https://ntrs.nasa.gov/search.jsp?R=20180004052 2019-04-29T07:25:40+00:00Z

Real-Time Prediction of Safety Margins in the National Airspace

Matthew Daigle,*
NASA Ames Research Center, Moffett Field, CA, 94035, USA

Indranil Roychoudhury†
Stinger Ghaffarian Technologies Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA

Lilly Spirkovska* and Kai Goebel*
NASA Ames Research Center, Moffett Field, CA, 94035, USA

Shankar Sankararaman† and John Ossenfort† and Chetan Kulkarni†
Stinger Ghaffarian Technologies Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA

Underlying all operations in the National Airspace System (NAS) is the concept of safety. Safety, as defined by acceptable levels of risk, is to be maintained at all times. The real-time safety monitoring (RTSM) framework is under development to provide an automated system to quantify safety in the NAS, estimate the current level of safety, and predict the future evolution of safety and the occurrence of events that pose an increased risk to flights so that these occurrences can be managed strategically rather than mitigated reactively. This paper presents the mathematical framework, the models, and the monitoring and prediction algorithms used to achieve this. RTSM computes safety as expressed through a set of safety margins based on user-defined safety metrics, thresolds, and events. Sources of uncertainty are modeled and propagated through the predictions in order to compute the probabilistic evolution of safety and the probability of events that introduce increased risk to operations. A prototype implementation is discussed and results demonstrating feasibility are presented. The results highlight the kinds of predictions that can be computed and the fidelity that is currently achieved.

I. Introduction

Safety pervades all operations within the National Airspace System (NAS). Flight controllers are tasked with maintaining safety while organizing and expediting the flow of traffic. If there is a conflict between the two tasks, efficiency is sacrificed to ensure an acceptable level of safety. Pilots, airline dispatchers, and all others involved in NAS operations also err on the side of safety when in doubt. To do so effectively requires sufficient awareness of any potential hazards, both current and evolving. Operators expend much effort in remaining cognizant of hazards by gathering, integrating and reasoning over many different information sources. Still, some information is missed due to human cognitive biases, such as confirmation bias in which humans search for, interpret, and focus on information that confirms their preconceptions, ignoring information that disputes their expectations.[1] Other information is missed and safety suffers when situational awareness falters as operators' workload increases.[2,3] Finally, some hazards, even when the various needed data are available to the operator, involve a great deal of analysis to extract useful information, requiring effort that cannot be spared given the operator's other tasks, such as tactical control of traffic or flying the aircraft while also navigating, communicating, and managing aircraft systems. An example of such a situation is maintaining the big picture of traffic and weather under a conceptual NextGen trajectory based operations (TBO) ("free flight") paradigm in which pilots are given more authority over their trajectories.

*Intelligent Systems Division, NASA Ames Research Center
†SGT Inc., Intelligent Systems Division, NASA Ames Research Center

Many diverse approaches have been utilized to improve safety, such as improved aircraft control systems that include envelope protection, regulated duty times, separation standards, and compulsory training requirements. Among such approaches are those that focus on improving aeronautical decision making via improved situational awareness and reduced workload. Toward that end, FAA programs have been developed that encourage a systematic analysis of hazards prior to and during an operation. Examples includes the FAA Flight Risk Assessment Tool,[4] AOPA's Flight Risk Evaluator tool,[5] and the hazard identification processes incorporated in airline Safety Management Systems.[6] Each of these formulates a process for perceiving relevant hazards, determining the risk profile of a operation, and considering risk mitigation strategies. The work associated with perceiving the hazards and determining their cumulative effects remains with the human.

Complementing these manual analyses are a number of tools that have been developed to assist controllers and pilots in maintaining safety. Some of these tools, such as the onboard Traffic Awareness Collision System (TCAS) and the controller's Short-Term Conflict Alert tool, are safety nets that alert the pilot and controller of imminent danger.[7] Other tools can alert about potential issues farther in advance, such as the Traffic Flow Management System (TFMS)[8] Monitor Alert that warns when any one-minute period within a fifteen-minute window is forecast to exceed a preset Monitor/Alert Parameter (MAP) for an airspace sector. Although it has a longer prediction horizon, this tool tends to over-alert because it does not take into account uncertainty in aircraft transitions through the sector; further, a 1-min peak is not necessarily indicative of excessive workload for the full 15-min period.

Still other tools provide situational awareness of only current hazards, such as the dispatcher tools that issue alerts when known conditions pass preset thresholds (e.g., visibility decreases below approach minimums). Although these tools are capable of comparing forecast conditions against preset thresholds, they do not offer any predictive capability. For example, when a new terminal area forecast is issued, the tool can compare the expected visibility values to a preset threshold at a user-specified estimated arrival time. However, it does not use received information, such as winds aloft forecast updates, to modify the estimated arrival time (i.e., predict the new trajectory).

