Conflict Detection Performance Analysis for Function Allocation Using Time-Shifted Recorded Traffic Data

Nelson M. Guerreiro[1], Ricky W. Butler[2], Jeffrey M. Maddalon[3], George E. Hagen[4], and Timothy A. Lewis[5]

NASA Langley Research Center, Hampton, VA 23681, USA

The performance of the conflict detection function in a separation assurance system is dependent on the content and quality of the data available to perform that function. Specifically, data quality and data content available to the conflict detection function have a direct impact on the accuracy of the prediction of an aircraft's future state or trajectory, which, in turn, impacts the ability to successfully anticipate potential losses of separation (detect future conflicts). Consequently, other separation assurance functions that rely on the conflict detection function – namely, conflict resolution – are prone to negative performance impacts. The many possible allocations and implementations of the conflict detection function between centralized and distributed systems drive the need to understand the key relationships that impact conflict detection performance, with respect to differences in data available. This paper presents the preliminary results of an analysis technique developed to investigate the impacts of data quality and data content on conflict detection performance. Flight track data recorded from a day of the National Airspace System is time-shifted to create conflicts not present in the un-shifted data. A methodology is used to smooth and filter the recorded data to eliminate sensor fusion noise, data drop-outs and other anomalies in the data. The metrics used to characterize conflict detection performance are presented and a set of preliminary results is discussed.

Nomenclature

alt	=	altitude, as a function of time
C	=	single conflict or conflict set, true
$\hat{C}$	=	single conflict or conflict set, predicted
d_{GCD}	=	great-circle distance function
D_H	=	horizontal distance
D_V	=	vertical distance
δ_t	=	time value for conflict comparison criteria
δ_H	=	horizontal distance value for conflict comparison criteria
δ_V	=	vertical distance value for conflict comparison criteria
Δt_{LOS}	=	relative time-to entry into loss-of-separation
$\Delta t_{LOS,mean}$	=	mean time-to loss-of-separation at first detection
Δt_{traj}	=	track data and trajectory prediction time interval
ε	=	component error vector
f_{GCD}	=	great-circle projection function
gs	=	groundspeed, as a function of time
H_{sep}	=	horizontal separation criterion
lat	=	latitudinal position, as a function of time
lon	=	longitudinal position, as a function of time
$N_{\hat{c}}$	=	number of predicted conflicts

---

[1] Aerospace Engineer, Crew Systems and Aviation Operations Branch, MS 152, AIAA Member.
[2] Research Engineer, Safety-Critical Avionics Systems Branch, Mail Stop 130.
[3] Research Engineer, Safety-Critical Avionics Systems Branch, Mail Stop 130.
[4] Research Engineer, Safety-Critical Avionics Systems Branch, Mail Stop 130.
[5] Sub-Project Manager, Crew Systems and Aviation Operations Branch, MS 152, AIAA Member.

N_C	=	number of true conflicts
$N_{C,sim}$	=	number of true losses-of-separation in one simulation run
N_{FA}	=	number of false conflict alerts
N_{MA}	=	number of missed conflict alerts
N_{MD}	=	number of loss-of-separation missed detections
P_{FA}	=	probability of false alert
P_{MA}	=	probability of missed alert
P_{MD}	=	probability of missed detection
R_s	=	surveillance range
$\boldsymbol{s}$	=	position state vector
$\hat{\boldsymbol{s}}$	=	position state vector estimate
σ_r	=	surveillance horizontal position error standard deviation
σ_a	=	surveillance altitude error standard deviation
σ_{gs}	=	surveillance groundspeed error standard deviation
σ_{vs}	=	surveillance vertical speed error standard deviation
σ_{trk}	=	surveillance track angle error standard deviation
t	=	time
$\boldsymbol{T}$	=	true trajectory
$\hat{\boldsymbol{T}}$	=	predicted trajectory
t_{cd}	=	conflict detection time
T_{det}	=	conflict detection cycle period
t_{LOS}	=	time at entry into loss-of-separation
T_{pred}	=	time horizon for trajectory prediction and conflict detection
t_s	=	surveillance time
$t_{shift,max}$	=	maximum track time shift
t_{sl}	=	surveillance lag time
trk	=	track angle, as a function of time
$\boldsymbol{v}$	=	velocity state vector
$\hat{\boldsymbol{v}}$	=	velocity state vector estimate
vs	=	vertical speed, as a function of time
V_{sep}	=	vertical separation criterion

I. Introduction

NASA has an extensive history of research in separation assurance (SA) concepts and technologies[1]. Over the past few years, this work has been focused on demonstrating the feasibility, and characterizing the performance of, various separation assurance function allocation concepts. Function allocation refers to the assignment of the separation assurance functions between humans and automation and between airborne or ground-based systems. The current focus under Function Allocation at NASA is to take a broader approach to the separation assurance function allocation problem by developing the ability to make early recommendations about various allocation architectures. Using a system-level analysis that leverages knowledge acquired under prior work as well as new fundamental relationships for the performance of the SA functions under different allocation environments, NASA will be able to provide recommendations about the separation assurance function allocation architectures with the highest benefit potential under different future scenarios.

A set of fundamental relationships that describe the performance of the conflict detection (CD) function under different data environments is needed. In separation assurance systems, the conflict detection function is critical to the overall system performance because the conflict resolution function necessarily relies on a conflict detection function to compute conflict-free resolution maneuvers. A set of relationships that characterize the performance of the conflict detection function, independent from conflict resolution, in terms of the fundamental differences between the input parameters will enable the evaluation and comparison of various possible allocations of the conflict detection function. This work is focused on identifying those relationships.

In this paper, a description of the analysis technique developed to identify the relationships in conflict detection performance is provided. The details of the recorded traffic data used in the analysis and the conditioning that was performed on that data are presented. Preliminary results from the portion of the analysis capability that had been

fully integrated at the time this paper was written are presented, including sensitivity of the performance metrics to the input parameters.

II. Background

Conflict detection performance (the ability to predict potential future losses-of-separation between aircraft) is largely driven by the accuracy of trajectory prediction for the future states of an aircraft, and trajectory prediction is highly dependent on the quality and content of the data available to the agent performing the prediction. Consequently, the allocation of the conflict detection function, assuming co-located trajectory prediction, between airborne systems and ground-based systems can also be characterized by the content and quality of the available data. The performance of the conflict detection function is also dependent on the allocation between humans and automation; however, this work does not address that impact to performance because there are other NASA research activities tasked with characterizing the human-automation axis in more detail. Specifically, the conflict detection performance that is the subject of this work seeks to answer the question:

What are the impacts of surveillance quality and level of intent on conflict detection performance?

An extensive body of work exists in the literature with respect to trajectory prediction[2,3] and conflict detection.[4-6] A large portion of prior work is focused on the design of conflict detection algorithms and the performance of those implementations in simulation. Additionally, a significant amount of work has been done to study the human performance in the face of different conflict alerting systems.[7,8]

The performance of various conflict detection and alerting algorithms have been analyzed in prior work.[9-13] Of particular interest is the work by Paglione[9] and Bilimoria.[10] Paglione evaluated the performance of the Federal Aviation Administration's (FAA) Host computer detection algorithms while Bilimoria performed a similar evaluation for the NASA Center/Tracon Automation System's (CTAS) detection algorithms. Prior work has also studied conflict detection independent of an algorithm, where research has shown that a conflict detection algorithm can be developed that is correct for an arbitrary trajectory within a given approximation bound.[14] Regardless of how many trajectory-pairs a particular correct algorithm examines, its performance, given the same input data quality and data content, will only differ from another correct algorithm by factors such as trajectory prediction sophistication, alerting buffers, and other heuristics.

The impacts of data quality and data content, in terms of surveillance and navigation error[15,16] and intent availability[17-20], have been discussed in prior work in the context of separation assurance. Chung[15] discussed the implication of surveillance delays and how those could be used to determine buffers for the trajectory prediction of future states while Finkelsztein[18] looked at the sensitivity of the range at closest point of approach in a parametric analysis.

The investigation of prior work in the area of conflict detection performance revealed that more work is needed to provide the architecture-agnostic and broadly-applicable characterization of conflict detection performance that will support the Function Allocation system-level assessment. In much of the prior work, it is difficult to separate the performance of the entire SA system, which is typically concerned with observed losses-of-separation, from the performance of conflict detection alone, which is typically characterized by false and missed alert rates. Prior work consisted of algorithm design, human factors implications, and full SA system performance, whereas this work is focused on the performance of the conflict detection function as a function of the quality and content of the input data.

III. Analysis Approach

Two analysis methods were chosen to answer the question of conflict detection performance. The first method uses trajectory data recorded in the National Airspace System (NAS) and time-shifting to create losses-of-separation that were not present in the subject traffic day. This is the method discussed in more detail in this paper. The second method uses a canonical "circular scenario" approach to simulate aircraft traversing a subject region of airspace. Each analysis method has its own advantages and disadvantages but their results are expected to be complimentary. This second analysis method, as well as comparison of results from the two methods, will be the subject of future publications.

A time-shifted, recorded traffic set has many advantages for conflict detection analysis. Recorded traffic data has the trajectory complexity present in the current system that is typically hard to model in simulation, such as step-climbs or step-descents. Realistic turn and climb performance for various aircraft at various flying conditions is afforded by these recorded trajectories. No trajectory generator is required to fly the recorded trajectories, which

reduces complexity and improves computational speed. However, recorded traffic data does add challenges, such as surveillance error and data fusion issues that need to be addressed before the data can be used for analysis.

The analysis method for this work involves simulating a set of traffic scenarios, while varying a defined set of parameters, to evaluate the performance of the conflict detection function. Trajectory prediction and conflict detection are exercised at specified intervals to predict conflicts between aircraft trajectories and to compute a set of performance metrics. Surveillance quality and level of intent are varied parametrically in the analysis to investigate the impacts on conflict detection performance. Only the analysis capability that had been fully implemented at the time of this writing is discussed in this paper.

