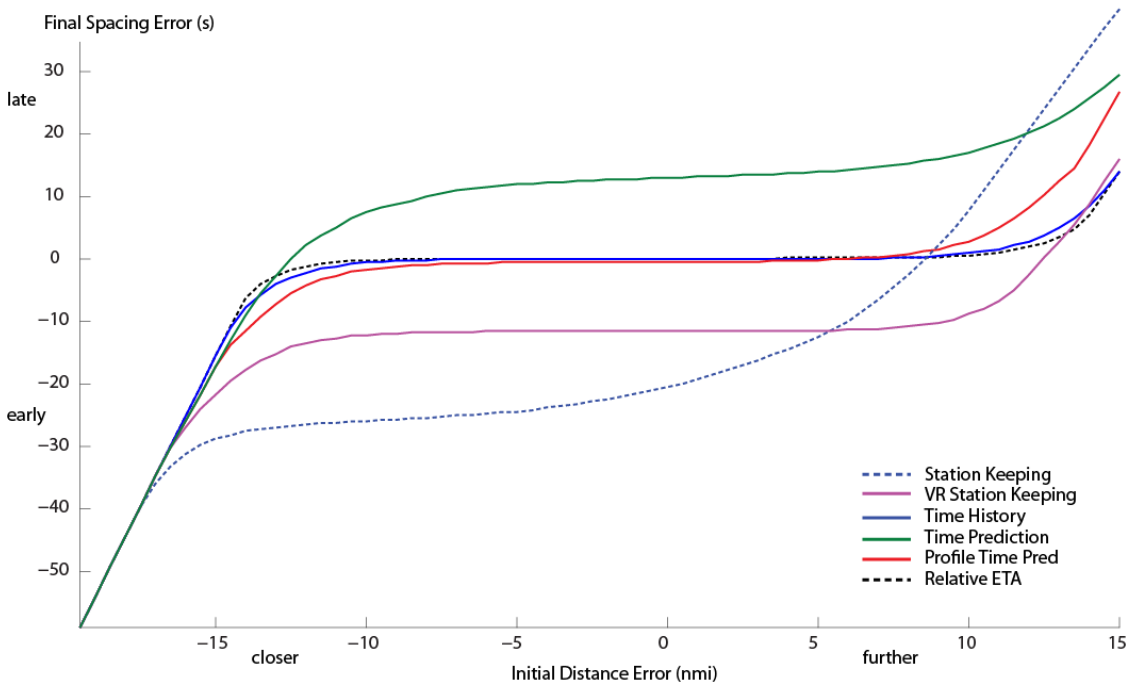
example, the profile-based time prediction algorithm corrects for the short-comings of the time prediction algorithm and would be a viable alternative.

#### Initial Distance Error

This condition is similar to the initial scheduling error except that the starting point of the spacing aircraft is moved instead of the starting time being adjusted. So every spacing aircraft is started 120 second after the Target Aircraft is initiated but the starting location is adjusted to be closer to the runway by 20 nmi or further away by 15 nmi, [-20, 15].

The overall behavior, see Figure 4, is similar to the initial scheduling error. Again the two station keeping-related algorithms have a bias towards arriving early due to not being able to slow sufficiently when the Target Aircraft decelerates to its final approach speed and the time prediction algorithm is late due to decelerating too soon.





