

Predicting Real-Time Safety of the National Airspace System

Indranil Roychoudhury*

SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA

Lilly Spirkovska[†] and Matthew Daigle[‡] and Edward Balaban[§]

NASA Ames Research Center, Moffett Field, CA 94035, USA

Shankar Sankararaman[¶] and Chetan Kulkarni^{||}

SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA

Scott Poll^{**} and Kai Goebel^{††}

NASA Ames Research Center, Moffett Field, CA 94035, USA

Situation awareness is necessary for operators to make informed decisions regarding avoidance of airspace hazards. To this end, each operator must consolidate operations-relevant information from disparate sources and apply extensive domain knowledge to correctly interpret the current state of the NAS as well as forecast its (combined) evolution over the duration of the NAS operation. This time- and workload-intensive process is periodically repeated throughout the operation so that changes can be managed in a timely manner. The imprecision, inaccuracy, inconsistency, and incompleteness of the incoming data further challenges the process. To facilitate informed decision making, this paper presents a model-based framework for the *automated* real-time monitoring and prediction of possible effects of airspace hazards on the safety of the National Airspace System (NAS). First, hazards to flight are identified and transformed into safety metrics, that is, quantities of interest that could be evaluated based on available data and are predictive of an unsafe event. The safety metrics and associated thresholds that specify when an event transitions from *safe* to *unsafe* are combined with models of airspace operations and aircraft dynamics. The framework can include any hazard to flight that can be modeled quantitatively. Models can be detailed and complex, or they can be considerably simplified, as appropriate to the application. Real-time NAS safety monitoring and prediction begins with an estimate of the state of the NAS using the dynamic models. Given the state estimate and a probability distribution of future inputs to the NAS, we can then predict the evolution of the NAS - the future state - and the occurrence of hazards and unsafe events. The entire probability distribution of airspace safety metrics is computed, not just point estimates, without significant assumptions regarding the distribution type and/or parameters. We demonstrate our overall approach through a simulated scenario in which we predict the occurrence of some unsafe events and show how these predictions evolve in time as flight operations progress. Predictions accounting for common sources of uncertainty are included and it is shown how the predictions improve in time, become more confident, and change dynamically as new information is made available to the prediction algorithm.

*Research Scientist, Intelligent Systems Division, Discovery and Systems Health Area, MS 269-3, AIAA Member.

[†]Computer Engineer, Intelligent Systems Division, Discovery and Systems Health Area, MS 269-3.

[‡]Research Computer Scientist, Intelligent Systems Division, Discovery and Systems Health Area, MS 269-3, AIAA Member.

[§]Computer Engineer, Intelligent Systems Division, Discovery and System Health Area, MS 269-4.

[¶]Research Scientist, Intelligent Systems Division, Discovery and Systems Health Area, MS 269-3.

^{||}Research Scientist, Intelligent Systems Division, Discovery and Systems Health Area, MS 269-3, AIAA Member.

**Computer Engineer, Intelligent Systems Division, Discovery and Systems Health Area, MS 269-1.

^{††}Senior Scientist, Intelligent Systems Division, Discovery and System Health Area, MS 269-4, AIAA Senior Member.

I. Introduction

Pilots, flight controllers, and other service providers must be familiar with existing and potential hazards of the National Airspace System (NAS) in order to plan for and execute safe and efficient aircraft flights. There are many types of hazards, e.g., busy airport location or bad airport layout, unsuitable aircraft and avionics design for given circumstances, poor operator training or fitness for flight. Some of these hazards are immutable, such as terrain; some are fixed over a long duration, such as runway layout; some change periodically, such as seasonal bird patterns; and some can change over the course of a single flight, such as the weather. Such hazards lead to increased flight risk and a decrease in NAS safety.

Current state-of-the-art approaches to deal with airspace hazards include the FAA’s Flight Risk Analysis Tool (FRAT)¹ and FAA’s Safety Management System (SMS),² among others. These approaches rely on the operator (pilot, dispatcher, controller, etc.) to identify known and potential hazards and maintain situational awareness of their evolution. However, in many cases, human cognitive biases, external pressures (e.g., passenger desires), and inertia often seep into decision-making and reduce the effectiveness of these approaches. In contrast, we develop an *automated* approach to assessing and predicting the safety of the airspace by transforming subjective, potentially-stale decisions to objective, regularly updated, informed decisions, thereby improving safety for all NAS users. Additionally, many existing tools focus on a subset of all possible hazards that can affect airspace safety. Instead, our framework is general enough to include any hazard that can be modeled quantitatively. Further, oftentimes, hazards are evaluated in isolation from other hazards; this can lead to locally-optimal decisions that do not consider global consequences. For example, avoiding en-route weather without considering airspace congestion along alternate routes could lead to fuel exhaustion. Our approach provides information about the current and predicted state of all (modeled) hazards concurrently, allowing for more informed decisions. Finally, our work differs in how we handle uncertainty. Rather than ignoring uncertainty (of weather forecast or expected traffic, for example) or just adding a “fudge factor” to account for uncertainty (separating aircraft by a little extra, for example, to account for uncertainty in when a pilot will slow on final approach to landing), we present a systematic, integrated framework for a rigorous, mathematical treatment of uncertainty. Based on principles of probability and Bayesian analysis, the proposed methodology can compute the entire probability distribution (expressed in terms of probability density functions) of NAS-related quantities of interest without making restrictive assumptions regarding their distribution type and/or parameters.

This work aims at providing techniques and tools to assess and predict the safety of the NAS in real time, enabling NAS operators to take preemptive actions that avoid unsafe situations and reducing the need to mitigate them. We limit the scope of the safety assessment to hazards that (i) can be measured or estimated through one or more data sources, and (ii) follow a predictable evolution. To this end, we have developed an initial framework for airspace real-time monitoring and prediction. The prediction horizon may vary from a few minutes to many hours, depending on availability and accuracy of forecasts. The framework is model-based, that is, we develop models of the various components of the NAS and how they interact, in order to predict its behavior over time. The overall approach consists of two parts: an offline phase, in which we define *safety metrics* and safety thresholds, and develop the models; and an online phase, in which we compute the probability distributions for quantities of interest, such as the state of the NAS, future state of the NAS, and occurrence, timing, and likelihood (measured in terms of probability) of undesirable events.

The paper is organized as follows. The safety monitoring and prediction problem is formulated in Section II. Sections III and IV present the offline safety analysis and the online real-time monitoring and prediction framework, respectively. Section V demonstrates the implementation of this framework in simulation through an illustrative scenario, concerning wake turbulence. Finally, Section VI concludes the paper and presents future work.