In contrast to these approaches that require the human to manually gather information, integrate diverse sources of information, and predict the evolution of conditions, we have been developing the Real-Time Safety Monitoring (RTSM) system to *automatically* provide safety-relevant information, consolidated in a set of safety metrics. The RTSM concept and sample use cases are described in a companion paper.[9] In this paper, we describe the framework currently under development that offers an approach to assess and predict *in real time* a variety of safety metrics. In previous papers, we documented the selection of safety metrics,[10] the RTSM framework applied to predicting safety metrics,[11] and an initial implementation connected to real-time surveillance and weather data.[12] In this paper, we describe how the framework can be applied to determining *safety margins*, that is, the difference between the current situation and the minimum acceptable situation. The "situation" can be an individual safety metric or an aggregate of multiple safety metrics. The predictive aspect of RTSM may promote further proactive management of risks and lessen the impact on efficiency, while the safety margin aspect may lessen information gathering workload and decrease the likelihood of human error due to missed information.

The paper is organized as follows. Section II develops the mathematical framework underlying the approach and formulates the problem. Section III describes the system architecture for our approach. Section IV describes the safety models and demonstrates the approach. Section V concludes the paper.

II. Problem Formulation

In this work, we are interested in measuring and predicting the safety of the NAS. At a fundamental level, there are some states within the NAS that are considered to be "unsafe" in some way, meaning, the risk associated with these states is unacceptably high. For example, a state in which a small aircraft is encountering wake turbulence from a much larger aircraft is to be avoided because of a high potential for loss of control by the following aircraft. The determination as to which states belong to which set are provided as input to our framework, and are determined externally via safety and risk analysis methods. Such analysis usually leads to the definition of "thresholds," such that states that exceed these thresholds are to be avoided. The thresholds are functions of relevant quantities of interest, such as separation between aircraft, number of aircraft in a sector, etc. For example, in the enroute airspace there is a horizontal separation standard of 5 nautical miles. In our framework, we adopt those thresholds defined in FAA regulations, best practices,

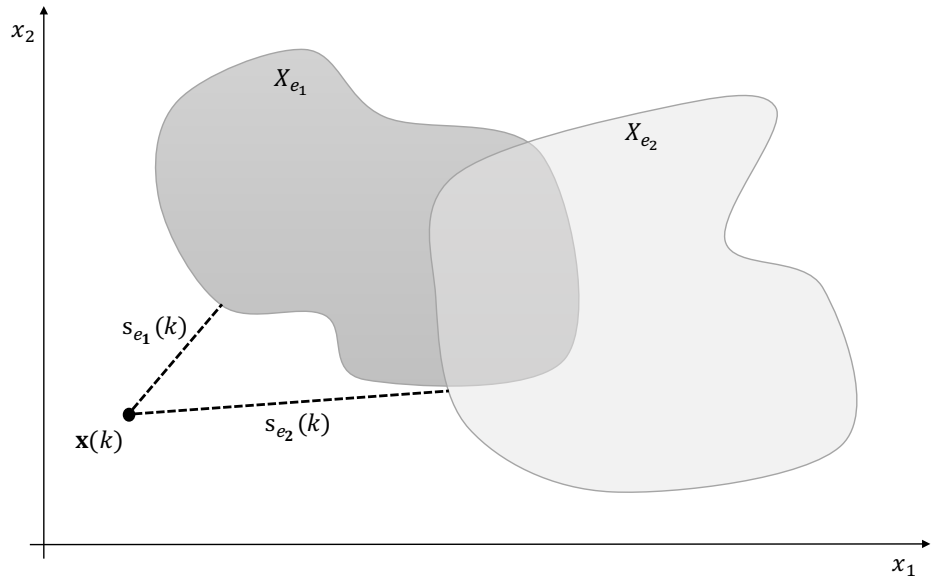

Figure 1. Two-dimensional state space with events e_1 and e_2.

etc., which are derived from such analyses. Here, safety is then expressed quantitatively in a relative manner through a (usually non-Euclidean) distance measure to these thresholds.

In mathematical terms, given a *state* of the NAS, $\mathbf{x} \in \mathbb{R}^{n_x}$, we define a set of events E that apply to the state space.[13] The events may indicate conditions such as loss of separation between two aircraft, congestion in a sector, etc. For each event $e \in E$, based on its corresponding safety thresholds, we introduce a Boolean detection function, $\mathbf{detect}_e : \mathbb{R}^{n_x} \to \{true, false\}$, that maps a given state to true if the event e applies to a given state $\mathbf{x}$, and false otherwise:

$$o_e(k) = \mathbf{detect}_e(\mathbf{x}), \tag{1}$$

where $o_e(k)$ is a Boolean variable indicating the occurrence of e in a given state. Using the detection functions, we can define corresponding subsets of the state space, i.e., for an event e, we define $\mathcal{X}_e = \{\mathbf{x} : \mathbf{detect}_e(\mathbf{x}) = true\}$. Note that many events may apply to the same state.