**Figure 4: Final Spacing Error as a Function of Initial Distance Error.**

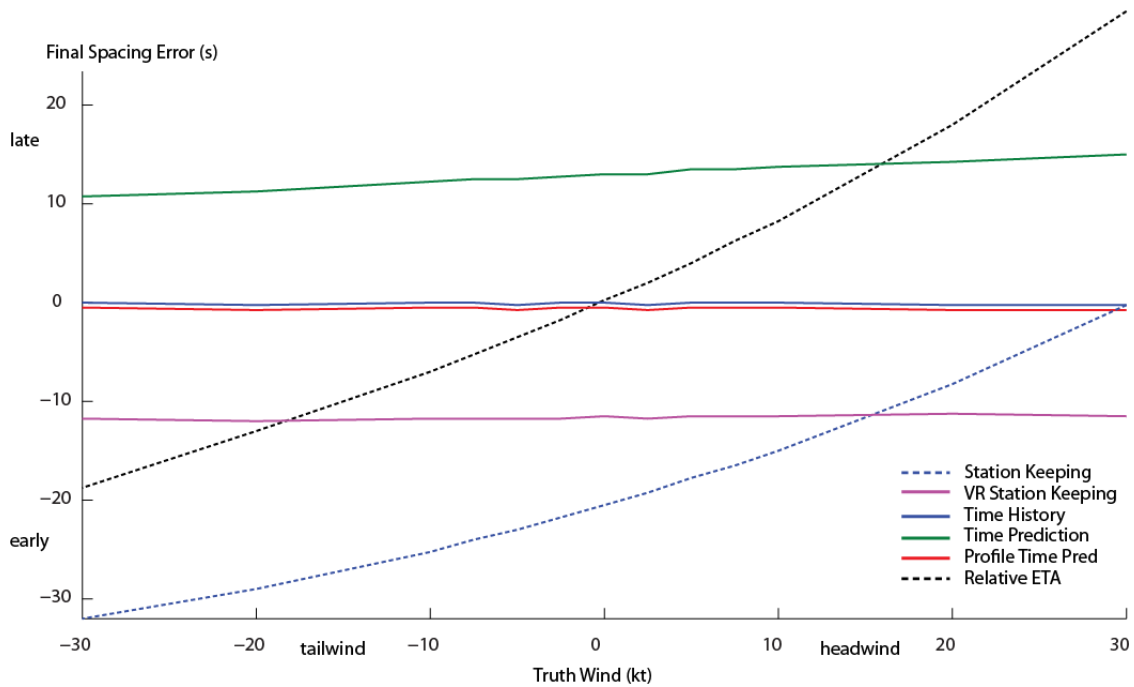
### ***Wind Forecast Error***

The next error studied was wind forecast error. Unlike the conditions discussed previously, this source is a continual error throughout the flight. For this analysis the aircraft were started at the zero scheduling error position and matching flight profiles. The wind forecast, also used for the scheduling, was for no wind. The truth wind was constant in both space and time and varied from a 30 kt headwind to a 30 kt tailwind, [-30, 30].

As would be expected from previous results, the station keeping and velocity-relative station keeping algorithms show an early bias and time prediction algorithm has a late bias. Since the velocity-relative station keeping algorithm uses the Target Aircraft's current ground speed for the conversion from distance to time, the spacing error at the runway is constant across wind conditions. The change in the spacing error for the station keeping algorithm is due

to the changing value of the conversion between distance and time as a function of the wind condition.

The relative ETA algorithm is dependent on an accurate wind forecast because this algorithm uses the forecast wind for the ETA calculation. While errors due to the poor wind forecast can be corrected far from the runway, there is not enough time to correct errors that occur close to the runway. For example, the aircraft is constrained to be at its final approach speed at 3.1 nmi from the runway and no more speed corrections are allowed. The calculated ETA using zero wind at that point would be roughly 89 s based on 125 kt final approach speed. However, with a 30 kt headwind, the ground speed over this segment would be 95 kt leading to a flight time of 117 s. This is a 28 second error over just the last 3.1 nmi. Adding the effects of altitude and the fact that the last speed adjustment has to occur prior to starting the final deceleration brings the error up to what is seen in Figure 5.