II. Problem Formulation

For any discrete time point k , the NAS, or a subset of interest, is in some state:

$$\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_n(k)]^T, \quad (1)$$

including, for example, aircraft positions and speeds, weather system positions, etc. From these states, at time k , we can compute a set of safety metrics, ϕ , using an algebraic function \mathbf{F} of the states:

$$\boldsymbol{\phi}(k) = \begin{bmatrix} \phi_1(k) \\ \phi_2(k) \\ \vdots \\ \phi_m(k) \end{bmatrix} = \begin{bmatrix} F_1(\mathbf{x}(k)) \\ F_2(\mathbf{x}(k)) \\ \vdots \\ F_m(\mathbf{x}(k)) \end{bmatrix} = \mathbf{F}(\mathbf{x}(k)). \quad (2)$$

So, if we know the state of the NAS at some point in time, using \mathbf{F} we can compute the corresponding safety metric vector.

There are points in the space of $\boldsymbol{\phi}$ (or, equivalently, \mathbf{x}) that are unacceptable to NAS operations, corresponding to unsafe events $e \in E$, such as loss of separation, an aircraft being within a convective weather region, high congestion, low fuel, etc:

$$E = \{e_1, e_2, \dots, e_l\}. \quad (3)$$

The boundary between the acceptable and unacceptable space of $\boldsymbol{\phi}$ is defined through a threshold function on that space:

$$\mathbf{o}_E(k) = \begin{bmatrix} o_{e_1}(k) \\ o_{e_2}(k) \\ \vdots \\ o_{e_l}(k) \end{bmatrix} = \begin{bmatrix} T_{e_1}(\boldsymbol{\phi}(k)) \\ T_{e_2}(\boldsymbol{\phi}(k)) \\ \vdots \\ T_{e_l}(\boldsymbol{\phi}(k)) \end{bmatrix} = \mathbf{T}_E(\boldsymbol{\phi}(k)), \quad (4)$$

where $\mathbf{o}_{e_i}(k)$ is a Boolean variable that indicates whether event e_i has occurred ($\mathbf{o}_{e_i}(k) = \text{true}$) or not ($\mathbf{o}_{e_i}(k) = \text{false}$). The NAS is in an acceptable state if all elements of $\mathbf{o}_E(k)$ are false, and in an unacceptable state otherwise. For each event e we have an individual threshold function T_e , which, in general, is a function of all the safety metrics, but, in practice is typically only a function of a small subset. The time of occurrence for some event e , k_e , is then defined as the first time at which its threshold function evaluates to true, i.e.,

$$k_e(k) \triangleq \inf\{k_i \in \mathbb{N} : k_i \geq k \wedge T_e(\boldsymbol{\phi}(k_i)) = \text{true}\}, \quad (5)$$

where k is the current time. The time remaining until that event, Δk_e , is defined as:

$$\Delta k_e(k) \triangleq k_e(k) - k. \quad (6)$$

We define the vector of event times (i.e., the earliest time that each event e_i occurs) as

$$\mathbf{k}_E = [k_{e_1}, k_{e_2}, \dots, k_{e_l}]^T. \quad (7)$$

Since the state of the NAS is not known exactly, and there is much uncertainty in future conditions and operations, all of these quantities are uncertain and must be described by probability distributions (expressed in terms of underlying probability density functions). Thus, the goal is to compute the probability distributions of these quantities, not point estimates. We can then define the *monitoring problem* as follows.

Problem 1 (Monitoring). The *monitoring* problem is, for the current time k , to compute the probability density function $p(\boldsymbol{\phi}(k))$.

Given $p(\boldsymbol{\phi}(k))$, we can also compute $p(\mathbf{o}_E(k))$ using \mathbf{T}_E to determine the probability of any of the safety-related events occurring at time k . Note that probability density functions (denoted by $p(\cdot)$) are used to represent the uncertainty in various variables. Later, $P(\cdot)$ is used to denote probabilities.

For prediction, without loss of generality, we introduce a prediction horizon, $k_H > k$. We are interested only in predicting what happens up to time k_H . We denote the future values of a variable \mathbf{a} from time k to k_H using $\mathbf{A}_k^{k_H}$.

Problem 2 (Prediction). The *prediction* problem is, for the current time k , to compute the probability density function $p(\boldsymbol{\Phi}_k^{k_H})$.

Given $p(\boldsymbol{\Phi}_k^{k_H})$, we can also compute (i) the probability density function $p(\mathbf{O}_{E,k}^{k_H})$ using \mathbf{T}_E , (ii) the probability of some event e_i occurring within $[k, k_H]$, (iii) the probability of any subset of events in E occurring within $[k, k_H]$, (iv) the probability density function of future event times, $p(\mathbf{k}_E)$, and (v) other derived quantities of interest.

III. Safety Analysis and Modeling

The first step in applying our framework to the aviation safety domain requires determining the quantities of interest for monitoring and prediction. To this end, we researched National Transportation Safety Board (NTSB) and FAA aviation accident and incident reports.³ Not every unsafe situation leads to an accident or incident, however, so we greatly expanded the list of potential hazards to airspace safety by also studying Aviation Safety Reporting System (ASRS) reports.⁴ For our work, we focused on a subset of hazards within three main categories: (i) airspace-related hazards, e.g., inoperative navigation aids or high congestion; (ii) environmental hazards, e.g., convective weather or animal activity; and (iii) human-performance hazards, e.g., pilot fatigue or pilot distraction. The list of hazards is too voluminous to include in this paper, and is documented in a separate report.⁵ Much prior research^{6,7,8,9,10,11,12} has focused on aircraft malfunction hazards, e.g., engine failure, structural issues, sensor malfunctions, and hence will not be covered in this paper, although our approach can include these hazards if desired.

After identifying potential hazards, the next step is to design a set of safety metrics, ϕ , as an algebraic function \mathbf{F} of the states, \mathbf{x} , that quantify these hazards. These safety metrics are quantities of interest that should be monitored and predicted in order to predict unsafe events. Recall from Section II that an unsafe event is a transition event from an acceptable to unacceptable space of ϕ , and these boundaries between the acceptable and unacceptable spaces are defined through threshold functions, $\mathbf{T}_E(\phi(k))$. Threshold functions can take any general form. Selecting an appropriate threshold can be challenging. For some safety metrics, rules and regulations dictate a threshold (however, each operator can select a more conservative threshold if desired). For example, FAA regulations require that certain categories of flights remain clear of particular special use airspaces (SUA). For this case, the threshold for the safety metric *distance from SUA* could be set to 0. For other safety metrics, analysis, data mining, or consultation with subject matter experts or the operators may be necessary. For example, the threshold for a *congested airspace* safety metric is affected by composition of traffic (e.g., all heavy jets vs gliders, helicopters, small general aviation aircraft, and large and heavy jets), flight paths (single stream vs multi-stream merging), and even the controller on console (some controllers move traffic more efficiently than others). In these cases, we can use a hybrid method to determine appropriate thresholds, combining information from multiple sources. We can also leverage individual information, such as personal minimums, level of certification, years of experience, etc., to determine such thresholds.