For example, consider Fig. 1. Here, we have a simplified state space with two states, x_1 and x_2, and two events, e_1 and e_2. In some regions of the state space, e_1 is detected, defining $\mathcal{X}_{e_1}$, and in others, e_2 is detected, defining $\mathcal{X}_{e_2}$. The boundaries of these regions are defined by corresponding safety thresholds. States inside $\mathcal{X}_{e_1} \cup \mathcal{X}_{e_2}$ have an unacceptable level of perceived risk, and all other states have an acceptable level.

In this framework, safety, with respect to an event e, is expressed as a measure of *distance* from the current state to $\mathcal{X}_e$. This distance is computed through a set of *safety metrics*, $\mathbf{m}_e$, that are dependent on the NAS state:

$$\mathbf{m}_e(k) = \mathbf{metrics}_e(\mathbf{x}(k)), \tag{2}$$

where $\mathbf{metrics}_e$ is the metrics function for event e. That is, we express distance in the metrics space, rather than the state space. The *safety margin*, $\mathbf{s}_e$, is that measure of distance, and is expressed as the *difference* between the metrics computed at the current state and the nearest state in $\mathcal{X}_e$:

$$s_e(k) = \mathbf{margin}(\mathbf{x}(k)) = \min\{\mathbf{metrics}_e(\mathbf{x}(k)) \ominus_e \mathbf{metrics}_e(\mathbf{x}_e(k)) : \mathbf{x}_e(k) \in \mathcal{X}_e\}, \tag{3}$$

where $\ominus_e$ is the difference operator for event e. For different events, this difference operator may be implemented differently. Typically, states on the border of $\mathcal{X}_e$ correspond to a threshold in the metrics space, and so this difference is easily computed. For example, horizontal separation between two aircraft is a simple metric, and for an event such as loss of separation, all states in $\mathcal{X}_e$ would have a horizontal separation of less than 5 nautical miles for the enroute airspace. So if the current separation is 8 nautical miles, the margin is

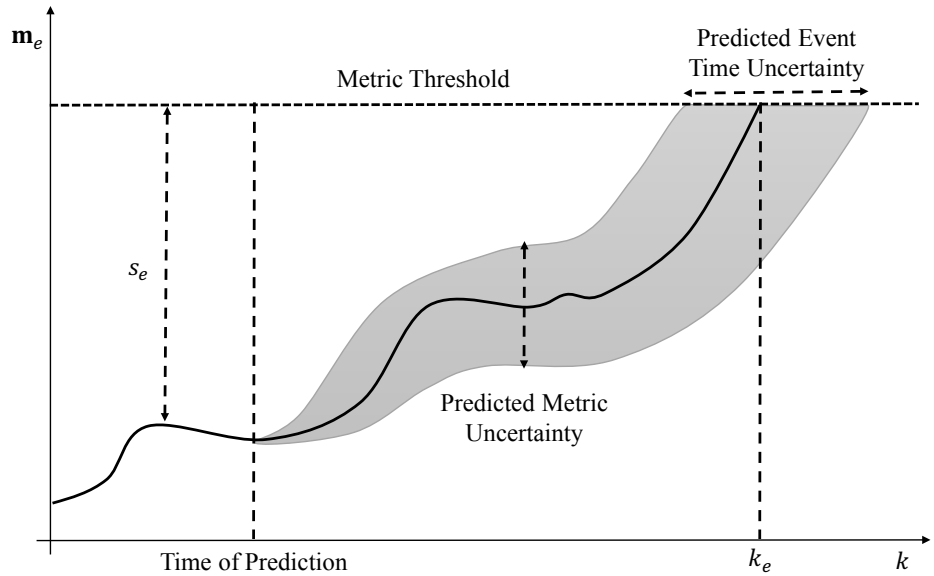

Figure 2. One-dimensional safety metric example for event e.

3 nautical miles. Consider again Fig. 1. The current state is $\mathbf{x}(k)$. The minimum distance to the boundaries of the two unsafe regions define the two safety margins, s_{e_1} and s_{e_2}.