A. Definitions and Overview of Conflict Detection

The review of prior work revealed several different metrics relating to conflict detection but the definitions varied. Those definitions varied, especially with respect to conflict resolution where, for example, a specific loss-of-separation event may never come to fruition, thereby rendering the idea of a missed detection somewhat unclear. As such, it is important to begin with a presentation of the key set of definitions used throughout this paper (Table 1).

Conflict detection can be defined as the identification of potential future losses-of-separation between any pair of aircraft using the predicted future trajectories of those aircraft. As such, trajectory prediction has a significant impact on conflict detection performance. The sophistication of the prediction, in terms of the content and quality of information available to predict a future trajectory, and the resultant accuracy of the prediction can have a significant impact in the performance of conflict detection. For example, a trajectory prediction informed only by the current state of an aircraft is likely to be less accurate at predicting that aircraft's future state than one that has additional knowledge of the aircraft's intent. Additionally, there is a dimension of trajectory prediction accuracy that is associated with the sophistication of the trajectory generator used to compute that prediction. In this effort, the simplest implementation of trajectory prediction that achieves the desired results will be used. That is, a simple linear projection of states will be implemented first, followed by the possible introduction of a kinematic trajectory generator that uses acceleration regions to create more realistic turns, with more complexity added as dictated by the required maximum level of uncertainty in the results.

Table 1. Conflict detection terms and definitions.

Term	Definition
loss(es)-of-separation (LOS)	violation(s) of the minimum separation criteria between two aircraft trajectories
separation criteria	vertical and horizontal distances that identify the boundary for a LOS event between two trajectories
conflict	two aircraft are in conflict if there exists a time, t, with a LOS in their true or predicted trajectories
detection	the identification of a true LOS event
true conflict	a conflict that exists between the true trajectories of two aircraft and that is within the conflict detection horizon, in the absence of a resolution maneuver
predicted conflict	a conflict that exists between the predicted trajectories of two aircraft and that is within the conflict detection horizon
alert	a warning provided by the conflict detector
false alert (FA)	an alert regarding a LOS <u>*that is not*</u> present in the true aircraft trajectories within the conflict detection horizon
missed alert (MA)	the lack of an alert regarding a LOS <u>*that is*</u> present in the true aircraft trajectories within the conflict detection horizon
missed detection (MD)	over all detection attempts, the absolute failure to detect a true LOS event

Trajectory propagation methods for conflict detection can be classified as probabilistic, deterministic, and worst-case. In a deterministic case, a single trajectory prediction is assumed to have a 100% probability of occurring. In the probabilistic case, a set of trajectory predictions each has a weight or probability of occurring, based on the current knowledge. Finally, a worst-case approach assumes 100% probability on any possible trajectory for an aircraft, thereby defining a bounded protected region. Each of these methods requires its own set of assumptions for implementation; the choice can be made based on the use case. Because of the NAS-wide nature of the results being sought here, and because this study explores other parameter variations in a Monte Carlo approach, the deterministic trajectory propagation method was chosen for this work.

B. The Analysis Approach

The conflict detection analysis is comprised of the following primary elements: a track playback/simulation framework, a surveillance error model, a trajectory prediction module, and a conflict detection module. The playback framework begins at the start time for the first time-shifted track, increments the simulation time at the CD cycle interval (T_{det}), and ends when the last time-shifted track has completed. As each flight is spawned into the simulation environment, a surveillance error model adds surveillance error to the track data for that flight. During each simulation cycle (CD cycle), a trajectory prediction is made for each active aircraft that predicts the future state of that aircraft up to the trajectory prediction and conflict detection horizon (T_{pred}) and conflict detection is performed between aircraft pairs that are within the surveillance range specified (R_s). The conflict detection module returns a set of predicted future conflicts based on the input trajectories. Statistics are collected from the set of predicted and true conflicts from each conflict detection cycle to compute the conflict detection metrics. Table 2 shows a comprehensive list of the analysis input parameters that will be discussed throughout this section.

Table 2. Analysis input parameters.

Parameter	Description	Units
T_{det}	conflict detection cycle period	seconds (s)
T_{pred}	trajectory prediction and conflict detection time horizon	seconds (s)
Δt_{traj}	track data and trajectory prediction time interval	seconds (s)
H_{sep}	horizontal separation criterion	nautical miles (NM)
V_{sep}	vertical separation criterion	feet (ft)
R_s	surveillance range	nautical miles (NM)
t_{sl}	surveillance lag time	seconds (s)
σ_r	surveillance horizontal position error standard deviation	nautical miles (NM)
σ_a	surveillance altitude error standard deviation	feet (ft)
σ_{gs}	surveillance groundspeed error standard deviation	knots (kn)
σ_{vs}	surveillance vertical speed error standard deviation	feet/minute (ft/min)
σ_{trk}	surveillance track angle error standard deviation	degrees (deg)
δ_t	time value for conflict comparison criteria	seconds (s)
δ_H	horizontal distance value for conflict comparison criteria	nautical miles (NM)
δ_V	vertical distance value for conflict comparison criteria	feet (ft)

The surveillance model is defined by standard deviation parameters for a set of Gaussian error distributions relating to the level of error on the position and velocity states of an aircraft. Each surveillance technology has its own unique error signature and the choice of Gaussian distributions was chosen for convenience as the simple compromise amongst the many possible error distributions.

A fixed surveillance lag parameter, t_{sl}, is applied equally over all aircraft tracks to simulate the un-compensated time delay that may exist between surveillance sampling time and the time of trajectory prediction. At any conflict detection cycle time, t_{cd}, the true track position states of an aircraft are given by $\boldsymbol{s}(t_{cd})$, where,

$$\boldsymbol{s}(t) = [lat, lon, alt] \tag{1}$$

and the velocity states are given by $\boldsymbol{v}(t_{cd})$, where,

$$\boldsymbol{v}(t) = [gs, vs, trk] \tag{2}$$

Linear interpolation is used to compute a true track data point at the surveillance time when the surveillance lag time is not an integer increment of the recorded track data. At that same conflict detection cycle, the surveillance position estimate used for trajectory prediction, $\hat{\boldsymbol{s}}(t_{cd})$, is given by the true position at surveillance time, $\boldsymbol{s}(t_s) = \boldsymbol{s}(t_{cd} - t_{sl})$, plus the surveillance error components in position and velocity, $\boldsymbol{\varepsilon}_s$ and $\boldsymbol{\varepsilon}_v$, respectively. The surveillance error components are computed using equations (3)-(7) as,

$$[\varepsilon_{lat}(t_s), \varepsilon_{lon}(t_s)] = \boldsymbol{f}_{GCD}\big(lat(t_s), lon(t_s), N(0, \sigma_r), U(0,360)\big) - [lat(t_s), lon(t_s)] \tag{3}$$

$$\varepsilon_{alt}(t_s) = N(0, \sigma_a) \tag{4}$$

$$\varepsilon_{gs}(t_s) = N(0, \sigma_{gs}) \tag{5}$$

$$\varepsilon_{vs}(t_s) = N(0, \sigma_{vs}) \tag{6}$$

$$\varepsilon_{trk}(t_s) = N(0, \sigma_{trk}) \tag{7}$$

where the horizontal position error components, $\varepsilon_{lat}(t_s)$ and $\varepsilon_{lon}(t_s)$, are computed using a great-circle projection from the true position at surveillance time, $lat(t_s)$ and $lon(t_s)$, with a radial distance error sampled from a Gaussian distribution, $N(0, \sigma_r)$, and an azimuth sampled from a Uniform distribution, $U(0,360)$. The great-circle projection function, f_{GCD}, computes a latitude and longitude position projected some arc-distance along an initial track on the surface of a sphere from a starting latitude and longitude position. Finally, the surveillance position and velocity states used at conflict detection time are given by:

$$\hat{s}(t_{cd}) = [lat(t_s) + \varepsilon_{lat}(t_s), lon(t_s) + \varepsilon_{lon}(t_s), alt(t_s) + \varepsilon_{alt}(t_s)] \tag{8}$$

$$\hat{v}(t_{cd}) = [gs(t_s) + \varepsilon_{gs}(t_s), vs(t_s) + \varepsilon_{vs}(t_s), trk(t_s) + \varepsilon_{trk}(t_s)] \tag{9}$$

Figure 1 shows a diagram of the surveillance error modeling described above.

The trajectory prediction module implements a simple state-projection model. In the absence of intent information, and without a high-fidelity trajectory generator that is beyond the fidelity of the results sought from this study, this is the simplest implementation for trajectory prediction. The state projection follows the set of equations given by equations (10) and (11), where the great-circle projection function is again used to project from the current estimated horizontal position using the current estimated groundspeed and track angle:

$$\left[\widehat{lat}(t_{cd} + \Delta t), \widehat{lon}(t_{cd} + \Delta t)\right] = f_{GCD}\left(\widehat{lat}(t_{cd}), \widehat{lon}(t_{cd}), \widehat{gs}(t_{cd}) * \Delta t, \widehat{trk}(t_{cd})\right) \tag{10}$$

$$\widehat{alt}(t_{cd} + \Delta t) = alt(t_{cd}) + \widehat{vs}(t_{cd}) * \Delta t \tag{11}$$

The trajectory prediction module computes a discretized trajectory estimate, $\widehat{T}_i(t)$, for each aircraft i in the simulation, using a fixed time interval, given by Δt_{traj}, from the current conflict detection cycle time, t_{cd}, to the conflict detection horizon time, given by $t_{cd} + T_{pred}$. A simple state-projection is used for the results presented in this paper.