To illustrate the process using an example, *aircraft separation* is a safety metric that constantly needs to be monitored and predicted in order to predict a *loss of separation* unsafe event. The safety metric function takes as inputs the positions of two aircraft, and outputs the horizontal distance between them, their heading with respect to each other, and the altitude difference. For the threshold on this safety metric, general separation standards for en-route flight provide the value of 5 nautical miles for lateral separation and 1000 feet for vertical separation (per FAA Order JO 7110.65).

Table 1 lists some example safety metrics, the arguments their corresponding safety metric functions can take, the outputs of these safety metric functions, and example threshold functions. Note that this list is provided for illustrative purposes and is by no means exhaustive. A NAS participant, such as a control center or an airline, would generate their own safety metrics, safety metric functions, and threshold equations based on their requirements. A more comprehensive list of safety metrics, along with a discussion on how to determine thresholds, is available in.⁵

Table 1: Some Example Safety Metrics

Safety metrics	Safety Metrics Function Arguments	Safety Metrics Function Outputs	Example of Threshold Functions
distance and heading to weather event	point of interest, weather severity, weather type, time	distance and heading	distance.thunderstorm > 20 mi and thunderstorm.intensity < MEDIUM
weather at coordinate	point of interest, time	matrix of all weather categories (e.g., hail, rain, snow, mist, mixed, turbulence, thunderstorm, wind, microburst, wind-shear, etc.) and their relevant properties (e.g., severity, phase, type, persistence, direction of movement, etc., temperature, humidity)	A threshold is needed for each element of the matrix. Some examples: turbulence.intensity < MODERATE, thunderstorm.intensity ≤ MODERATE, rain.intensity < SEVERE

Table 1: Some Example Safety Metrics

Safety metrics	Safety Metrics Function Arguments	Safety Metrics Function Outputs	Example of Threshold Functions
risk of wake turbulence	point of interest, time, {weather at coordinate}, type of preceding aircraft	risk category, e.g., low, medium, high	wake_turbulence_risk ≤ MEDIUM
avoidance areas at coordinate	point of interest, time	matrix of avoidance area categories (e.g., SUAs such as restricted, warning, alert, TFR, practice areas, controller training areas, special event areas, etc., obstacles, noise sensitive areas, airport hotspots) and status (e.g., active/inactive, restriction type, allowed noise level, etc.)	avoidance_area.tfr.active = FALSE
degree of operational normalcy	volume of interest, time, number of restrictions (like miles in trail (MIT) to an adjoining center), cumulative scheduled delay, flow control programs in effect, emergency/non-emergency situations	Normalcy score, e.g., low, medium, high	ops_normalcy > LOW

IV. Real-Time Monitoring and Prediction

Once potential hazards are selected and safety metrics defined, the goal is to, in real-time, monitor the safety metrics, $\phi(k)$, and predict the occurrence of unsafe events, E . In order to estimate the current value of $\phi(k)$, we must first estimate the state of the NAS, \mathbf{x} . Then, given the state estimate and probability distributions (in terms of probability density functions) of future inputs to the NAS (e.g., winds aloft forecasts, planned aircraft routes, etc., for some future duration of interest), we can predict the future values of \mathbf{x} and ϕ and the occurrence of events E . Following a model-based approach, both these tasks require dynamic models of the NAS.

First, Section IV-A describes a generic architecture that enables real-time monitoring and prediction. Then, Section IV-B describes modeling approaches for the NAS. Section IV-C covers uncertainty quantification and management. Section IV-D discusses solutions to the monitoring problem, and Section IV-E discusses solutions to the prediction problem, using these models. Section IV-F discusses how likelihoods of unsafe events are combined.

A. Architecture

Monitoring and prediction must be performed in an integrated manner in order to continuously assess the safety of the overall NAS; the outputs of the monitoring step are inputs to the prediction step. Our approach to such an architecture is model-based, that is, we develop models of the various components of NAS and how such components interact, in order to predict its behavior over time. In order to predict the value of $\phi(k)$ in the future, and to predict when undesirable events will occur, we require a model describing how the state \mathbf{x} evolves in time:

$$\mathbf{x}(k+1) = \mathbf{f}(k, \mathbf{x}(k), \mathbf{u}(k), \mathbf{v}(k)), \quad (8)$$

where \mathbf{f} is the state function, \mathbf{u} is the input vector (exogenous inputs to the system, such as the aircraft's intended flight routes and wind velocity at various altitudes), and \mathbf{v} is the process noise vector. The state equation allows us to compute future values of the state given the inputs, and to compute future values of ϕ and evaluate the threshold function, \mathbf{T}_E .

In order to make a prediction at time k using \mathbf{f} , we require $\mathbf{x}(k)$, which, in general, is not known. Instead, we have available an output vector \mathbf{y} , defined through an output equation:

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{n}(k)), \quad (9)$$

where \mathbf{h} is the output function, and \mathbf{n} is the sensor noise vector. We need to infer $\mathbf{x}(k)$ from $\mathbf{y}(k)$ using

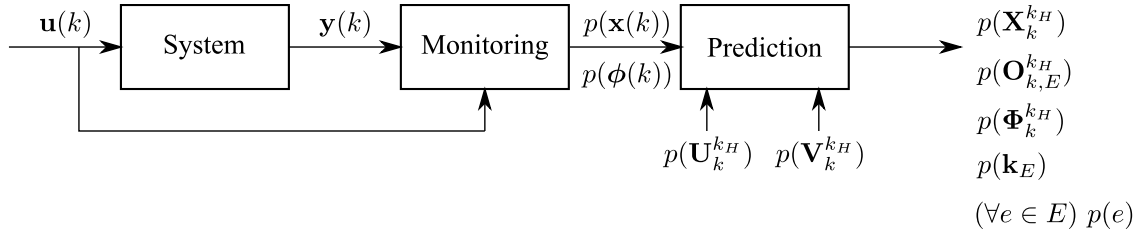


Figure 1. Real-time architecture.

\mathbf{f} and \mathbf{h} , and for this we use state estimation algorithms. Due to the sensor noise $\mathbf{n}(k)$, we obtain only a probability density function $p(\mathbf{x}(k))$.

The overall architecture is shown in Fig. 1. In the *monitoring* step, given the inputs \mathbf{u} , the outputs $\mathbf{y}(k)$, the state equation \mathbf{f} , and the output equation \mathbf{h} , we must estimate the current state $p(\mathbf{x})$, and, given $p(\mathbf{x})$ and the safety metric equation \mathbf{F} , compute $p(\phi(k))$. In the *prediction* step, at time k , given $p(\mathbf{x}(k))$, the state function \mathbf{f} , the safety metrics ϕ , the safety metric function \mathbf{F} , the set of events E and threshold function \mathbf{T}_E , the future input probability density function $p(\mathbf{U}_k^{kH})$, and the future process noise probability density function $p(\mathbf{V}_k^{kH})$, we must compute probability density functions for \mathbf{X}_k^{kH} , Φ_k^{kH} , $\mathbf{O}_{E,k}^{kH}$, \mathbf{k}_E , and $P(e)$ for each $e \in E$.