In some cases, it is more convenient to express the safety margin as a percentage in $[0, 100]\%$, such that a 0% safety margin corresponds to being within $\mathcal{X}_e$, and 100% corresponds to begin far enough away from $\mathcal{X}_e$ relative to the threshold that defines $\mathcal{X}_e$. So, in the previous example, the margin would be $3/5 = 60\%$. If the margin is greater than 5 nautical miles, we limit the relative margin to 100%.

Given this definition of safety, as expressed through safety metrics and margins, we can define the monitoring and prediction problems. The *monitoring* problem is to determine, for the current NAS state, the values of all the safety metrics and margins and whether the current state belongs to any subset of the NAS state space where an event is present. The *prediction* problem is to determine for all future NAS states, up to a finite prediction horizon, all the safety metrics and margins and whether (and when) any future state belongs to any subset of the NAS state space where an event is present.

The time of an event is denoted using k_e. For an event e, we define k_e as follows:

$$k_e(k) = \min\{k' : k' \geq k \textbf{ and detect}(\mathbf{x}(k')) = \mathit{true}\}, \tag{4}$$

i.e., k_e is the earliest time point at which the event e is detected for the system state. This result provides an estimate of when the safety margin would be depleted without any intervening operator action.

Fig. 2 shows a simplified one-dimensional metric example, where the threshold is a constant. The value of the metric is tracked in time, and at the prediction time, the evolution of the safety metric (as computed from the predicted NAS state trajectory) is computed. Within the prediction horizon, the event occurs (i.e., the value of the metric exceeds the threshold) and due to the predicted uncertainty in the metric (which, in turn, is due to the uncertainty in the predicted NAS state), this results in a probability distribution for the event time k_e.

As Fig. 2 demonstrates, uncertainty is inherent to both the *monitoring* and *prediction* problems. Due to various sources of uncertainty (e.g., surveillance precision and update rate, wind velocity, etc), the current NAS state is not known precisely and so the current and future safety cannot be determined precisely. Therefore, probabilistic tools are used to represent and estimate the uncertainty in the current NAS state, and predict the uncertainty in the current and future safety of the NAS. This is achieved by estimating the probability density function of the current NAS state, and predicting the probability density function of the safety metrics and margins at all desired time instants (i.e., until the desired time horizon). In addition, the probability that the NAS is in a state where an event of interest has occurred is also computed; note that this is a discrete event and hence the probability (as against probability density for continuously varying uncertain quantities) is used. Therefore, monitoring and prediction problems can be formally described as

follows.

Problem 1 (Monitoring)**.** At time k, the *monitoring* problem is to determine

- the probability distribution of the current NAS state, $p(\mathbf{x}(k))$,
- the probability distribution of the safety metrics $p(\mathbf{m}_e(k))$ for all events $e \in E$,
- the probability distribution of the safety margins $p(s_e(k))$ for all events $e \in E$,
- and the probability that the state is in $\mathcal{X}_e$, $P(\mathbf{x}(k) \in \mathcal{X}_e)$ for all events $e \in E$.

Problem 2 (Prediction)**.** At time k_o, given the probability distribution of the current NAS state, $p(\mathbf{x}(k_o))$, and a prediction horizon $k_h > k_o$, the *prediction* problem is to determine for each $e \in E$:

- the probability distribution of the safety metrics $p(\mathbf{m}_e(k))$ for each $k \in (k_o, k_h]$,
- the probability distribution of the safety margins $p(s_e(k))$ for each $k \in (k_o, k_h]$,
- the probability that the state is in $\mathcal{X}_e$, $P(\mathbf{x}(k) \in \mathcal{X}_e)$ over the interval $(k_o, k_h]$,
- and the probability distribution of the timing of an event $p(k_e(k))$.

III. System Architecture

We adopt a model-based approach to solve the monitoring and prediction problems presented in the previous section. This means that we have to define the state of the NAS, the events, the detection functions, the metrics, the metrics functions, and the margin functions. Further, we need dynamic models that describe how the NAS state evolves in time, given the inputs to the NAS. Based on the given NAS data and the developed models, we can estimate the current state of safety in the NAS and predict its evolution with considered sources of uncertainty.

III.A. NAS Modeling

The NAS, as a dynamic system, consists mainly of aircraft, weather systems, and human operators. In this work, we are interested in how the states of aircraft evolve given assumed operator behavior, and how the states of weather systems evolve. Thus, we do not include dynamic models of operator behavior and do not consider any safety metrics associated with internal operator states (e.g., workload, fatigue).