At each conflict detection cycle, the conflict detection module compares the actual trajectories of aircraft to determine true LOS events, and compares the predicted trajectories of aircraft to determine predicted LOS events, within a specified time horizon. A true conflict, $C_{m,i,j}$, exists at conflict detection time, t_{cd}, between aircraft i and j if there exists some time, t, within the time horizon ($t_{cd} < t \le t_{cd} + T_{pred}$), where the true trajectories, $T_i(t)$ and $T_j(t)$, have a LOS (the horizontal distance between the aircraft trajectories is less than the

Figure 1. Position and velocity error components for surveillance modeling.

horizontal separation criterion, H_{sep}, and the vertical distance is less than the vertical separation criterion, V_{sep}, simultaneously). Aircraft i and j are said to be in (or to have a) true conflict at time t_{cd} if such a true LOS event exists. The LOS event from a predicted conflict, $\widehat{C}_{n,i,j}$, follows the same definition as for a true LOS event when comparing the estimated or predicted trajectory pair $\widehat{T}_i(t)$ and $\widehat{T}_j(t)$. Similarly, aircraft i and j are said to be in (or to have a) predicted conflict at time t_{cd} if such a predicted LOS event exists. Note that, in this analysis, the conflict detection is done over a discretized trajectory where $t := k * \Delta t_{traj}$ and $k \in \{1, 2, \ldots, T_{pred}/\Delta t_{traj}\}$. A surveillance range parameter, R_s, is used to limit the pairwise conflict detection to aircraft that are within range, where infinite range would be equivalent to the full N² conflict detection problem. The conflict detection module also captures the statistics regarding conflict detection performance via the metrics described in section C.

American Institute of Aeronautics and Astronautics

C. Metrics Definitions

The primary metrics used to characterize conflict detection performance in this paper are: false alerts, missed alerts, missed detections, and time-to entry into loss-of-separation at first detection for all true LOS. Note the distinction between an alert and a detection (see Table 1), where an alert refers to a warning that is either issued or not issued (for missed alerts) for a conflict at any point in time, while a detection refers to the overall successful detection of a LOS event over an entire simulation run. The traditional definition of a false alert, as it is most commonly found in other examples of conflict detection-related literature, is when *there exists a predicted conflict* between the predicted trajectories of two aircraft but *there does not exist a true conflict* between the true trajectories of those same two aircraft, within the same time horizon. Similarly, the traditional definition of a missed alert is when *there does not exist a predicted conflict* between the predicted trajectories of two aircraft but *there does exist a true conflict* between the true trajectories of those same two aircraft, within the same time horizon. As will be shown next, however, the definitions should be more rigorous than these, especially when working with state-projected trajectory predictions.

The false alert and missed alert definitions need to contain criteria to evaluate whether a predicted conflict and a true conflict are "the same;" that is, the LOS occurs roughly in the same place and at the same time. Consider the schematic in Figure 2 where two aircraft are shown to be in predicted conflict *and* in true conflict. The traditional definitions for false alert and missed alert would indicate that there is no false alert and no missed alert; that is, this example would be considered a correct alert due to the presence of both a predicted and a true conflict. However, the scale of the diagram in Figure 2 might be large and the location of the LOS in the two conflicts may be many nautical miles apart. This difference in location is important for functions such as conflict resolution where the difference between a conflict ahead-of-track and a conflict to the right-of-track is important. Using the traditional definitions for false alert and missed alert may produce an artificially better conflict detection performance. This is because, if some criteria (yet to be defined) for evaluating whether two conflicts are "the same" is applied to the diagram of Figure 2, either of two conditions will occur: the two conflicts are considered to be the same and the false alert and missed alerts are both zero, just as the traditional definitions indicate, or the two conflict are not the same,

Figure 2. Example predicted and actual conflicts between a pair of aircraft.

in which case the predicted conflict is considered to be a false alert to a non-existent LOS, and the true conflict is considered to be a missed alert of a true LOS. Therefore, using a criteria for evaluating whether two conflicts are the same within each conflict detection cycle,

- a false alert is a *predicted conflict* in the predicted trajectories of two aircraft where *the same true conflict* in the true trajectories of those aircraft *does not exist*, within the conflict detection time horizon, and
- a missed alert is a *true conflict* in the true trajectories of two aircraft where *the same predicted conflict* in the predicted trajectories of those aircraft *does not exist*, within the conflict detection time horizon.

There are many definitions that could be used to deem two conflicts to be the same. A LOS is an event that has a time extent where there exists a time-in and a time-out of LOS and those times could be used for comparison. However, that comparison would fail for an example such as the one in Figure 2 where the temporal location of the LOS for the predicted and true LOS events may be very close, even though the spatial location of the two LOS is much different. Another comparison method would be to use the location of the closest-point-of-approach – where the separation between the two trajectories is the smallest, either vertical or horizontal – of either of the two aircraft, but this approach is problematic because, for a shallow LOS event where the two aircraft trajectories are on nearly parallel tracks, the closest-point-of-approach may be many minutes after entering into LOS, which may not be very useful. Because one of the primary metrics for conflict detection performance is the time-to loss-of-separation, Δt_{LOS}, at first detection, the clear choice for defining a conflict vector is using the time-in to loss-of-separation, combined with the spatial location of the conflict, where the location is described by the positions of the two aircraft involved at the entry into loss-of-separation. Define a conflict vector, $C_{m,i,j}$, between aircraft i and aircraft j as:
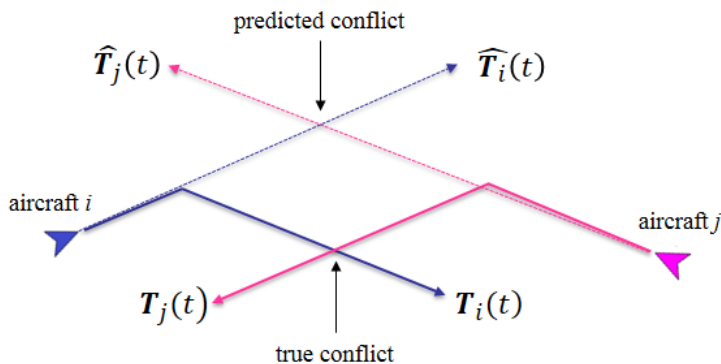
$$C_{m,i,j} := \left[t_{LOS,m}, s_{i,m}(t_{LOS,m}), s_{j,m}(t_{LOS,m}) \right] \tag{12}$$

where the first element is the time-in to LOS, $t_{LOS,m}$, for conflict m, the next 3 elements are the position of aircraft i at $t_{LOS,m}$, and the last 3 elements are the position of aircraft j at $t_{LOS,m}$. Figure 3 shows the estimated positions of aircraft i and j at the time-in to LOS, $t_{LOS,m}$, of a predicted conflict m, and the actual positions of those aircraft at the time-in to LOS, $t_{LOS,n}$, of a true conflict n.
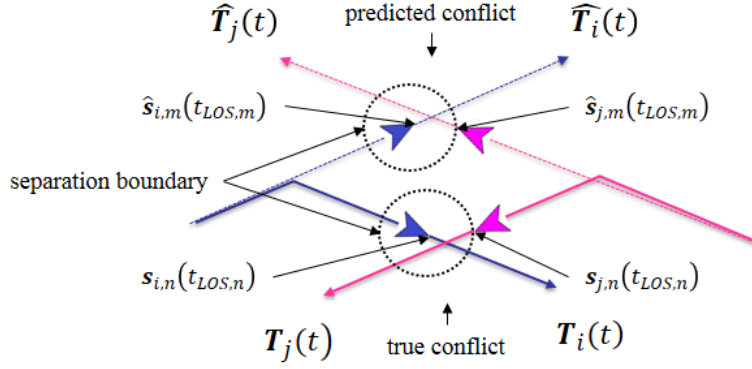
Figure 3. Estimated aircraft positions at time-in to LOS, $t_{LOS,m}$, for a predicted conflict, m, and true aircraft positions at, $t_{LOS,n}$, for a true conflict, n.

Comparison of two conflicts is done by evaluating the space/time differences between the conflict vectors against a set of threshold values. Let,

$$D_{H,i,m,n} = d_{GCD}\left(lat_{i,m}(t_{LOS,m}), lon_{i,m}(t_{LOS,m}), lat_{i,n}(t_{LOS,n}), lon_{i,n}(t_{LOS,n}) \right) \tag{13}$$

be the horizontal arc-distance between the positions of aircraft i in conflicts m and n, where d_{GCD} is a function that computes the great-circle distance between two points on a sphere. Let,

$$D_{V,i,m,n} = \left| alt_{i,m}(t_{LOS,m}) - alt_{i,n}(t_{LOS,n}) \right| \tag{14}$$

be the magnitude of the difference between the altitude of aircraft i in conflict m, and the altitude of aircraft i in conflict n. Then, two conflicts m and n are said to be _the same_ ($C_{m,i,j} \approx C_{m,i,j}$) if and only if _all_ of the conditions in equations (15)-(19) are satisfied:

$$\left| t_{LOS,m} - t_{LOS,n} \right| \leq \delta_t \tag{15}$$

$$D_{H,i,m,n} \leq \delta_H \tag{16}$$

$$D_{H,j,m,n} \leq \delta_H \tag{17}$$

$$D_{V,i,m,n} \leq \delta_V \tag{18}$$

$$D_{V,j,m,n} \leq \delta_V \tag{19}$$

where δ_t, δ_H, and δ_V, are threshold values selected for the maximum allowable time difference between two LOS entry times, the maximum allowable horizontal distance between the position of each aircraft in both conflicts, and maximum allowable vertical distance between the altitudes of each aircraft in both conflicts, respectively. This novel conflict comparison approach led to the following logic being used to determine a set of reasonable threshold values, based on a lateral separation standard of 5 NM and a vertical separation standard of 1000 ft:

- $\delta_t = 120 \text{ s}$: the time it takes to traverse 10 NM at an assumed groundspeed of 300 knots
- $\delta_H = 10 \text{ NM}$: the horizontal position of an aircraft i can vary by as much as 10 NM and still be in conflict with an aircraft j
- $\delta_V = 2000 \text{ ft}$: the vertical position of an aircraft i can vary by as much as 2000 ft and still be in conflict with an aircraft j

The comparison criteria defined by equations (15)-(19) leads directly to the computation of the primary metrics. At any conflict detection time, a false alert is identified when a predicted conflict is not the same as any true conflict within the same conflict detection horizon. Similarly, at any conflict detection time, a missed alert is identified when a true conflict is not the same as any conflict predicted within the same conflict detection horizon. The sum total of all false and missed alerts are collected over the entire simulation run. A missed detection is identified by a true LOS that was not predicted in any conflict detection cycle over an entire simulation run. The time-to entry into loss-of-separation at first detection, $\Delta t_{LOS,m}$, for any true conflict, m, is given by:

$$\Delta t_{LOS,m} = t_{LOS,m} - t_{cd} \qquad (20)$$

and is recorded at the earliest detection cycle in which a correct alert (a predicted conflict is the same as the true conflict) is issued for that true conflict.