B. Modeling

The models must define the state \mathbf{x} , the inputs \mathbf{u} , the outputs \mathbf{y} , the state function \mathbf{f} , and the output function \mathbf{h} . In addition, we require models of the process noise \mathbf{v} and sensor noise \mathbf{n} and other sources of uncertainty.

To model the NAS at a system-level, we require models of aircraft, pilots, controllers, weather phenomena, restricted airspace, etc. In this paper, we limit to models of aircraft only. We assume that pilots control the aircraft according to a known flight plan, and consider only wind as a weather phenomenon. In the following, we describe the equations of the models used.^a Note that because our approach is model-based, the medium-fidelity models used here can always be replaced with higher fidelity models, without any changes required in the algorithms (i.e., the models are inputs to the algorithms).

We use kinematic models of aircraft navigation with simplified dynamics and control, similar to the models developed by others.^{13,14,15} The aircraft state vector is defined as

$$\mathbf{x}(t) = \begin{bmatrix} V_a(t) & V_h(t) & \chi_a(t) & h(t) & \lambda(t) & \tau(t) \end{bmatrix}^T, \quad (10)$$

where V_a is the indicated airspeed, V_h is the vertical speed, χ_a is the aircraft heading, h is the mean sea level (MSL) altitude, λ is the latitude, and τ is the longitude. Note that we assume there is no roll and the aircraft may rotate only along the other two axes.

The wind vector, with magnitude V_w and heading χ_w , causes the aircraft to drift. The groundspeed vector, with magnitude V_g and heading χ_g , is the vector addition of the airspeed and wind vectors.¹⁴ Thus, in order to follow a desired ground track, the commanded heading must be corrected for the wind.

For the purposes of this paper, we assume that each aircraft has a set of waypoints or fixes which it must navigate to, each defined by a latitude λ^* , a longitude τ^* , an altitude h^* , and a time t^* . We assume also that the control is such that the commanded airspeed, climb rate (vertical speed), and heading are set to reach the desired position at the desired time, using great-circle heading (χ_{GC}) and distance (d_{GC}) calculations:

$$d_{GC} = 2R \arcsin \left(\sqrt{\sin^2 \left(\frac{\lambda^* - \lambda}{2} \right) + \cos(\lambda) \cos(\lambda^*) \sin^2 \left(\frac{\tau^* - \tau}{2} \right)} \right), \quad (11)$$

$$\chi_{GC} = \arctan \frac{\sin(\tau^* - \tau) \cos(\lambda^*)}{(\sin(\lambda^*) \cos(\lambda) - \sin(\lambda) \cos(\lambda^*) \cos(\tau^* - \tau))}, \quad (12)$$

^aThe models are described in continuous time t ; note that they are converted to discrete time for a given sampling rate for use in the algorithmic framework.

where $R = R_E + h$, with R_E being the MSL radius of the Earth. Then, commanded airspeed V_a^* , commanded heading χ_a^* , and commanded vertical speed V_h^* are described by

$$V_a^* = \frac{d_{GC}}{t^* - t} - V_w \cos(\chi_w - \chi_a), \quad (13)$$

$$\chi_a^* = \chi_{GC} - \arcsin\left(\frac{V_w}{V_a} \sin(\chi_w - \chi_g)\right), \quad (14)$$

$$V_h^* = \frac{h^* - h}{t^* - t}, \quad (15)$$

where $V_w \cos(\chi_w - \chi_a)$ is the wind correction term to the airspeed, and the arcsin term is a wind correction term to the heading.¹⁴ The wind vector W is considered an input defined by its speed and heading, or equivalently its north and east components, W_N and W_E , and is assumed to be a function of position.

The latitude, longitude, and altitude evolve in time as described by

$$\dot{\lambda}_t = (V_a \cos \gamma \cos \chi_a + W_N)/R, \quad (16)$$

$$\dot{\tau}_t = (V_a \cos \gamma \sin \chi_a + W_E)/(R \cos \lambda), \quad (17)$$

$$\dot{h} = V_h, \quad (18)$$

where γ is the flight path angle, approximated by

$$\gamma = \arcsin(\dot{h}/V_a). \quad (19)$$

We assume some simplified dynamics, where the airspeed, climb rate, and heading all change to the commanded values with some ‘‘inertia’’ that abstracts the more complex aerodynamic equations:

$$\dot{V}_a = (V_a^* - V_a)/J_a, \quad (20)$$

$$\dot{V}_h = (V_h^* - V_h)/J_h, \quad (21)$$

$$\dot{\chi}_a = (\chi_a^* - \chi_a)/J_{\chi_a}, \quad (22)$$

where the J parameters are represent the inertia.

The inputs to this model include the wind and desired position variables:

$$\mathbf{u}(t) = \left[V_w(t) \quad \chi_w(t) \quad \lambda^*(t) \quad \tau^*(t) \quad h^*(t) \quad t^*(t) \right]^T. \quad (23)$$

The overall model of a given region of the NAS is composed of the models for the aircraft in that region. The system-level inputs are the winds and waypoint sets for each aircraft.

C. Uncertainty Quantification and Management

In order to systematically and accurately account for the presence of uncertainty in the NAS, a series of uncertainty-related activities needs to be performed. These activities are (i) uncertainty characterization/quantification, (ii) uncertainty propagation, and (iii) uncertainty management. Uncertainty characterization/quantification deals with estimating and characterizing the various sources of uncertainty that affect NAS safety analysis. Uncertainty propagation deals with systematically quantifying the effect of the aforementioned uncertainties on unsafe events and quantities of interest (safety metrics, time to unsafe event, etc.). Finally, uncertainty management deals with various activities that can effectively conduct safe operations given the estimated amount of uncertainty; typically, uncertainty management is closely tied into decision-making activities.

It is important to identify all quantities that can be potentially uncertain at the start of the modeling stage. Consider any generic time index k_P at which prediction is desired to be performed; the potential sources of uncertainty include: (i) the state at k_P , given by $\mathbf{x}(k_P)$; (ii) the measurement error (sensor noise) at k_P , given by $\mathbf{n}(k_P)$; (iii) the future input values, given by $\mathbf{U}_{k_P}^{k_H}$, and (iv) the future process noise values, $\mathbf{V}_{k_P}^{k_H}$. As a result of the above sources of uncertainty, the safety metrics are also uncertain at all future time instances, i.e., $\Phi_{k_P}^{k_H}$ is uncertain. Hence, it would be necessary to compute the probability density function as $p(\Phi_{k_P}^{k_H})$. Also, there is a probability of occurrence of each safety-related event, denoted as $p(\mathbf{O}_{E,k_P}^{k_H})$. Finally, the time until the occurrence of each safety-related event is also uncertain and such a time also needs to be represented using a probability density function, denoted as $p(\mathbf{k}_E)$.