So, in this context, the NAS state is defined solely by the state of the individual aircraft and discrete weather systems.[a] The NAS as a whole must then be described using a state equation:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{v}(k)), \tag{5}$$

where $k \in \mathbb{N}$ is the discrete time variable, $\mathbf{x}(k) \in \mathbb{R}^{n_x}$ is the state vector, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{v}(k) \in \mathbb{R}^{n_v}$ is the process noise vector, and $\mathbf{f}$ is the state update function. Since the dynamics of aircraft and weather systems are independent of each other, $\mathbf{f}$ can be described using a set of independent state equations, where for each aircraft a we have:

$$\mathbf{x}_a(k+1) = \mathbf{f}_a(\mathbf{x}_a(k), \mathbf{u}_a(k), \mathbf{v}_a(k)), \tag{6}$$

and for each weather system w we have:

$$\mathbf{x}_w(k+1) = \mathbf{f}_w(\mathbf{x}_w(k), \mathbf{u}_w(k), \mathbf{v}_w(k)). \tag{7}$$

[a] In this context, we consider wind to be an independent input to both aircraft and weather systems.

So, the NAS state $\mathbf{x}(k)$ is:

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}_{a_1}(k) \\ \mathbf{x}_{a_2}(k) \\ \vdots \\ \mathbf{x}_{a_n}(k) \\ \mathbf{x}_{w_1}(k) \\ \mathbf{x}_{w_2}(k) \\ \vdots \\ \mathbf{x}_{w_m}(k) \end{bmatrix}, \tag{8}$$

for n aircraft and m weather systems.

The aircraft and weather models currently implemented are described in our previous manuscripts.[10,13] They are based on those implemented in FACET.[14]

III.B. Monitoring

To solve the monitoring problem, we use a Bayesian filtering approach, utilizing the unscented Kalman filter (UKF).[15] For each aircraft and weather system, we create a UKF with the respective dynamic model. The UKF requires an output equation as part of the model for each aircraft a and weather system w:

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

$$\mathbf{y}_w(k) = \mathbf{h}_w(\mathbf{x}_w(k), \mathbf{u}_w(k), \mathbf{n}_w(k)), \tag{10}$$

where $\mathbf{y}_{a(w)} \in \mathbb{R}^{n_{y_{a(w)}}}$ is the output vector (i.e., the vector of measured values), $\mathbf{n}_{a(w)} \in \mathbb{R}^{n_{n_{a(w)}}}$ is the sensor noise vector, and $\mathbf{h}_{a(w)}$ is the output function.

The UKF maintains the estimated state and output vectors for a given model. It uses the state equation to update the state estimate one time step ahead, uses the output equation to obtain the predicted outputs, then corrects the state estimate given the measured outputs using Bayesian filtering theory, to obtain $p(\mathbf{x}_{a(w)})$ for each aircraft (weather system). More details on the UKF can be found in the literature.[10,15,16] Given the probability distributions for all aircraft and weather states, we can construct the probability distribution for the NAS state as a whole, $p(\mathbf{x}(k))$, assuming each individual distribution is independent from each other.

Using Eq. 2 and 3, we can transform $p(\mathbf{x}(k))$ into $p(\mathbf{m}_e(k))$ and $p(\mathbf{s}_e(k))$, respectively, for each event e. We can then also determine $P(\mathbf{x} \in \mathcal{X}_e)$, given $p(\mathbf{x}(k))$. Thus, for the current time step, once we estimate the state of the NAS, we can compute the probability distribution for safety metrics and safety margins for all events, and the probablity of encountering each event at any time point and within the entire prediction horizon.

III.C. Prediction

To solve the prediction problem, we use methods of uncertainty propagation.[13,17] The inputs to the uncertainty propagation problem are the following:[b]

1. a time of prediction, k_o;
2. a time interval of prediction, $[k_o, k_h]$;
3. the initial state probability distribution, $p(\mathbf{x}_o(k_o))$;
4. the future input trajectory distribution, $p(\mathbf{U}_{k_o,k_h})$, where $\mathbf{U}_{k_o,k_h} = [\mathbf{u}(k_o), \mathbf{u}(k_o+1), \ldots, \mathbf{u}(k_h)]$; and
5. the future process noise trajectory distribution, $p(\mathbf{V}_{k_o,k_h})$, where $\mathbf{V}_{k_o,k_h} = [\mathbf{v}(k_o), \mathbf{v}(k_o+1), \ldots, \mathbf{v}(k_h)]$.

[b]Note that the following notation is adopted. For a vector $\mathbf{a}$, a trajectory of that vector over a time interval $[k_o, k_h]$ is denoted by $\mathbf{A}_{k_o,k_h}$, and $\mathbf{A}_{k_o,k_h}(k) = \mathbf{a}(k)$, for $k \in [k_o, k_h]$.