The metrics are reported in terms of absolute counts, ratios or probabilities, mean values, and can also be presented in terms of distributions. Table 3 lists the full set of conflict detection metrics, including those that are computed for the sole purpose of denominators in the probability value computations. In each of the definitions, the true and predicted conflict sets, C and $\widehat{C}$, respectively, contain all conflicts within the time horizon of a single detection cycle when summing over all detection cycles, and all LOS over an entire simulation run otherwise. Note that the observed probability metrics are subject to the conditions of each simulation run. The observed probability of false alert metric indicates the likelihood that any alert provided by the conflict detector will be false. The observed probability of missed alert indicates the likelihood that, at any detection cycle, there is a true LOS present within the conflict detection horizon that was not alerted. Finally, the observed probability of missed detection indicates the likelihood that the conflict detector will fail to provide any alert for a true LOS. The metrics associated with false alerts and missed alerts provide a measure of the confidence of the conflict alerting, whereas the metrics associated with missed detections provides a measure of the overall performance of the conflict detector under the conditions simulated.

Table 3. Conflict detection performance metrics.

Metric	Definition
$N_{FA} = \sum\limits_{sim\_run} \sum\limits_{t_{cd}} \|\widehat{C} \setminus C\|$	the **number of false alerts** from the set of predicted conflicts (conflict alerts), $\widehat{C}$, identified during each detection cycle, summed over every detection cycle and over an entire simulation run
$N_{\widehat{C}} = \sum\limits_{sim\_run} \sum\limits_{t_{cd}} \|\widehat{C}\|$	the **number of predicted conflicts** (conflict alerts), $\widehat{C}$, identified during each detection cycle, summed over every detection cycle and over an entire simulation run
$N_{MA} = \sum\limits_{sim\_run} \sum\limits_{t_{cd}} \|C \setminus \widehat{C}\|$	the **number of missed alerts** from the set of true conflicts, C, within the time horizon of each detection cycle, summed over every detection cycle and over an entire simulation run
$N_C = \sum\limits_{sim\_run} \sum\limits_{t_{cd}} \|C\|$	the **number of true conflicts** present during each conflict detection cycle, summed over every detection cycle and over an entire simulation run
$N_{C,sim} = \|C\|$	the number of unique true conflicts (**number of LOS**) in a simulation run
$\Delta t_{LOS,mean} = \sum\limits_{m=1,2,\ldots}^{M} \dfrac{\Delta t_{LOS,m}}{M}$	**mean time-to loss-of-separation at first detection** for all M successfully detected true conflicts (missed detections not included)
$N_{MD} = \|C \setminus \widehat{C}\|$	the **number of missed detections** – from the set of all true conflicts, C, not in the set of predicted conflicts (conflict alerts), $\widehat{C}$, over an entire simulation run
$P_{FA} = N_{FA}/N_{\widehat{C}}$	the false alert ratio or **false alert probability**
$P_{MA} = N_{MA}/N_C$	the missed alert ratio or **missed alert probability**
$P_{MD} = N_{MD}/N_{C,sim}$	the missed detection ratio or **missed detection probability**

D. Data Pre-processing
The analysis methodology described in this paper leverages a set of recorded traffic tracks from the NAS. The source data is a fused, NAS-wide data feed from Exelis that covers 45 days in 2013 and is stored in a Data Warehouse at the NASA Ames Research Center. Surveillance radar and Automatic Dependent Surveillance-Broadcast (ADS-B) sources are fused to create full visual- and instrument-flights-rules (VFR and IFR) track profiles

(latitude, longitude, altitude, and time) for flights within the Continental United States (CONUS), and partial track profiles for flights entering and exiting the NAS.

A representative day of traffic was chosen from the Exelis data set for use in this study. Figure 4 shows the Federal Aviation Administration's Aviation System Performance Metrics (ASPM77)[22] database values for total daily operations and total daily delayed operations for one portion of the available data days. The representative day of traffic was chosen to be March 28th, 2013 due to the high demand and low level of delayed operations, likely indicating a day with low weather impacts on the flown tracks. A 36-hour window was selected to capture 6 hours of traffic prior to and 6 hours of traffic after the selected date. This allows for the inclusion of the complete track profile for flights that were already airborne before the start of the selected date and those that were still airborne at the conclusion of the selected date.
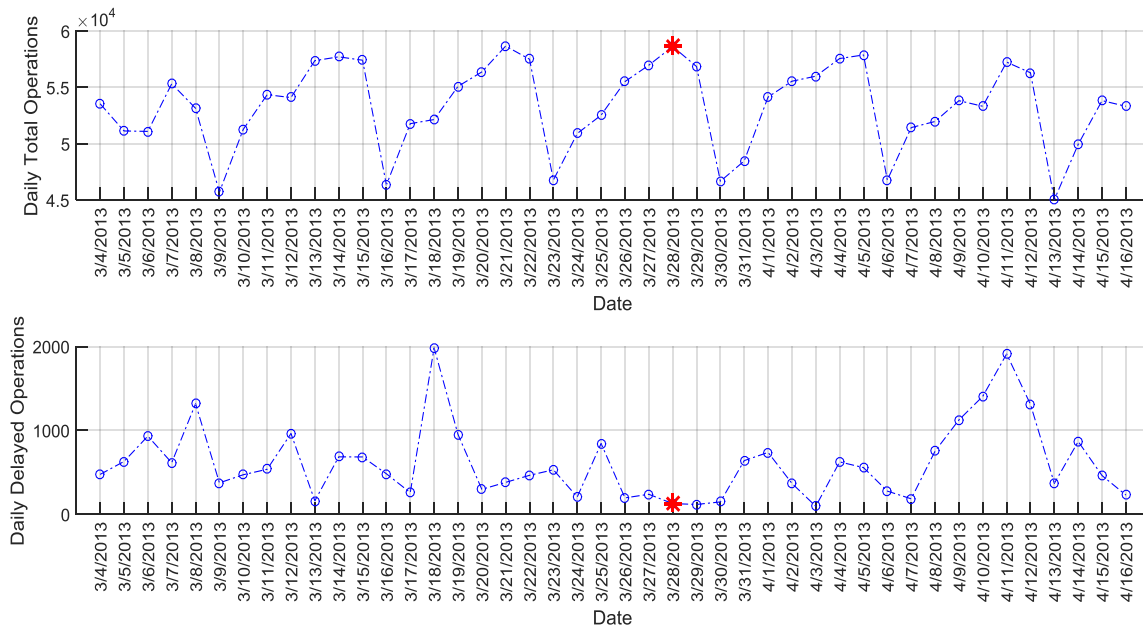
Figure 4. ASPM77 daily total operations (top), and ASPM daily delayed operations (bottom). March 28, 2013 was chosen as a high volume and low delay day within the recorded traffic data available.

The raw Exelis track data required pre-conditioning and filtering before it could be used to conduct this analysis. The data inherently contains the surveillance noise from the surveillance sensors that captured the data. Additionally, the fusion of multiple data sources can introduce an artificial noise due to the various sampling frequencies and timestamps. Figure 5 and Figure 6 show the track and altitude profiles, respectively, for a randomly
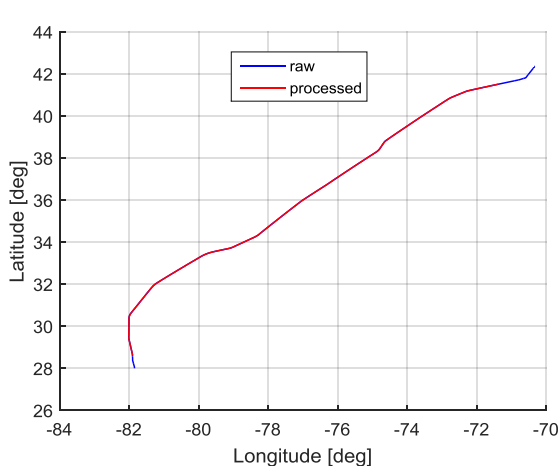
Figure 5. Track profile for sample flight (KTPA-KBOS).
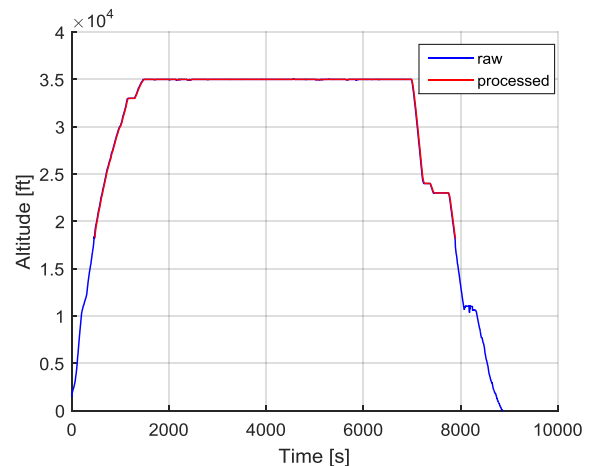
Figure 6. Altitude profile for sample flight (KTPA-KBOS).