**Figure 5: Final Spacing Error as a function of Truth Wind for a Forecast of No Wind.**

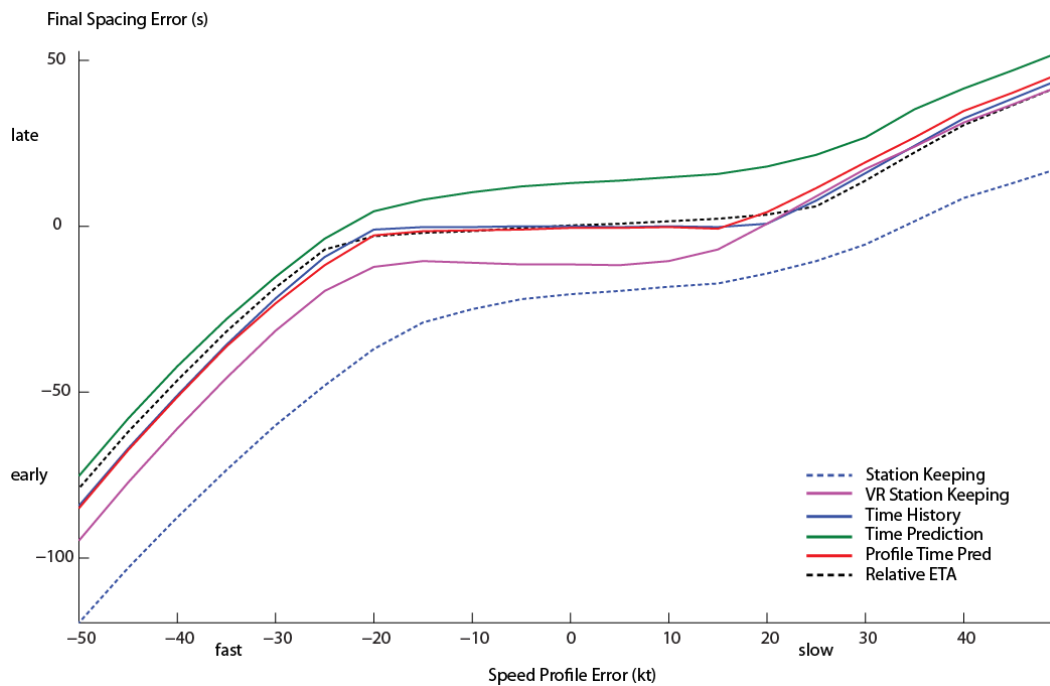
The time history and profile based time prediction algorithms do not show a dependence on the wind forecast error since they do not use the wind forecast for any predictions of the aircraft behavior and use the Target Aircraft's reported ground speed as the baseline for its speed. Using the Target Aircraft's reported ground speed automatically adjusts the baseline speed profile to account for the actual winds. This is an exact correction if there is no time-dependence to the winds and the two aircraft follow the same altitude profile. In an operational environment, this is not a perfect correction, but it is a good approximation, making the time history and time prediction algorithms more robust to differences between the actual and forecasted winds.

### ***Speed Profile Error***

The final condition reported in this paper is the condition where the Target Aircraft does not follow the expected speed profile. This models the case where the Target Aircraft is also spacing and has

adjusted its speed to meet its Assigned Spacing Goal, or the ownship or the controller's scheduling tool is using an erroneous trajectory for the Target Aircraft. For this analysis, we have modeled the Target Aircraft following its expected speed profile until reaching 10,000 ft, at which point it does a constant offset to the expected speed profile until decelerating to its final approach speed. The constant speed offset still respects the 250 kt upper limit. The speed offset varied from 50 kt slower to 50 kt faster, [-50, 50].

Final spacing errors are shown in Figure 6. Again, the station keeping and velocity-relative station keeping algorithms demonstrated an early bias in the final spacing error. At roughly  $\pm 20$  kt offset, the spacing algorithms can no longer compensate for the Target Aircraft's behavior. This is because the spacing aircraft are still being limited by the  $\pm 10\%$  bound on the nominal speed. Once the Target Aircraft is more than 25 kt off the nominal speed profile, the spacing aircraft cannot match the Target Aircraft's speed and the spacing error grows.