Algorithm 1 `Simulate`

Inputs: $k_o, k_h, \mathbf{x}(k_o), \mathbf{U}_{k_o,k_h}, \mathbf{V}_{k_o,k_h}$
Outputs: $\mathbf{X}_{k_o,k_h}$
$\mathbf{X}_{k_o,k_h}(k_o) \leftarrow \mathbf{x}(k_o)$
for $k = k_o$ **to** $k_h - 1$ **do**
  $\mathbf{X}_{k_o,k_h}(k+1) \leftarrow \mathbf{f}(\mathbf{X}_{k_o,k_h}(k), \mathbf{U}_{k_o,k_h}(k), \mathbf{V}_{k_o,k_h}(k))$
end for

Algorithm 2 `MonteCarloPrediction`

1: **Inputs:** $k_o, k_h, p(\mathbf{x}(k_o)), p(\mathbf{U}_{k_o,k_h}), p(\mathbf{V}_{k_o,k_h})$
2: **Outputs:** $\{\mathbf{X}^i_{k_o,k_h}\}_{i=1}^N$
3: $\{(\mathbf{x}(k_o)^i, \mathbf{U}^i_{k_o,k_h}, \mathbf{V}^i_{k_o,k_h})\}_{i=1}^N \leftarrow$ `MonteCarloSampling` $(N, p(\mathbf{x}(k_o)), p(\mathbf{U}_{k_o,k_h}), p(\mathbf{V}_{k_o,k_h}))$
4: **for** $i = 1$ **to** N **do**
5: $\mathbf{X}^i_{k_o,k_h} \leftarrow$ `Simulate`$(k_o, k_h, \mathbf{x}(k_o)^i, \mathbf{U}^i_{k_o,k_h}, \mathbf{V}^i_{k_o,k_h})$
6: **end for**

These inputs may change for every new time of prediction. Note that we require some estimate of the future inputs to the system. For aircraft, this reduces to intent information and environmental conditions such as wind (see[18] for details). For the weather, this reduces to only environmental conditions.

Currently, we use a Monte Carlo-based algorithm for prediction that uses the system model.[13] The core simulation algorithm is given as Algorithm 1. For a given NAS state, time of prediction, prediction horizon, future input trajectory, and future process noise trajectory, the state is predicted ahead for each future time point to form the predicted future state trajectory.

The Monte Carlo prediction algorithm is given as Algorithm 2. Given the probability distribution of the state at the time of prediction, as computed in the monitoring step, and assumed probability distributions for the future input and process noise trajectories, the algorithm generates N samples from these distributions and calls the `Simulate` function for each to obtain a set of state trajectory samples, $\{\mathbf{X}^i_{k_o,k_h}\}_{i=1}^N$, which is an approximation of $p(\mathbf{X}_{k_o,k_h})$. Note that in line 3, alternative sampling algorithms could also be substituted.

From the state trajectory samples, $\{\mathbf{X}^i_{k_o,k_h}\}_{i=1}^N$, we can then compute the metrics and margin trajectories, and compute the event probabilities. This computation is shown as Algorithm 3. The probability of an event occurring at any given time can be computed from the occurrence trajectory, $\mathbf{O}_{e,k_o,k_h}$, as the number of samples for which this is true at the given time divided by N. The probability of any event occurring within the prediction horizon is simply the number of finite k^i_e values divided by N.

Algorithm 3 `ComputeSafety`

Inputs: $e, \{\mathbf{X}^i_{k_o,k_h}\}_{i=1}^N$
Outputs: $\{\mathbf{M}^i_{e,k_o,k_h}\}_{i=1}^N, \{\mathbf{S}^i_{e,k_o,k_h}\}_{i=1}^N, \{\mathbf{O}^i_{e,k_o,k_h}\}_{i=1}^N, \{k^i_e\}_{i=1}^N$
for all $i = 1$ **to** N **do**
  $k^i_e \leftarrow \infty$
end for
for all $\mathbf{X}^i_{k_o,k_h} \in \{\mathbf{X}^i_{k_o,k_h}\}_{i=1}^N$ **do**
  for all $k \in [k_o, k_h]$ **do**
    $\mathbf{M}^i_{e,k_o,k_h}(k) = \mathbf{metrics}_e(\mathbf{X}^i_{k_o,k_h}(k))$
    $\mathbf{S}^i_{e,k_o,k_h}(k) = \mathbf{margin}_e(\mathbf{X}^i_{k_o,k_h}(k))$
    $\mathbf{O}^i_{e,k_o,k_h}(k) = \mathbf{detect}_e(\mathbf{X}^i_{k_o,k_h}(k))$
    if $k_e = \infty$ **and** $\mathbf{O}^i_{e,k_o,k_h}(k) = \mathit{true}$ **then**
      $k^i_e \leftarrow k$
    end if
  end for
end for