Algorithm 1 $\Phi_{k_P}^{k_H} \leftarrow \text{Predict}(\mathbf{x}(k_P), \mathbf{U}_{k_P}^{k_H}, \mathbf{V}_{k_P}^{k_H}, k_H)$

```
1: for  $k = k_P$  to  $k_H$  do  
2:    $\mathbf{x}(k+1) \leftarrow \mathbf{f}(k, \mathbf{x}(k), \mathbf{U}_{k_P}^{k_H}(k), \mathbf{V}_{k_P}^{k_H}(k))$   
3:    $\phi(k+1) \leftarrow \mathbf{F}(\mathbf{x}(k+1))$   
4: end for
```

D. Monitoring

Monitoring the NAS can be viewed as a Bayesian inference problem. We have various sensors measuring the current state of the NAS, such as GPS, radar, aircraft instrumentation, and weather observations. We have also a dynamic model of how the NAS evolves, \mathbf{f} , given its inputs \mathbf{u} , such as intended flight routes, scheduled departure times, etc. The inputs to the monitoring problem are the known inputs to the NAS, \mathbf{u} , and the available observations, \mathbf{y} , and the goal is to infer, i.e., estimate, the true state of the NAS, \mathbf{x} using the model. From the state, we can evaluate the safety metrics, ϕ .

Fundamentally, the monitoring step involves the computation of state $\mathbf{x}(k_P)$ at any desired time-of-prediction, k_P . This step can be expressed using Bayes theorem, as:

$$p(\mathbf{x}(k_P)|\mathbf{y}(k_P)) = \frac{p(\mathbf{x}(k_P-1))p(\mathbf{y}(k_P)|\mathbf{x}(k_P))}{\int p(\mathbf{x}(k_P-1))p(\mathbf{y}(k_P)|\mathbf{x}(k_P))d\mathbf{x}(k_P)}. \quad (24)$$

Note that the state-estimate from the previous time step ($p(\mathbf{x}(k_P-1))$) is used as a prior for the current time step. The likelihood term is denoted as $p(\mathbf{y}(k_P)|\mathbf{x}(k_P))$, and is computed based on the output model, explained earlier in Eq. 9.

There are many algorithms that may be used to solve this general problem. Since our models are nonlinear, they require a nonlinear filter (such as unscented Kalman filters or particle filters^{16,17,18,19}). These algorithms all have the same basic structure. For a new time step, we first predict the new state, based on the previous state and the inputs. Then, we correct the state based on the available observations and the likelihood functions. Given some assumption of uncertainty in the model and some assumption of uncertainty in the observations, the algorithms find the most likely state.

E. Prediction

The goal of the prediction problem is to compute the future evolution of the NAS state \mathbf{x} , the safety metrics ϕ , and the occurrence of safety events E from prediction time k_P up through the prediction horizon k_H . In loss of separation prediction, for example, $k_H - k_P = 20$ minutes is used in practice. Ultimately, we want to find the probability density functions of the state trajectory, $p(\mathbf{X}_{k_P}^{k_H})$, the safety metric trajectory, $p(\Phi_{k_P}^{k_H})$, and the event times $p(\mathbf{k}_E)$, as well as the probability of the different events occurring within the prediction horizon.

Ultimately, the prediction problem is an uncertainty propagation problem.²⁰ We have an uncertain estimate of the current system state, and uncertain future inputs to the NAS. We need to propagate all these uncertainties through the model \mathbf{f} to determine the probability density functions of future variables. This is a key difference of our work from past approaches to prediction in the NAS: we bring in all the sources of uncertainty from the beginning, in a formal, principled way, rather than using deterministic models for prediction and then accounting for errors after-the-fact.

The underlying function to any prediction algorithm is given as Algorithm 1. At a given time of prediction k_P , with realizations of the state $\mathbf{x}(k_P)$, the future inputs $\mathbf{U}_{k_P}^{k_H}$, the future process noise $\mathbf{V}_{k_P}^{k_H}$, and a prediction horizon $k_H > k_P$, the algorithm simulates the NAS state forward up to k_H , and computing the corresponding safety metrics. From the safety metrics, we can compute whether any of the unsafe events have occurred and, if so, when they first occur. Note that the performance of this prediction algorithm is directly dependent on the accuracy of the information fed in the form of inputs into this algorithm.

We use Monte Carlo sampling for prediction in this paper. A standard Monte Carlo approach generates many realizations of the prediction inputs, and calls Algorithm 1 for each realization. We compute samples of future trajectories of the safety metrics, from which we can directly compute the times of unsafe events, probabilities of events occurring at a certain time or within a certain interval, etc. This set of samples establishes a probability distribution (in terms of density) for the safety metrics and the related variables. For example, we can compute the probability of some e occurring by k_H simply as the number of samples

Algorithm 2 $\{\Phi_{k_P}^{k_H}\}_{i=1}^N = \text{MonteCarlo}(p(\mathbf{x}(k_P)), p(\mathbf{U}_{k_P}^{k_H}), p(\mathbf{V}_{k_P}^{k_H}), k_P, k_H, N)$

- 1: **for** $i = 1$ **to** N **do**
 - 2: $\mathbf{x}(k_P)^i \sim p(\mathbf{x}(k_P))$ {Sample state}
 - 3: $\mathbf{U}_{k_P}^{k_H,i} \sim p(\mathbf{U}_{k_P}^{k_H})$ {Sample future inputs}
 - 4: $\mathbf{V}_{k_P}^{k_H,i} \sim p(\mathbf{V}_{k_P}^{k_H})$ {Sample future process noise}
 - 5: $\Phi_{k_P}^{k_H,i} \leftarrow \text{Predict}(\mathbf{x}(k_P)^i, \mathbf{U}_{k_P}^{k_H,i}, \mathbf{V}_{k_P}^{k_H,i}, k_P, k_H)$
 - 6: **end for**
-

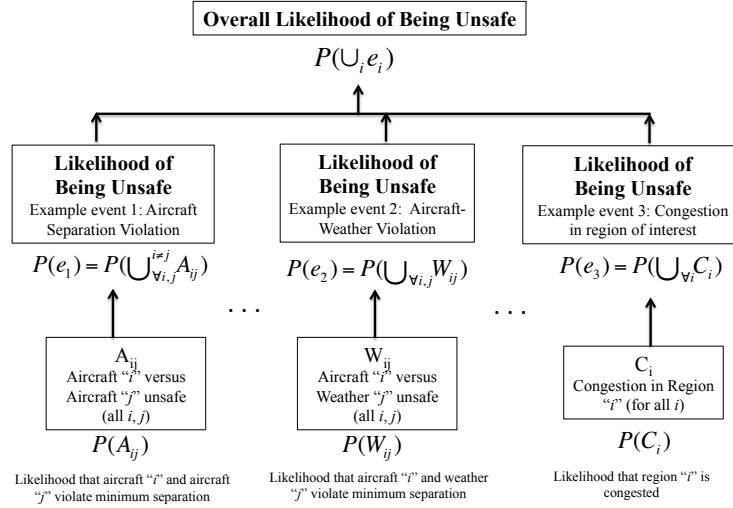


Figure 2. Probability Tree: Likelihood of Unsafe Events

with finite k_E divided by N . Algorithms 1 and 2 can also be easily extended to provide the probability of the event occurring at each time point within $[k_P, k_H]$. Over a set of events, we can easily also compute the probability of any safety event occurring within the prediction horizon, and the probability of one occurring at each time point, and related probability distributions. Though exhaustive sampling can be computationally expensive in large scale applications, alternative efficient methods will be considered in future work.