**Figure 6: Final Spacing Error As a Function of the Target Aircraft's Speed Profile Error.**

## Conclusions

This paper has presented the first analysis from a new tool (IM MAST) designed to help study the behavior of airborne spacing algorithms. This tool is expected to be used to support the development of avionics equipment standards through the Flight deck-based Interval Management (FIM) MOPS being developed by a joint RTCA/EUROCAE committee. This tool should also add insight to behavior and possible design trade-offs for the implementation of airborne spacing algorithms. For example, wind forecast results suggest that an airborne spacing tool that uses a relative ETA algorithm when the aircraft are not in-trial, to allow for use in the widest range of conditions, could be combined with a time history or time prediction algorithm that would be less sensitive to wind forecast errors close to the runway (or final spacing point).

This tool, IM MAST, uses a simple, deterministic model and a modular algorithm design format to allow for the rapid prototyping of control algorithms and the mixing of different components of the algorithms. Early work with this tool has focused on the behavior with respect to a set of common sources of spacing error: initial scheduling or delivery errors, wind forecast errors and Target Aircraft's conformance to the expected speed profile.

The simple station keeping and velocity-relative station keeping algorithms show a bias towards being early, i.e., reduced spacing, due to speed bounding enforced on the spacing aircraft. While the removal of the speed bounds would allow these algorithms to achieve the assigned spacing, it would require the aircraft to be at its final approach speeds at 7 nmi from the runway instead of the normal 3.1 nmi. The delivery performance of the relative ETA algorithm suffered in the wind forecast error condition since it used the erroneous wind forecast to predict both the Target Aircraft's and its own arrival times. In previous research, this has been largely overcome by allowing for updates to the wind forecast using ownship's sensed winds [25], [26]. Also, forecast winds at the surface are generally good.

Additional work is being planned to look at a wider range of control algorithms such as calculating the speed error term as a function of not only the spacing error but also the distance to the achieve-by point, thus applying the correction over the entire remaining trajectory instead of over just the next few minutes. It is also planned to look at more sources of spacing error including the interaction of multiple sources. IM MAST also allows for the investigation of string stability which is the behavior of several aircraft all spacing relative to the aircraft in front of them. Algorithms that might show good behavior for

a single aircraft might lead to unstable behavior if applied across several aircraft. The interaction of multiple algorithms in a string of aircraft is also of interest. The behavior of a mix of spacing algorithms operating together also needs to be studied to determine if they produce a stable operation.