10

selected flight. Note that, using these figures alone, it is difficult to spot the noisy characteristics of the data. However, an inspection of the velocity profiles of this example flight taken from simple differential derivatives reveals the noise characteristics. Figure 7-Figure 9 show the groundspeed, vertical speed, and track angle profiles, respectively, for the same sample flight. Note the high level of noise in these velocity profiles. In each of the figures for the sample flight, the raw track data is shown in blue and the final processed track data for use in this analysis is shown in red. The processing of the track data was completed in three basic steps. First, a pre-screening step was used to discard flights that did not have sufficient data useful for this analysis. This included discarding flights with no track data above flight level (FL) 180, discarding flights fully outside the time window of interest, and discarding flights with anomalous track data. Next, a conditioning step was used to correct any data drop-outs and to trim the flight track data to the portion above FL180. Flight segments below FL180 include VFR flights with altitude separations in 500 foot increments and traffic entering or exiting the terminal airspace environment where the separation standards are different from the en-route environment and special separation rules exist[23]; these conditions were observed in a few test runs of the analysis model with the un-time-shifted track data. Finally, a filtering step was used to remove the surveillance and data fusion noise in order to create a baseline set of track data. A combination of excessive groundspeed and vertical speed data point removal, followed by a one-dimensional Kalman Filter, generated the final processed track data that can be seen in Figure 5-Figure 9. The thresholds and algorithms used for the processing of the raw data were developed largely by observations drawn from the raw data. Here is a list of the pre-processing and filtering steps that were applied to the raw data:

Pre-screening:
- Discard all flights with no data points above FL180
- Discard all flights with the same origin and destination airport
- Discard flights with un-identified origin or destination airport
- Discard any flights already airborne at the beginning of the 36 hour sample window
- Discard any flights still airborne at the end of the 36 hour sample window
- Discard any flights with excessive data gaps in time (greater than 5 minutes)
- Discard any flights with a processed track time less than 10 minutes
- Discard any flights with incomplete track data (missing descent or climb portion but fully within the CONUS)
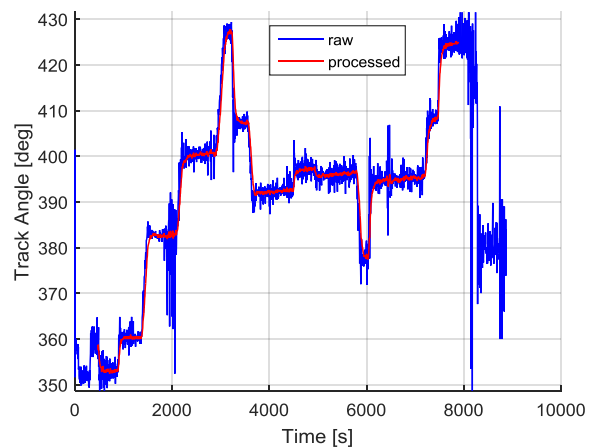
Conditioning:
- Trim all track data to retain only the portion above FL180
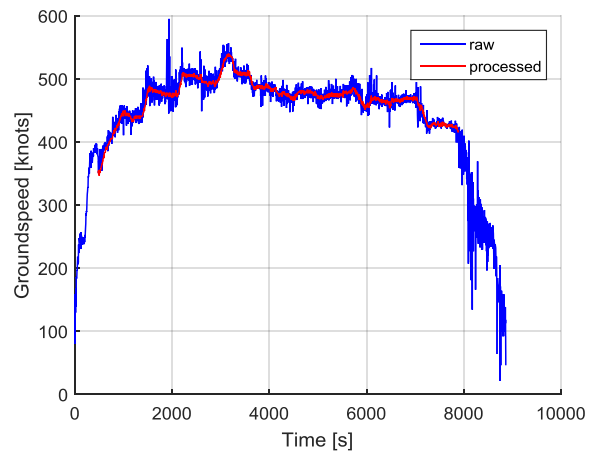
Figure 7. Groundspeed profile for sample flight (KTPA-KBOS).
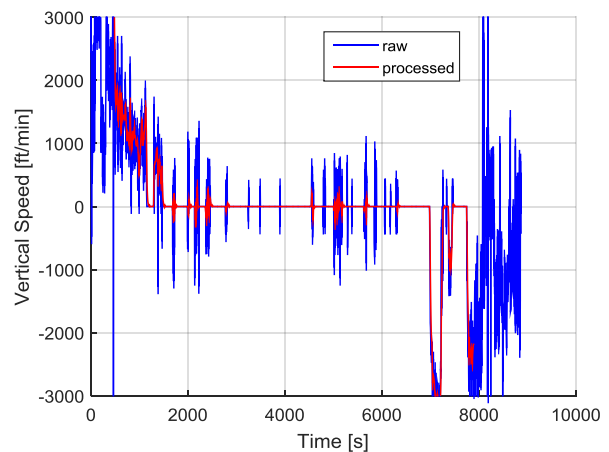
Figure 8. Vertical speed profile for sample flight (KTPA-KBOS).

Figure 9. Track angle profile for sample flight (KTPA-KBOS).

11

- Remove any duplicated time data points
- Remove any data drop-outs (identified via vertical speeds in excess of 10,000 ft/min)

Filtering:
- Recursively remove any data points producing excessive groundspeed and vertical speed
  - Groundspeeds greater than 590 knots
  - Vertical speeds greater than 10000 ft/min
- Recursively remove any data points with large groundspeed and vertical speed deviations away from a 9-point centered moving average
  - Groundspeed differences larger than 100 knots
  - Vertical speed differences larger than 100 ft/min
- Interpolate the track data for a uniform time sampling of 5 seconds
- Apply a one-dimensional Kalman Filter[24] to the track data (altitude, latitude, and longitude)

The selected traffic demand day of March 28th, 2013, with the additional 6 hours of pre- and 6 hours of post-traffic, contained a total of 72,769 recorded traffic tracks. After the processing steps above were implemented, the remaining traffic set contained 30,779 recorded traffic tracks.

E. Scenario Generation

Simulation scenarios for this analysis were created by time-shifting the recorded traffic data in order to generate losses-of-separation that were not present in the un-shifted tracks. The objective was to create as many LOS events as possible. A similar approach was used by Paglione and Bilimoria.[21] Density and complexity of a traffic scenario is not important for conflict detection because conflict detection is a pair-wise function; regardless of any background traffic, a conflict detector has the same likelihood of successfully detecting/alerting a conflict for the same pair of aircraft, in the absence of resolution. Note that, even though scenario complexity is not a primary factor in conflict detection performance, trajectory complexity, on the other hand, is an important factor and these two measures can have some correlation when working with recorded data. The density of traffic in a scenario can be increased if more conflicts are needed beyond those created by time-shifting the original traffic demand, but this was not necessary in this preliminary set of runs. The time-shifting of the traffic tracks was done by adding a random time-shift to each track's original start time, sampled from a uniform distribution, $U(-t_{shift,max}, t_{shift,max})$.

IV. Results

The results presented here encompass a preliminary set of analysis runs to investigate the correctness of the observed trends and the sensitivity of the metrics to the input parameters. In each of these runs, only a single parameter was varied to determine the impact on the metrics, while a nominal set of parameters was held constant for the parameters not under investigation. Table 4 lists the nominal set of input parameters that were used in these preliminary runs. Notice that the minimum vertical separation criterion used is less than the minimum separation standard of 1000 feet. The criterion of 800 feet is used instead because the recorded traffic data is subject to the Mode C resolution of 100 feet in some instances, which causes a high number of conflicts in the un-shifted track data that are not real conflicts.

The first set of analysis runs was conducted to understand the impact of various time-shifts of the original schedule on the number of LOS created. Ten different scenarios using the full recorded traffic data set were generated with a maximum time shift parameter, $t_{shift,max}$, of zero, two, five, and fifteen minutes, and one, two, four, six, twelve, and twenty-four hours and with the nominal model parameter values. These runs helped to identify the time shift that produces the highest number of true losses-of-separation. As seen in Figure 10, the peak number of true LOS is achieved with a time shift around fifteen minutes to one hour. At these time shift values, the full day of time-shifted traffic produces over 15,000 true LOS. The increase followed by a decrease in the number of true LOS is a result of the small increase in time shift values that

Table 4. Nominal parameter values.

Parameter	Value
T_{det}	60 s
T_{pred}	1200 s
Δt_{traj}	5 s
H_{sep}	5 NM
V_{sep}	800 ft
R_s	1500 NM
t_{sl}	0 s
σ_r	0 NM
σ_a	0 ft
σ_{gs}	0 kn
σ_{vs}	0 ft/min
σ_{trk}	0 deg
δ_t	120 s
δ_H	10 NM
δ_V	2000 ft

increasingly create LOS events in the recorded tracks until such point as the peaks of traffic start to flattens out and the density of traffic is reduced, as shown by the number of simulated traffic over the simulation time in Figure 11.
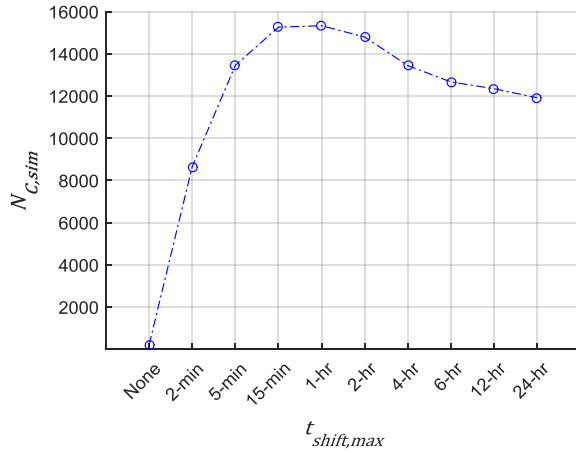
Figure 10. Number of true LOS, $N_{C,sim}$, created as a function of the maximum time shift parameter, $t_{shift,max}$.
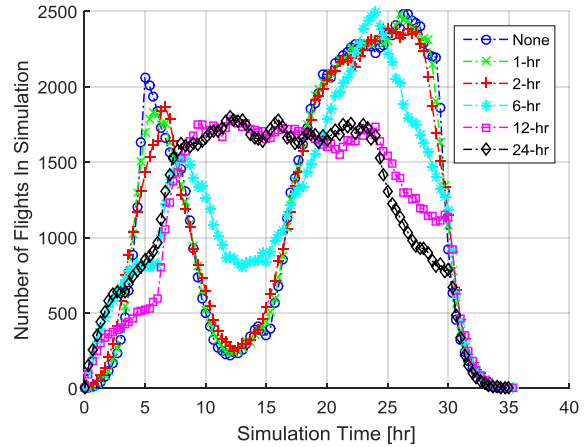
Figure 11. Number of flights as a function of the simulation clock time for a few of the time shift scenarios.