F. Effect of Uncertainty on Safety: Likelihood of Unsafe Events

Once the uncertainty in the prediction of the safety metrics is computed, the likelihood of unsafe events can be computed. Consider a generic unsafe/undesirable event e , and the above prediction framework can be used to compute the probability of occurrence of e ; let $P(e)$ denote the corresponding probability.

Consider multiple events related to safety, as shown in Fig. 2. For the sake of illustration, three unsafe events are chosen: loss of aircraft-to-aircraft separation (minimum distance requirement not being met), loss of aircraft-to-weather separation (insufficient distance between an aircraft and an adverse weather region), and high congestion (upper workload limit in number of aircraft being violated) in a sector/region of interest; however, the proposed method is general enough to be extended to consider arbitrary types of NAS-related events.

In general, the computation of likelihood and probability of unsafe events varies from one event to another. These probabilities are calculated for all future time instants within $[k_P, k_H]$, based on information available at the present time instant. As time keeps evolving, more information is available, and the probabilities for all future safety-related incidents keep evolving as well. Multiple safety metrics can be computed and the likelihood of safety in each scenario can be computed.

While it is important to compute the likelihood of each unsafe event, it is also important to combine the likelihoods across multiple events. Any occurrence of any unsafe event renders the NAS unsafe. In other words, if any unsafe event were to occur, then it means that the airspace is unsafe. Consider events e_i ($i = 1$ to n) that correspond to multiple unsafe events; the occurrence of event e_i means that safety has been

violated with respect that particular event. Let e denote the event that the NAS is unsafe; hence, e is said to have occurred when any of the events e_i has occurred. Hence, it is possible to calculate the probability that the entire airspace is unsafe, as:

$$P(e) = P(\cup_{\forall i} e_i) \quad (25)$$

Note that $P(e_i)$ ($\forall i$, i.e., for each hazard) and $P(e)$ can be calculated for all future time instants based on information available at the present time instant. As explained earlier, time keeps evolving and these probabilities (for occurrence of unsafe-incidents in the future) keep evolving too.

Also, note that the above equation simply calculates the probability that any unsafe event occurs. However, it does not take into account the criticality associated with each event. Future work needs to account for this factor in computing the overall safety of the airspace and develop metrics that not only include the information regarding uncertainty but also important information regarding risk and criticality.

V. Case Study - Wake Turbulence in the Terminal Airspace

In this section, we illustrate our overall framework with an example simulation case study. The case study centers around a wake turbulence event in the terminal airspace - a very common hazard for aircraft. Additional details about this case study, as well as information about loss of separation in terminal airspace and enroute weather event case studies can be found in a separate report.⁵

Wake turbulence is caused by the wake vortex produced by aircraft at the wing tips due to the pressure differences on the wing.^{21,22,23} The weight, wingspan, and speed of the generating aircraft determine the initial strength and motion of the vortices; however, ambient atmosphere (wind, stability, turbulence, etc.) dictates the eventual motion and decay rate of the vortices. The maximum core velocity of the vortex may exceed 300 ft/sec and the induced rolling moment on an aircraft encountering the wake vortex of another aircraft can exceed its roll control, leading to loss of control.

Even though controllers may provide wake turbulence advisories in certain situations, pilots are generally solely responsible for maintaining adequate horizontal or vertical separation for wake turbulence avoidance. For arrivals to and departures from controlled airports, the controller generally follows separation standards tailored to the particular airport to mitigate potential issues. Such standards generally work well to optimize traffic flow while maintaining safety; however, when less-common situations occur (such as extended departure rolls, go-arounds, or unusual wind conditions), such standards may prove inadequate. One approach to preventing mishaps under such circumstances is to provide additional information about the safety of the airspace in the airport vicinity to both pilots and controllers, such as warnings that standard operating rules or heuristics may not work in a particular situation.

For this case study we consider the terminal airspace of the San Francisco International Airport (SFO), with its two sets of intersecting parallel runways, one set used for departures and the other set used for arrivals. In our scenario, a light aircraft A1 (e.g. Piper Aztec), is waiting on runway 01L for takeoff clearance from tower controllers (see Fig. 3). A large aircraft A2 (e.g. Boeing 777) is coming in for a landing on crossing runway 28L. As the scenario proceeds, the pilot of A2 decides to do a go-around right after touching down because of the difficulties with maintaining directional control due to a strong crosswind (19 knots, coming from the north). As the aircraft accelerates again for the go-around and starts generating lift, a region of wake turbulence is formed behind it. Due to the wind, the vortices will drift onto runways 01L and 01R, toward where aircraft A1 is waiting for takeoff. Ordinarily, there are no issues with residual wake turbulence in the general area of the runway intersection as landing aircraft touch down (and stop generating lift) well before it and departing aircraft rotate (and start generating wake vortices) well after the intersection. In this, somewhat rare, scenario a controller may not consider all of the implications of A2's go-around and clear A1 for departure as soon as A2 has rotated, while the region of wake turbulence is still present (which may be particularly dangerous for the lighter A1, since it may rotate before the intersection of the runways).

As described above, wake vortices are generated as a byproduct of an aircraft generating lift. The initial descent rate of the vortices is determined by weight, flight speed, and wingspan of the generating aircraft and is usually between 300 and 500 fpm for about 30 sec. The descent rate decreases and eventually reaches zero between 500 to 900 ft below the flight path. For our model, we simplified this behavior and compute the wake turbulence region as follows. Given the aircraft position for the past m minutes (in which the turbulence generated before $t - m$ has dissipated), we compute points at the wingtips at each time step, and extend them along the line between the wingtips proportional to $t - t_{prev}$ for $m \leq t_{prev} \leq t$, and then shift