## References

- [1] *Safety, Performance and Interoperability Requirements Document for Airborne Spacing-Flight Deck Interval Management (ASPA-FIM)*, RTCA Document number RTCA/DO-328, 2011.
- [2] L. Credeur, "Basic Analysis of Terminal Operation Benefits Resulting from Reduced Vortex Separation Minima," NASA, Washington, DC, NASA TM-78624, 1977.
- [3] W. J. Swedish, "Evaluation of the Potential for Reduced Longitudinal Spacing on Final Approach," FAA, Washington, DC FAA-EM-79-7, 1979.
- [4] T. S. Abbott and G. C. Moen, "Effect of Display Size on Utilization of Traffic Situation Display for Self-Spacing Task," NASA, Washington, DC, NASA TP-1885, 1981.
- [5] D. H. Williams, "Time-Based Self-Spacing Techniques Using Cockpit Display of Traffic Information During Approach to Landing in a Terminal Area Vectoring Environment," NASA, Washington, DC, NASA TM-84601, 1983.
- [6] E. Koenke and P. Abramson, "DAG-TM Concept Element 11, Terminal Arrival: Self Spacing for Merging and In-trail Separation," NASA, Advanced Air Transportation Technologies Project, Washington, DC, 2004.
- [7] T. S. Abbott "Speed Control Law for Precision Terminal Area In-Trail Self Spacing," NASA, Washington, DC, NASA TM 2002-211742, 2002.
- [8] G. W. Lohr *et al.*, "Flight Evaluation of a Time-based Airborne Inter-arrival Spacing Tool," in Proc. 5<sup>th</sup> USA/Europe ATM Seminar, Budapest, Hungary, 2003, Paper 56.
- [9] G. W. Lohr *et al.*, "A Time-based Airborne Interval-arrival Spacing Tool: Flight Evaluation Results," *Air Traffic Control Quarterly* vol. 13, no. 2, 2005.
- [10] B.E. Barmore *et al.*, "Airborne-Managed Spacing in Multiple Arrival Streams," in Proc. 24<sup>th</sup> Congr. Int. Council Aeronautical Sciences, Yokohama, Japan, 2004.
- [11] T. S. Abbott, "An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing to Include Parallel Runway Operations," NASA Washington, DC, NASA CR-2011-217194, 2011.
- [12] B. E. Barmore, "Airborne Precision Spacing: A Trajectory-Based Approach to Improve Terminal Area Operations," Proc. 25<sup>th</sup> Digital Avionics Systems Conference, Portland, Oregon, 2006.
- [13] E. Hoffman *et al.*, "Analysis of Constant Time Delay Airborne Spacing Between Aircraft of Mixed Types in Varying Wind Conditions," in Proc. 5<sup>th</sup> USA/Europe ATM Seminar, Budapest, Hungary, 2003, Paper 77.
- [14] D. Ivanescu *et al.*, "Effect Of Aircraft Self-Merging In Sequence On An Airborne Collision Avoidance System," AIAA, AIAA 2004-4994, 2004.
- [15] I. Grimaud *et al.*, "Spacing Instructions in Approach: Benefits and Limits From an Air Traffic controller Perspective," in Proc. 6<sup>th</sup> USA/Europe ATM Seminar, Baltimore, Maryland, 2005.
- [16] K. Krishnamurthy *et al.*, "Fast-Time Evaluations Of Airborne Merging and Spacing In Terminal Arrival Operations," AIAA, AIAA-2005-6143, 2005.
- [17] L. A. Weitz and J. E. Hurtado, "A Time-to-Go Control Law for Spacing Vehicles at a Point," presented at AIAA Guidance, Navigation, and Control Conference, Portland, OR, August 9–11, 2011.
- [18] T. S. Abbott, "A Brief History of Airborne Self-Spacing Concepts," NASA Washington, DC, NASA CR-2009-215695, 2009.
- [19] J. A. Sorensen *et al.*, "Traffic Alert and Collision Avoidance System (TCAS) - Cockpit Display of Traffic Information (CDTI) Investigation," FAA Washington, DC, DOT/FAA/RD-91/8, 1991.

[20] J. R. Kelly and T. S. Abbott, "In-Trail Spacing Dynamics of Multiple CDTI-Equipped Aircraft Queues," NASA, Washington, DC, NASA TM 85699, 1984.

[21] D. H. Williams and D. C. Wells, "Jet Transport Flight Operations Using Cockpit Display of Traffic Information During Instruments Meteorological Conditions, Simulation Evaluation," NASA, Washington, DC, NASA TP-2567, 1986.

[22] S. L. Chappell and E. A. Palmer, "In-Trail Following During Profile Descents With A Cockpit Display Of Traffic Information," in Proc. 1<sup>st</sup> Int. Symp. Aviation Psychology, 1981, p. 94.

[23] J. M. C. de Groot *et al.*, "Pilot Support for Distance-Based In-Trail Following Tasks," in Proc. 13<sup>th</sup> Int. Symp. Aviation Psychology, 2005, p. 139.

[24] J. M. C. de Groot *et al.*, "Distance-Based Self-Spacing in Arrival Streams," AIAA, AIAA-2005-6275, 2005.

[25] K. Krishnamurthy *et al.*, "Airborne Precision Spacing in Merging Terminal Arrival Routes: A Fast-time Simulation Study," in Proc. 6<sup>th</sup> USA/Europe ATM Seminar, Baltimore, Maryland, 2005.

[26] F. Wieland *et al.*, "Performance of Airborne Precision Spacing under realistic weather conditions," in Proc. 30<sup>th</sup> Digital Avionics Systems Conference, Seattle, WA, 2011.

*31<sup>st</sup> Digital Avionics Systems Conference*  
*October 14-18, 2012*