The next set of analysis runs were conducted to show the variability in the metrics from different randomizations of the same maximum time shift. Ten different random scenarios of the full recorded traffic data set were created using the same maximum time shift of one hour. Table 5 shows the statistics for the metrics for these ten different scenario runs. The results indicate that the variability is small with less than 2% in standard deviation for most of the metrics, while the number and probability of missed detections had a higher standard deviation (under 7%). This small variability indicates that the selection of a single scenario from amongst these 10 randomizations as a baseline scenario is sufficient for investigating the trends in the analysis metrics. For the remaining analysis runs, one of these 1-hour time shift randomized scenarios was used as a baseline for comparison.

Table 5. Conflict detection performance metrics* for ten randomization scenarios with a 1-hour time shift.

Metric	Mean	Min	Max	Standard Deviation (%)
N_{FA}	185,815	184,222	188,468	1,325 (0.71%)
$N_{\hat{C}}$	249,582	246,964	252,128	1,781 (0.71%)
N_{MA}	176,945	171,280	182,040	3,285 (1.86%)
N_C	240,712	234,952	247,688	3,923 (1.63%)
$N_{C,sim}$	15,604	15,274	15,988	205 (1.31%)
$\Delta t_{LOS,mean}$	289	285	293	2.9 (1.04%)
N_{MD}	1,012	878	1,092	70 (6.92%)
P_{FA}	0.745	0.740	0.748	0.003 (0.40%)
P_{MA}	0.735	0.729	0.738	0.003 (0.41%)
P_{MD}	0.065	0.057	0.070	0.004 (6.15%)

* Refer to Table 3 or the Nomenclature for metrics definitions.

The baseline performance of the conflict detection function using the nominal set of parameters outlined in Table 4 is relatively poor, with false alert and missed alert probabilities of 74.5% and 73.5%, respectively, and a missed detection probability of 6.5%. The high false and missed alerts are primarily due to the state-only trajectory projection method used over a relatively long conflict detection time horizon; analysis runs with shorter conflict detection time horizon presented in this section show improved performance.
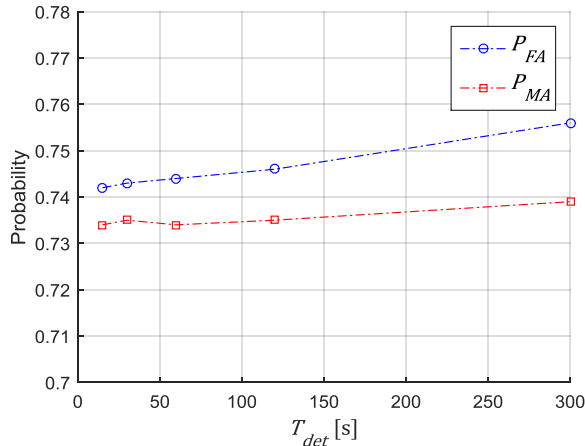
A nominal conflict detection cycle period, T_{det}, of 60 seconds was chosen based on the time required to produce one analysis run. Nonetheless, it was important to understand the impact of this parameter on the CD performance metrics. The 1-hour time shift baseline scenario was run using the nominal parameter values of Table 4 but with a conflict detection cycle period of 15, 30, 60, 120, and 300 seconds. Table 6 shows the conflict detection performance metrics for each of these runs. As the conflict detection cycle period is increased, the number of alerts decreases proportionally with the decrease in the number of conflict detection cycles within the same simulation

time. The observed false alert and missed alert probabilities, as shown in Figure 12, are nearly un-changed with a 1.4% spread in false alert probability and a 0.5% spread in missed alert probability. However, there is a significant impact on the observed missed detection probability, as shown in Figure 13, with an increase from 0.4% at a CD cycle period of 15 seconds, to more than 45% with a CD cycle period of 5 minutes. The penalty for using a CD cycle period of 60 seconds as the nominal parameter for a baseline scenario is a missed detection probability around ~7%; this missed detection rate in a baseline run is better than one very close to zero because it will allow us to better demonstrate positive impacts to performance.

Table 6. Conflict detection performance metrics* for various conflict detection cycle periods.

Metric	Conflict Detection Cycle Period, T_{det}, [s]				
	15	**30**	**60**	**120**	**300**
N_{FA}	731,088	366,160	184,222	93,230	38,544
$N_{\hat{c}}$	984,686	492,828	247,630	124,892	50,992
N_{MA}	700,628	350,442	175,182	87,648	35,270
N_C	954,226	477,110	238,590	119,310	47,718
$N_{C,sim}$	15,764	15,714	15,574	15,368	14,780
$\Delta t_{LOS,mean}$	328	305	287	280	298
N_{MD}	56	252	988	2,908	6,684
P_{FA}	0.742	0.743	0.744	0.746	0.756
P_{MA}	0.734	0.735	0.734	0.735	0.739
P_{MD}	0.004	0.016	0.063	0.189	0.452

* Refer to Table 3 or the Nomenclature for metrics definitions.

Figure 12. Observed false alert and missed alert probabilities, P_{FA} and P_{MA}, as a function of the conflict detection cycle period, T_{det}.
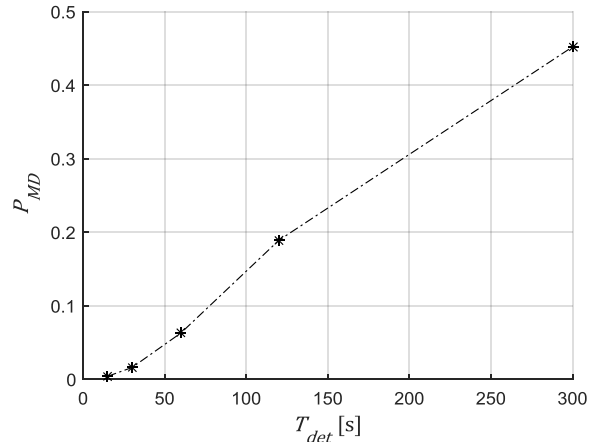
Figure 13. Observed missed detection probability, P_{MD}, as a function of the conflict detection cycle period, T_{det}.

The conflict detection time horizon should have a significant impact on the conflict detection performance metrics. A shorter detection and prediction horizon has a positive impact on the trajectory prediction quality for a state-only predictor because the prediction is less likely to miss an intended trajectory change. The 1-hour time shift baseline scenario was run using the nominal parameter values of Table 4 but with conflict detection time horizons, T_{pred}, of {1,2,3,4,5,6,8,10,12,14,16,18,20} minutes. Table 7 shows the resulting conflict detection performance metrics. Regardless of the conflict detection time horizon, the baseline scenario contains 15,574 true LOS. Note that, for the case where the CD time horizon is one minute, the number of true conflicts present during each conflict detection cycle, N_C, is exactly equal to the total number of true LOS in the simulation run, $N_{C,sim}$, and the missed alert probability is equal to the missed detection probability. This occurs because the CD cycle period is equal to the CD time horizon and each LOS is only present for detection during one detection cycle; therefore, in this case, a missed alert is equivalently a missed detection.
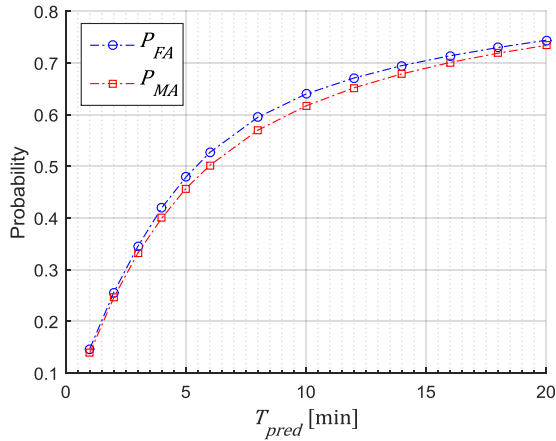
Larger conflict detection time horizons have an increased probability of both false alerts and missed alerts but a decreased probability of missed detections. As the detection time horizon is increased, the number of conflict alerts,

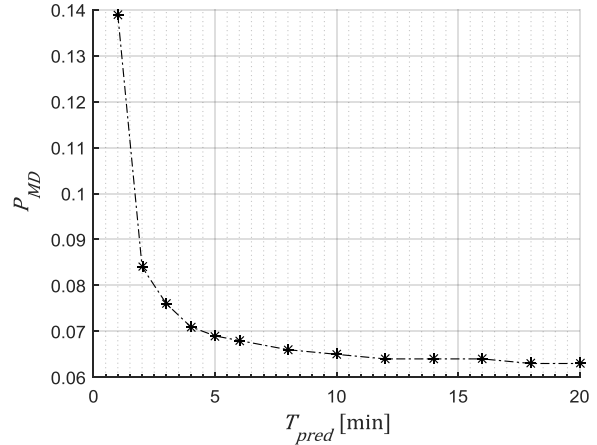**Table 7. Conflict detection performance metrics* for various conflict detection time horizons.**

Metric	Conflict Detection Time Horizon, T_{pred}, [min]												
	1	**2**	**3**	**4**	**5**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**
N_{FA}	2,304	7,990	15,998	25,948	36,866	48,608	71,842	94,238	114,418	133,200	150,864	167,794	184,222
$N_{\hat{C}}$	15,720	31,166	46,398	61,846	77,030	92,218	120,824	147,236	170,482	191,710	211,398	229,916	247,630
N_{MA}	2,158	7,550	15,082	23,972	33,748	44,004	64,860	85,408	105,108	123,878	141,690	158,774	175,182
N_C	15,574	30,726	45,482	59,870	73,912	87,614	113,842	138,406	161,172	182,388	202,224	220,896	238,590
$N_{C,sim}$	15,574	15,574	15,574	15,574	15,574	15,574	15,574	15,574	15,574	15,574	15,574	15,574	15,574
$\Delta t_{LOS,mean}$	30	70	102	127	149	166	195	218	236	251	265	277	287
N_{MD}	2,158	1,316	1,176	1,108	1,080	1,056	1,026	1,008	1,002	998	990	988	988
P_{FA}	0.147	0.256	0.345	0.420	0.479	0.527	0.595	0.640	0.671	0.695	0.714	0.730	0.744
P_{MA}	0.139	0.246	0.332	0.400	0.457	0.502	0.570	0.617	0.652	0.679	0.701	0.719	0.734
P_{MD}	0.139	0.084	0.076	0.071	0.069	0.068	0.066	0.065	0.064	0.064	0.064	0.063	0.063

* Refer to Table 3 or the Nomenclature for metrics definitions.