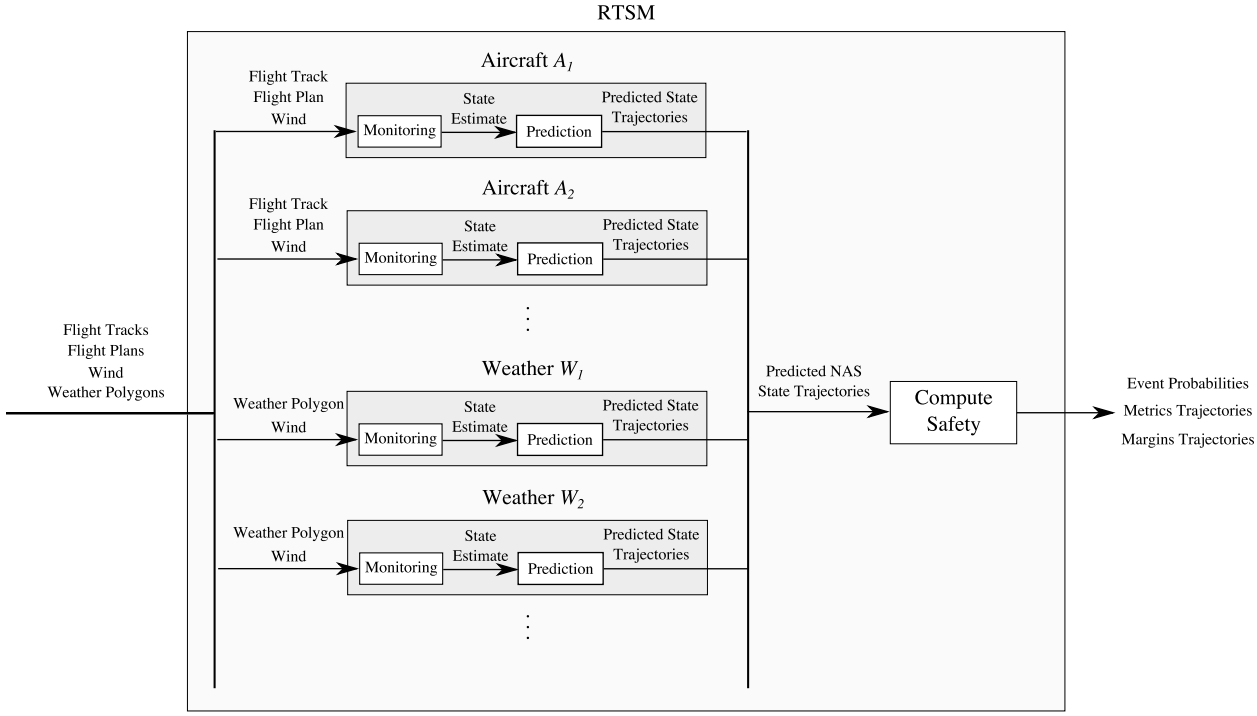

Figure 3. Computational architecture of RTSM.

III.D. Computational Architecture

The computational architecture of the RTSM system is depicted in Fig. 3. First, data from the NAS (flight tracks, flight plans, weather polygons, and wind) is pre-processed to account for missing data, time-order the data, and separate into data streams for individual aircraft and weather objects. When new aircraft or weather systems are recognized in the airspace, new objects are spawned. When they leave the airspace (e.g., an aircraft lands), their corresponding objects are deleted.

For each active aircraft a and weather object w, RTSM first estimates their state, $\mathbf{x}_a$ and $\mathbf{x}_w$, using the UKF. For aircraft, track data, flight plans, and wind are consumed and for the weather systems, wind data and weather polygon data are consumed. For each aircraft/weather, the prediction algorithm (Algorithm 2) is used to generate predicted state trajectories, $\{\mathbf{X}_a^i\}_{i=1}^N$ and $\{\mathbf{X}_w^i\}_{i=1}^N$.

The safety computation is then performed given the predicted state trajectories. For every event $e \in E$, the safety metrics and margins are computed along with the event probabilities using Algorithm 3. RTSM can be easily extended by adding new events to E, and defining the corresponding metrics and margins functions.

Although Fig. 3 may imply a single, centralized architecture, some distribution of the computation is possible. The monitoring/prediction modules for each aircraft/weather are computationally independent, and so can be executed in parallel on separate computational units. Within the prediction algorithm, each realization of the state trajectory can be computed in parallel (e.g., using a GPU). The computations for each event $e \in E$ can also be done in parallel. Such parallelization enables a high degree of scalability.

It should also be noted that different pieces of RTSM can be located in different places. For example, each aircraft in the airspace can be responsible for producing its own trajectory predictions, which can then be handled through a cloud service that implements RTSM.