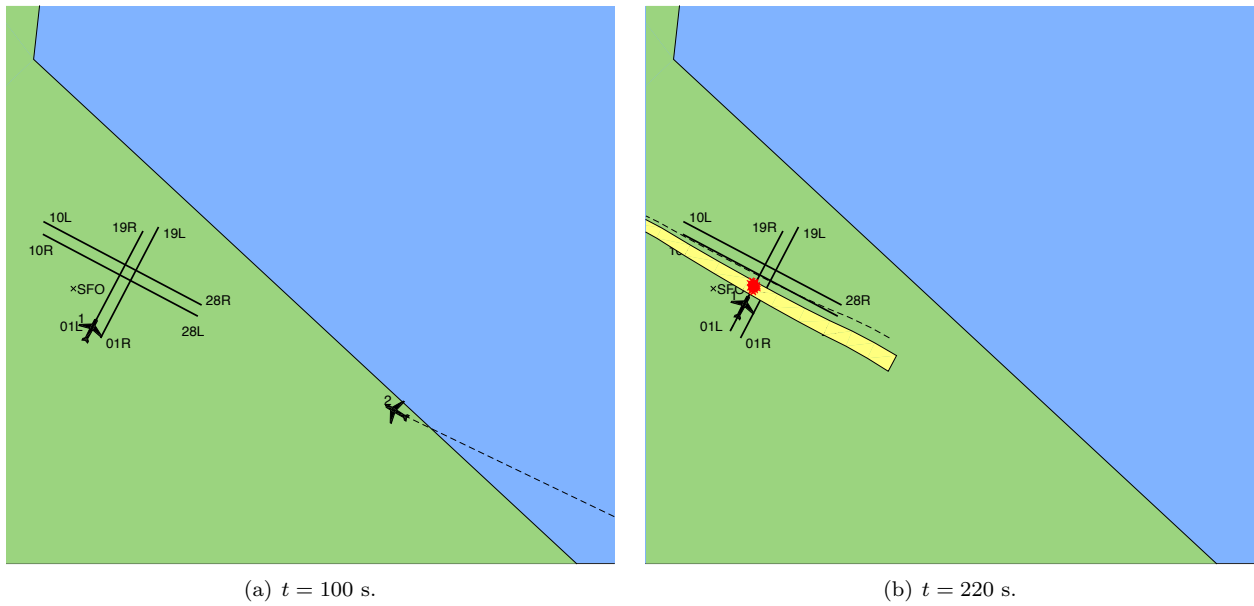


Figure 3. Wake turbulence scenario. Aircraft not drawn to scale.

them in the direction of the wind at t_{prev} . The wake turbulence region is then defined by these points; this procedure is summarized in Fig. 4, where the wake region at time t is computed based on the aircraft position at times t , $t-1$, and $t-2$ (p_t , p_{t-1} , and p_{t-2} , respectively). At p_t , the width is equal to the distance between the wingtips and the wake region points are placed there. At p_{t-1} , the width of the region has become larger than the original wingtip distance and shifted in the direction of the wind. At p_{t-2} , the turbulence has spread still wider and shifted further by the wind from its original location, and this continues for each past time step (the finer the sampling time, the finer the resolution of the region). This region extends below for 1000 feet. This approximation of the wake region is meant to be more realistic than a simple region computed based on separation standards, but simple enough for demonstration purposes. More advanced models, such as those in^{21,22,23} can be used in future work.

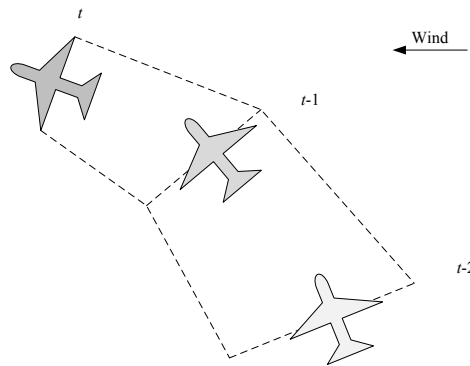


Figure 4. Computed wake turbulence region at time t .

Connecting the description of this scenario to our framework, the event e we are concerned with is whether an aircraft enters a region of wake turbulence created by another aircraft. Given two aircraft, T_e is defined to be true when the position of the first aircraft is within the wake turbulence region of the second aircraft. Since we have two aircraft, there are 2 different wake turbulence events we want to predict (A1 in wake of A2, A2 in wake of A1). The safety metric in this case is computed as the distance between an aircraft and the wake region of another aircraft.

Initially, A1 is on the runway. The time of takeoff clearance is unknown, and will be given once the

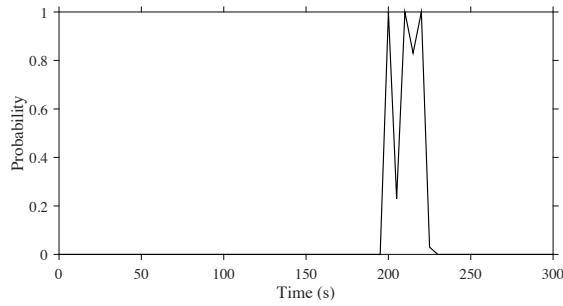


Figure 5. Probability of A1 being in the wake of A2 within the next 5 minutes as a function of time.

approaching aircraft on the intersecting runways is clear. A2 is lined up for approach at about 150 knots. As mentioned earlier, there is a wind coming from the north of 19 knots, that will be pushing the wake turbulence region south toward A1.

The safety metric is constantly monitored, and predictions are made up through 5 minutes into the future at a discrete time step of 10 s. From the pilot’s perspective, we compute the probability that a wake turbulence event will happen in the next 5 minutes. Fig. 5 shows the computed probability of A1 being in the wake region of A2 within the next 5 minutes, computed at each time. Initially, since the takeoff time is unknown, the probability is zero. At 195 s, takeoff clearance is known and immediately the prediction algorithm computes a high probability of encountering wake turbulence, since we know that A2 is not landing. The probability remains high until the aircraft is expected to have moved past the region of turbulence.^b

From the controller’s perspective, we can compute the probability of a wake turbulence event happening in the next 5 minutes at every spatial position. Aggregating all that information, we can overlap trouble spots on the controller’s view of the map. This is shown in Fig. 3 at $t = 220$ s. The yellow region is the region of wake turbulence created by A2 after its missed approach, and the marked red region indicates future locations of a wake turbulence event within the next 5 minutes.

VI. Summary and Future Work

In this paper, we presented a methodology and framework for computing safety metrics, and predicting the occurrence and timing of unsafe events in the NAS. Our approach utilizes a model-based prediction framework that first requires offline analysis and modeling of hazards, safety metrics, and thresholds. The models are then utilized for (online) real-time monitoring and prediction. For robustness to actual operations, which are highly stochastic, the monitoring and prediction algorithms treat uncertainty with mathematical rigor. We demonstrated the full framework through a case study that computes the effects of wake turbulence on airspace safety.