$N_{\hat{C}}$, increases at a slower rate than the number of false alerts, causing an increase in the false alert probability. A similar trend is observed for the missed alert probability as shown in Figure 14. This is the expected behavior of conflict detection based on a state-projection trajectory prediction that does not take into account trajectory changes within the same time horizon. The missed detection probability shows a sharply decreasing trend for small values of CD time horizon, but quickly settles to a value below 7% probability with CD time horizon of five minutes or greater, as seen in Figure 15. The mean time-to LOS at first detection for the successfully detected conflicts, shown in Figure 16, also has an increasing trend due to the fact that the longer time horizons provide the detector the opportunity to identify LOS earlier. Consequently, a trade-off can be made to keep false alerts and missed alerts low while decreasing the missed detections and increasing the mean time-to LOS in the process of selecting the best conflict detection time horizon under the conditions simulated in this scenario run.
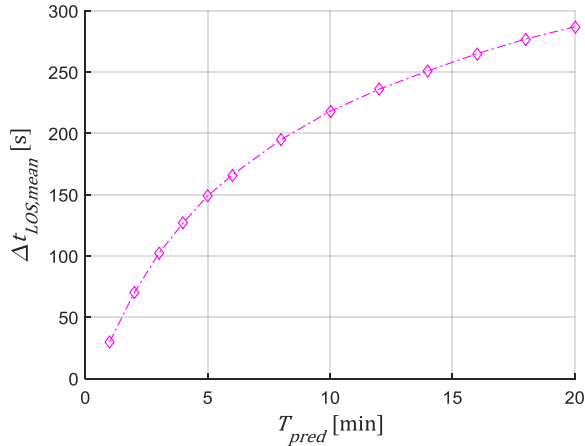

Figure 14. Observed false alert and missed alert probabilities, P_{FA} and P_{MA}, as a function of the conflict detection time horizon, T_{pred}.


Figure 15. Observed missed detection probability, P_{MD}, as a function of the conflict detection time horizon, T_{pred}.

As a better indication of the poor performance of conflict detection with state-only trajectory prediction, we can look at the probability of false alerts and the probability of missed alerts as a function of the time-to LOS. Figure 17 shows those probability distributions for the baseline scenario with a 20 minute CD time horizon and 60 second CD cycle period (same run as the last column of Table 7). This figure clearly shows that the conflict alerts that report a large time-to LOS have a high probability of being false; in fact, conflict alerts with a time-to LOS greater than approximately 2.5 minutes have a higher than 50% probability of being false alerts and that probability increases to greater than 80% at just over 6 minutes reported time-to LOS. Similarly, missed alerts are more likely for conflicts that are farther out in the CD horizon. Because false alerts are typically characterized as "nuisance alerts" that can cause distrust in alerting algorithms where humans are involved, and because false alerts can add unnecessary workload to functions that leverage the conflict detection results (e.g., conflict resolution), it would be ill-advised to implement a state-projection trajectory prediction methodology and CD algorithm with a long detection time horizon.
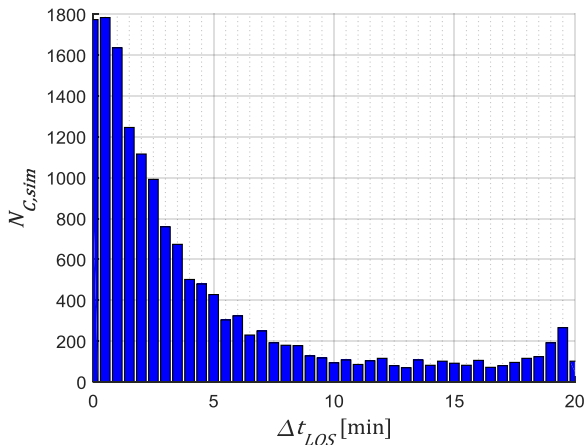
Figure 16. Mean time-to LOS at first detection, $\Delta t_{LOS,mean}$, as a function of the conflict detection time horizon, T_{pred}.
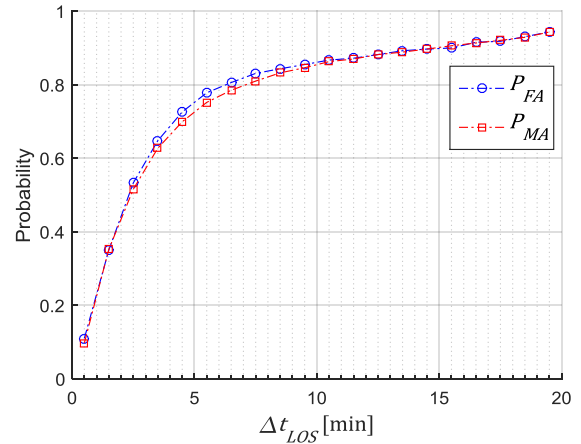

Figure 17. False alert and missed alert probabilities, P_{FA} and P_{MA}, as a function of the alert's time-to LOS, Δt_{LOS}.

The distribution of the detected true LOS versus the time-to LOS at first detection does indicate that some conflicts were successfully detected and alerted with a time-to LOS up to the full CD time horizon of 20 minutes, as seen in Figure 18. However, the distribution indicates that the bulk of the LOS were detected with less than 5 minutes time-to LOS. This is verified in Figure 19 which shows the percent of conflicts detected as a function of a minimum time-to LOS; only 28% percent of detected LOS events provided 5 minutes or more of time-to LOS alerting time. Note that the intercept at a minimum time-to LOS of zero in Figure 19 (93.7%) indicates the overall probability of successful detection for this scenario, which is equal to one minus the probability of missed detections.

Figure 18. Distribution of detected true LOS, $N_{C,sim}$, versus the time-to LOS at first detection, Δt_{LOS}.
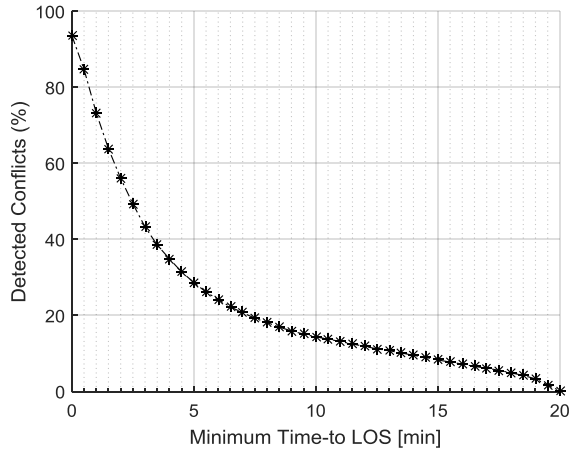
Figure 19. Percent detected conflicts versus minimum alert time-to LOS.

Preliminary analysis runs were also conducted to investigate the sensitivity of the CD performance metrics to the surveillance error model parameters. Each of the six sensitivity parameters was varied individually in a set of six analysis runs, where only the single parameter under investigation was varied while the remaining model parameters retained their nominal values. Table 8 shows the six surveillance model parameters whose values were selected arbitrarily. The purpose of these analysis runs was simply to observe the trend in the primary metrics. Table 9 shows the CD performance metrics for the baseline run and the six surveillance error parameter variations.

Table 8. Surveillance error parameter values for individual sensitivity runs.

| Parameter	Value
t_{sl}	15 s
σ_r	0.25 NM
σ_a	100 ft
σ_{gs}	20 kn
σ_{vs}	200 ft/min
σ_{trk}	1 deg

Figure 20 shows the variation of the false alert and missed alert probabilities for the six surveillance error model parameters as compared to the baseline, while Figure 21 shows the same comparison for the missed detection probability.

The CD performance metrics indicate an appropriate direction for the sensitivity relative to the surveillance error model parameters. That is, any amount of non-zero surveillance error in the model parameters has a negative impact on CD performance, with higher false alerts, missed alert, and missed detections. Note that the missed detection probability for the surveillance error parameter in the vertical speed, σ_{vs}, is the only primary metric that shows an un-expected sensitivity with a decrease in the likelihood of missed detections with increasing vertical speed error. This could be due to the high variability of this parameter to random variations that was identified in Table 5; further investigation would be needed to characterize the uncertainty in the performance metrics due to each of the surveillance error parameters.
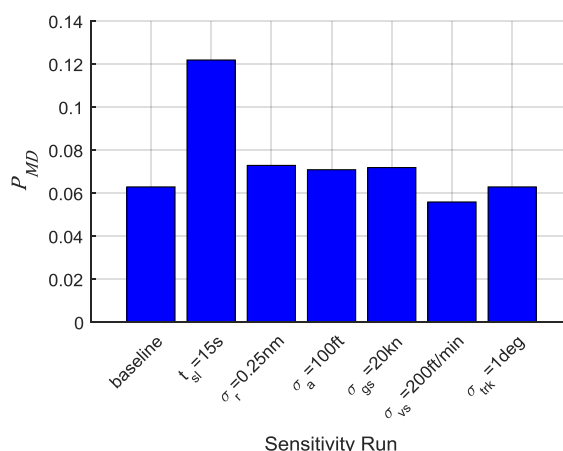
Table 9. Conflict detection performance metrics* for single surveillance error parameter runs.