We believe that our real-time monitoring and prediction framework will benefit many of the diverse NAS operators. In future work, this framework can be used for several applications, such as improving shared situational awareness through automated assessment of multiple factors for potential flight routes, minimizing the necessity of in-flight route modification through more informed route selection, and supporting strategic planning between users and the ATC system. Through predictive safety computation that includes rigorous handling of uncertainty, pilots and ATC controllers can receive advance warning of precursors to unsafe events. This enables preemptive actions that aim to avoid unsafe events altogether, rather than having to mitigate them. In addition to the traditional NAS participants, predicted safety metrics could also be incorporated into Unmanned Aerial System (UAS) autonomy and decision-making software, perhaps enabling unmanned aircraft to automatically avoid unsafe NAS regions. In addition, sensitivity analysis techniques can be used to determine the most significant contributors of uncertainty to unsafe events and thus aid in decision-making. Finally, risk analysis methods can be used to incorporate the criticality of different hazards (rather than treating all hazards as being equally critical to safety) and so the overall risk associated with unsafe events can be computed. Although we present the approach here in a centralized manner, it can also be distributed, which will allow the approach to scale easily.²⁴ For example, each aircraft can produce a

^bHere, we used $N = 100$ samples. With more samples, the probability curve in Fig. 5 would be smoother, and the decrease around $t = 200$ s would disappear.

probabilistic trajectory using our framework, and, at a sector level, safety metrics can be computed. This is a function allocation problem²⁵ and is a subject of future work.

Acknowledgements

This work was supported by the SMART-NAS project under the Airspace Operations and Safety (AOSP) program within the NASA Aeronautics Research Mission Directorate (ARMD).

References

- ¹FAA, “Flight Risk Assessment Tool (FRAT),” https://www.faa.gov/other_visit/aviation_industry/airline_operators/airline_safety/info/all_infos/media/2007/info07015.pdf, published 7/3/2007.
- ²FAA, “Safety Management System (SMS),” <http://www.faa.gov/about/initiatives/sms/>, published 1/12/2015.
- ³National Transportation Safety Board, “Aviation Accident Database & Synopses,” http://www.ntsb.gov/_layouts/ntsb.aviation/index.aspx, accessed 9/30/15.
- ⁴NASA, “Aviation Safety Reporting System,” <http://asrs.arc.nasa.gov/>, accessed 9/29/15.
- ⁵Roychoudhury, I., Spirkovska, L., Daigle, M., Balaban, E., Sankararaman, S., Kulkarni, C., Poll, S., and Goebel, K., “Real-Time Monitoring and Prediction of Airspace Safety,” Tech. Rep. NASA/TM-2015-218928, NASA Ames Research Center, Moffett Field, CA, USA, Nov. 2015.
- ⁶Bole, B., Kulkarni, C., and Daigle, M., “Adaptation of an Electrochemistry-based Li-Ion Battery Model to Account for Deterioration Observed Under Randomized Use,” *Annual Conference of the Prognostics and Health Management Society 2014*, September 2014, pp. 502–510.
- ⁷Bole, B., Daigle, M., and Gorospe, G., “Online Prediction of Battery Discharge and Estimation of Parasitic Loads for an Electric Aircraft,” *Second European Conference of the Prognostics and Health Management Society 2014*, July 2014, pp. 23–32.
- ⁸Quach, C., Bole, B., Hogge, E., Vazquez, S., Daigle, M., Celaya, J., Weber, A., and Goebel, K., “Battery Charge Depletion Prediction on an Electric Aircraft,” *Annual Conference of the Prognostics and Health Management Society 2013*, October 2013, pp. 503–512.
- ⁹Daigle, M. and Kulkarni, C., “Electrochemistry-based Battery Modeling for Prognostics,” *Annual Conference of the Prognostics and Health Management Society 2013*, October 2013, pp. 249–261.
- ¹⁰Daigle, M. and Goebel, K., “Model-based Prognostics with Concurrent Damage Progression Processes,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 43, No. 4, May 2013, pp. 535–546.
- ¹¹Balaban, E., Saxena, A., Narasimhan, S., Roychoudhury, I., Koopmans, M., Ott, C., and Goebel, K., “Prognostic Health-Management System Development for Electromechanical Actuators,” *Journal of Aerospace Information Systems*, 2015.
- ¹²Narasimhan, S., Roychoudhury, I., Balaban, E., and Saxena, A., “Combining model-based and feature-driven diagnosis approaches—a case study on electromechanical actuators,” *21st International Workshop on Principles of Diagnosis*, 2010, pp. 1–8.
- ¹³Bilmoria, K. D., Banavar, S., Chatterji, G. B., Sheth, K. S., and Grabbe, S., “FACET: Future ATM concepts evaluation tool,” *3rd USA/Europe ATM R&D Seminar*, June 2000.
- ¹⁴Chatterji, G., Sridhar, B., and Bilmoria, K., “En-route flight trajectory prediction for conflict avoidance and traffic management,” *AIAA Guidance, Navigation, and Control and Conference*, July 1996.
- ¹⁵Tandale, M. D., Wiraatmadja, S., Menon, P. K., and Rios, J. ., “High-Speed Prediction of Air Traffic for Real-Time Decision Support,” *AIAA Guidance Navigation and Control Conference, Portland OR*, 2011, pp. 8–11.
- ¹⁶Julier, S. J. and Uhlmann, J. K., “A new extension of the Kalman filter to nonlinear systems,” *Proceedings of the 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 1997, pp. 182–193.
- ¹⁷Julier, S. J. and Uhlmann, J. K., “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, Vol. 92, No. 3, March 2004, pp. 401–422.
- ¹⁸Daigle, M., Saha, B., and Goebel, K., “A Comparison of Filter-based Approaches for Model-based Prognostics,” *2012 IEEE Aerospace Conference*, March 2012.
- ¹⁹Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, 2002, pp. 174–188.
- ²⁰Sankararaman, S., Daigle, M., and Goebel, K., “Uncertainty Quantification in Remaining Useful Life Prediction using First-Order Reliability Methods,” *IEEE Transactions on Reliability*, Vol. 63, No. 2, June 2014, pp. 603–619.
- ²¹Ding, F., Proctor, F. H., Shen, S., et al., “Numerical modeling studies of wake vortices: real case simulations,” *37th Aerospace Sciences Meeting & Exhibit, AIAA 99-0755*, 1999.
- ²²Holzäpfel, F., “Probabilistic two-phase wake vortex decay and transport model,” *Journal of Aircraft*, Vol. 40, No. 2, 2003, pp. 323–331.
- ²³Holzäpfel, F., “Probabilistic two-phase aircraft wake-vortex model: further development and assessment,” *Journal of Aircraft*, Vol. 43, No. 3, 2006, pp. 700–708.
- ²⁴Daigle, M., Bregon, A., and Roychoudhury, I., “Distributed Prognostics Based on Structural Model Decomposition,” *IEEE Transactions on Reliability*, Vol. 63, No. 2, June 2014, pp. 495–510.
- ²⁵Guerreiro, N. M., Maddalon, J. M., Lewis, T. A., and Hagen, G., “Conflict Detection Performance Analysis for Function Allocation Using Time-Shifted Recorded Traffic Data,” *15th AIAA Aviation Technology, Integration, and Operations Conference*, 2015.