Metric	Sensitivity Parameter Run						
	baseline	$t_{sl} = 15\ s$	$\sigma_r = 0.25\ NM$	$\sigma_a = 100\ ft$	$\sigma_{gs} = 20\ kn$	$\sigma_{vs} = 200\ ft/min$	$\sigma_{trk} = 1\ deg$
N_{FA}	184,222	192,792	185,298	217,890	200,014	333,770	183,832
$N_{\hat{C}}$	247,630	252,202	247,728	281,032	254,556	378,124	245,342
N_{MA}	175,182	179,180	176,160	175,448	184,048	194,236	177,080
N_C	238,590	238,590	238,590	238,590	238,590	238,590	238,590
$N_{C,sim}$	15,574	15,574	15,574	15,574	15,574	15,574	15,574
$\Delta t_{LOS,mean}$	287	289	289	295	277	254	286
N_{MD}	988	1,900	1,134	1,104	1,124	866	988
P_{FA}	0.744	0.764	0.748	0.775	0.786	0.883	0.749
P_{MA}	0.734	0.751	0.738	0.735	0.771	0.814	0.742
P_{MD}	0.063	0.122	0.073	0.071	0.072	0.056	0.063

* Refer to Table 3 or the Nomenclature for metrics definitions.
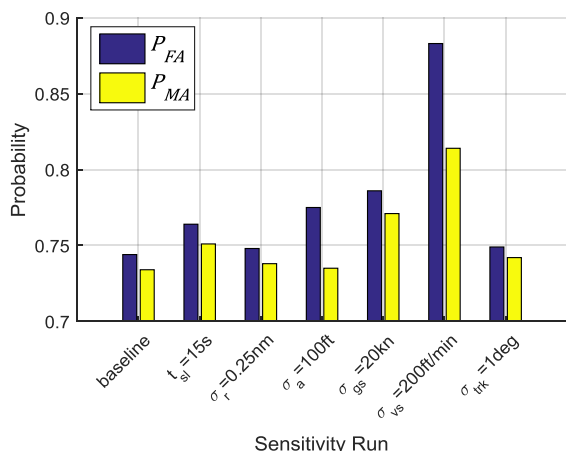
Figure 20. Observed false alert and missed alert probabilities, P_{FA} and P_{MA}, for the baseline scenario and the six sensitivity scenarios.

Figure 21. Observed missed detection probabilities, P_{MD}, for the baseline scenario and the six sensitivity scenarios.

V. Summary

There is an effort underway at NASA to parametrically characterize the performance of the conflict detection function of a separation assurance system in terms of the data quality and data content available to that function. This parametric characterization will allow for the quick understanding of the performance of various allocations of the CD function (e.g., airborne, ground-based, and under various surveillance conditions) where the data quality and data content can be parametrically described.

This paper presents an analysis capability that has been developed for the characterization of conflict detection performance. A surveillance error model has been developed and implemented. An appropriate set of performance

metrics have been rigorously defined. The analysis model was exercised by using time-shifted, recorded track data from the NAS. The appropriate data conditioning and filtering approaches for removing surveillance error and sensor fusion-induced noise were described in this paper. Finally a set of preliminary analysis runs were presented to demonstrate the ability of time-shifted recorded track data scenarios in creating losses-of-separation and to show the observed trends in the performance metrics due to variations in the input parameters. Preliminary sensitivity runs for the surveillance error model parameters were also presented.

The preliminary results from this conflict detection analysis capability indicate appropriate trends in the performance metrics and demonstrate that time-shifted, recorded traffic data playback and analysis is a suitable approach for evaluating conflict detection performance in a parametric way. The peak number of true LOS created by time-shifting of the recorded data was achieved using a uniform maximum time shift of one hour, $U(-1hr, 1hr)$. Different randomization scenarios of that one-hour time shift showed small variability in the metrics, thereby justifying the use of a single baseline scenario from that set for investigating the trends in the analysis metrics. The baseline performance of the conflict detection function using a nominal set of parameters and under the conditions tested was relatively poor, with false alert and missed alert probabilities of 74.5% and 73.5%, respectively, and a missed detection probability of 6.5%, primarily due to the state-projection trajectory prediction method used. The CD cycle period showed a small change in the false alert and missed alert probabilities but a significant increase in missed detection probability, from 0.4% at a CD cycle period of 15 seconds, to more than 45% at a CD cycle period of 5 minutes. As expected, longer conflict detection time horizons had an increased probability of both false alerts and missed alerts but a decreased probability of missed detections. Additionally, conflict alerts that reported a large time-to LOS had a high probability of being false, missed alerts were more likely for LOS farther out in the detection horizon, and the bulk of the LOS successfully detected were detected with less than 5 minutes time-to LOS. The CD performance metrics indicated an appropriate direction for the sensitivity relative to the surveillance error model parameters while further analysis is required to understand the magnitude of uncertainty in the performance metrics.

The next steps in this work will be to continue to develop the analysis capability in order to conduct the complete characterization of CD performance with respect to data quality and data content. Intent information as well as a trajectory predictor that can use that intent information will be the next components added to the analysis. Parametric descriptions for data quality and data content will then be analyzed for current day and proposed future surveillance technologies to identify the performance of the CD function under these conditions. Finally, a complete set of sensitivity analyses will be generated to understand the impact of the various input parameters on the CD performance metrics.

Acknowledgments

References

[1]Bilimoria, K.D., "Survey of Air/Ground and Human/Automation Functional Allocation for Separation Assurance," *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 17th AIAA/ISSM*, Indianapolis, Indiana, September 17-19, 2012.

[2]Lauderdale, T.A., Cone, A.C., Bowe, A.R., "Relative Significance of Trajectory Prediction Errors on an Automated Separation Assurance Algorithm," *Ninth UAS/Europe Air Traffic Management Research and Development Seminar (ATM2011)*, Berlin, Germany, 14-17 June 2011.

[3]Yang, L.C., Kuchar, J.K., *Aircraft Conflict Analysis and Real-Time Conflict Probing Using Probabilistic Trajectory Modeling*, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 2000.

[4]Kuchar, J.K., Yang, L.C., "A Review of Conflict Detection and Resolution Modeling Methods," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1. No. 4, December, 2000.

[5]Krozel, J., Peters, M.E., "Conflict Detection and Resolution for Future Air Transportation Management," *NASA Technical Report, NAS2-14285*, April, 1997.

[6]Karr, D.A., Vivona, R.A., "Conflict Detection Using Variable Four-Dimensional Uncertainty Bounds to Control Missed Alerts," *AIAA Guidance Navigation, and Control Conference and Exhibit*, Keystone, Colorado, August 21-24, 2006.

[7]Wickens, C.D., et.al., "False Alerts in Air Traffic Control Conflict Alerting System: Is There a 'Cry Wolf' Effect?" *HUMAN FACTORS,* Vol. 51, No. 4, August 2009, pp. 446-462.

[8]Thomas, L.C., Wickens, C.D., Rantanen, E.M., "Imperfect Automation in Air Traffic Alerts: A Review of Conflict Detection Algorithms and Their Implications for Human Factors Research," *Proceedings of the 47th Annual Meeting of the Human Factors and Ergonomics Society*, Santa Monica, California, 2003.

[9]Paglione, M.M., Ryan, H.F., Liu, S., "Evaluation of En Route Host Computer System's Tactical Alert Processing: Description and Methodology," Technical Note, US Department of Transportation, Federal Aviation Administration, DOT/FAA/TC/TN07/13, February 2007.

[10]Bilimoria, K.D., "Methodology for the Performance Evaluation of a Conflict Probe," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 3, May-June 2001.

[11]Gaydos, T.L., Liao, B., Smith, E.C., Wang, L., "Analysis of Conflict Detection Performance for Trajectory-Based Descent Operations," *Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013)*, Chicago, Illinois, June 10-13, 2013.

[12]Shakarian, A., Haraldsdottir, A., "Required Total System Performance and Results of a Short Term Conflict Alert Simulation Study," *4th US/Europe Air Traffic Management R&D Seminar*, Santa Fe, New Mexico, December 3-7, 2001.

[13]Paielli, R., Erzberger, H., "Improved Conflict Detection for Reducing Operational Errors in Air Traffic Control," *AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum*, Chicago, Illinois, September 20-22, 2004.

[14]Narkawicz, A., Muñoz, C. "A Formally Verified Conflict Detection Algorithm for Polynomial Trajectories" *Proceedings of the 2015 AIAA Infotech @ Aerospace Conference*, AIAA-2015-0795, January 2015.

[15]Chung, W.W., et. al., "Surveillance and Datalink Communication Performance Analysis for Distributed Separation Assurance System Architectures," *NASA Contractor Report, CR-2012-217590*, July 2012.

[16]Finkelsztein, D.M., Sturdy, J.L., Alaverdi, O., Hochwarth, J.K., "4D Dynamic Required Navigation Performance Final Cashion, P., Lozito, S., "The Effects of Different Levels of Intent Information on Pilot Self-Separation Performance," *Proceedings of International Symposium on Aviation Psychology*, 1999.

[17]Lewis, T.A., Phojanamongkolkij, N., Wing, D.J., "The Effects of Limited Intent Information Availability on Self-Separation in Mixed Operations," *2012 Integrated Communications Navigation and Surveillance (ICNS) Conference*, Herndon, Virginia, April 24-26, 2012.

[18]Report," *NASA Contractor Report, CR-2011-217051*, February, 2011.

[19]Wing, D.J., Barmore, B.E., Krishnamurthy, K., "Use of Traffic Intent Information by Autonomous Aircraft in Constrained Operations," *AIAA Guidance Navigation and Control Conference and Exhibit, AIAA-2002-4555*, Monterey, California, August 5-8, 2002.

[20]Ruigrok, R.C., Clari, M.S., "The Use of Aircraft Intent Information in Airborne Separation Assurance Systems," *FAA-Eurocontrol Third Technical Interchange Meeting (TIM) on Airborne Separation Assurance Systems*, San Francisco, California, October 21-23, 2002.

[21]Paglione, M.M., Oaks, R.D., Bilimoria, K.D., "Methodology for Generating Conflict Scenarios by Time Shifting Recorded Traffic Data," *AIAA's 3rd Annual Aviation Technology, Integration, and Operations (ATIO) Forum*, Denver, Colorado, November 17-19, 2003.

[22]Federal Aviation Administration, *Aviation System Performance Metrics (ASPM),* https://aspm.faa.gov/.

[23]Federal Aviation Administration, *Air Traffic Organizational Policy,* JO 7110.65V, CHG 2, February 19, 2014.

[24]Welch, G., Bishop, G., "An Introduction to the Kalman Filter," University of North Carolina at Chapel Hill, UNC-Chapel Hill, TR 95-041, July 24, 2006.