IV. Application

In this section, we demonstrate the framework with a set of four event categories. We first describe the models, followed by specific results obtained by applying the framework with real data.

IV.A. Safety Modeling

We consider four different kinds of events: (1) loss of separation, (2) wake vortex encounter, (3) convective weather encounter, and (4) sector over-demand. The first two potentially apply to any pair of aircraft, the third to any aircraft-weather pair, and the fourth to a single sector.

Loss of separation (LOS) occurs when both horizontal and vertical separation minimums established by the FAA are not met. Any pair of aircraft can potentially be in conflict, so for a region of airspace with A aircraft, there are $\binom{A}{2} = (A-1)\cdot A/2$ potential events. Two metrics considered in this case are the horizontal and vertical separations:

$$\mathbf{metrics}_{\mathrm{LOS}_{a_i,a_j}}(\mathbf{x}) = \begin{bmatrix} h_{sep}(\mathbf{x}_{a_i}, \mathbf{x}_{a_j}) \\ v_{sep}(\mathbf{x}_{a_i}, \mathbf{x}_{a_j}) \end{bmatrix}, \tag{11}$$

where LOS_{a_i,a_j} denotes the LOS between aircraft a_i and a_j, h_{sep} is a function that computes horizontal separation between two aircraft using great-circle distance, and v_{sep} is a function that computes the absolute altitude difference between two aircraft. The margin, as a percentage, is then computed as:

$$\mathbf{margin}_{\mathrm{LOS}_{a_i,a_j}}(\mathbf{x}) = \min\left\{\frac{\mathbf{m}_{\mathrm{LOS}_{a_i,a_j}}[0] - T_h}{T_h}, \frac{\mathbf{m}_{\mathrm{LOS}_{a_i,a_j}}[1] - T_v}{T_v}\right\}, \tag{12}$$

where T_h is the minimum horizontal separation (e.g., 5 nautical miles for en-route airspace), and T_v is the minimum vertical separation (e.g., 1000 ft). A meta-event representing the situation where any aircraft pair is in conflict is also introduced. The corresponding detection function returns true when any of the individual detection functions for aircraft pairs returns true. In this case, the margin is computed as the average of all individual margins. The minimum margin is also computed.

A wake vortex encounter (WAKE) occurs when one aircraft enters the wake vortex region of a leading aircraft. Any one aircraft can potentially be in the wake of another, so there are $A\cdot(A-1) = A^2 - A$ potential wake events for an airspace containing A aircraft. A polygon representing the region of a wake vortex tailing an aircraft is computed as described in our previous manuscript,[10] and is based on the aircraft's current and past position over a specified time window (e.g., 2 minutes). As with LOS, we compute both horizontal and vertical separation, this time between one aircraft and the wake polygon of another. The margin is then computed as

$$\mathbf{margin}_{\mathrm{WAKE}_{a_i,a_j}}(\mathbf{x}) = \min\left\{\frac{\mathbf{m}_{\mathrm{WAKE}_{a_i,a_j}}[0]}{T_h}, \frac{\mathbf{m}_{\mathrm{WAKE}_{a_i,a_j}}[1] - T_v}{T_v}\right\}, \tag{13}$$

where $WAKE_{a_i,a_j}$ denotes the event where aircraft a_i is in the wake of a_j. Typically here, T_v is also 1000 ft. T_h can be determined from wake separation standards that depend on aircraft size. Like with LOS, we define here also a meta-event that is detected when any single wake event is detected, and compute the overall margin as the average of the individual margins.

A convective weather encounter (WX) occurs when an aircraft reaches within a minimum specified horizontal distance from the nearest edge of a weather polygon. For an airspace with A aircraft and W weather, there are $A \cdot W$ potential events. The single metric is then:

$$\mathbf{metrics}_{\mathrm{WX}_{a_i,w_j}}(\mathbf{x}) = \left[h_{sep}(\mathbf{x}_{a_i}, \mathbf{x}_{w_j})\right], \tag{14}$$

where WX_{a_i,w_j} denotes a convective weather encounter between aircraft a_i and weather w_j, and h_{sep} computes the minimum great-circle distance between an aircraft and a weather polygon. The margin is then computed as:

$$\mathbf{margin}_{\mathrm{WX}_{a_i,w_j}}(\mathbf{x}) = \frac{\mathbf{m}_{\mathrm{WX}_{a_i,w_j}} - T_w}{T_w}, \tag{15}$$

where T_w is the required separation threshold (e.g., 20 nautical miles). Like with the previous event categories, we define also a meta-event that is detected when any individual convective weather encounter is detected, and compute the overall margin as the average of the individual margins.

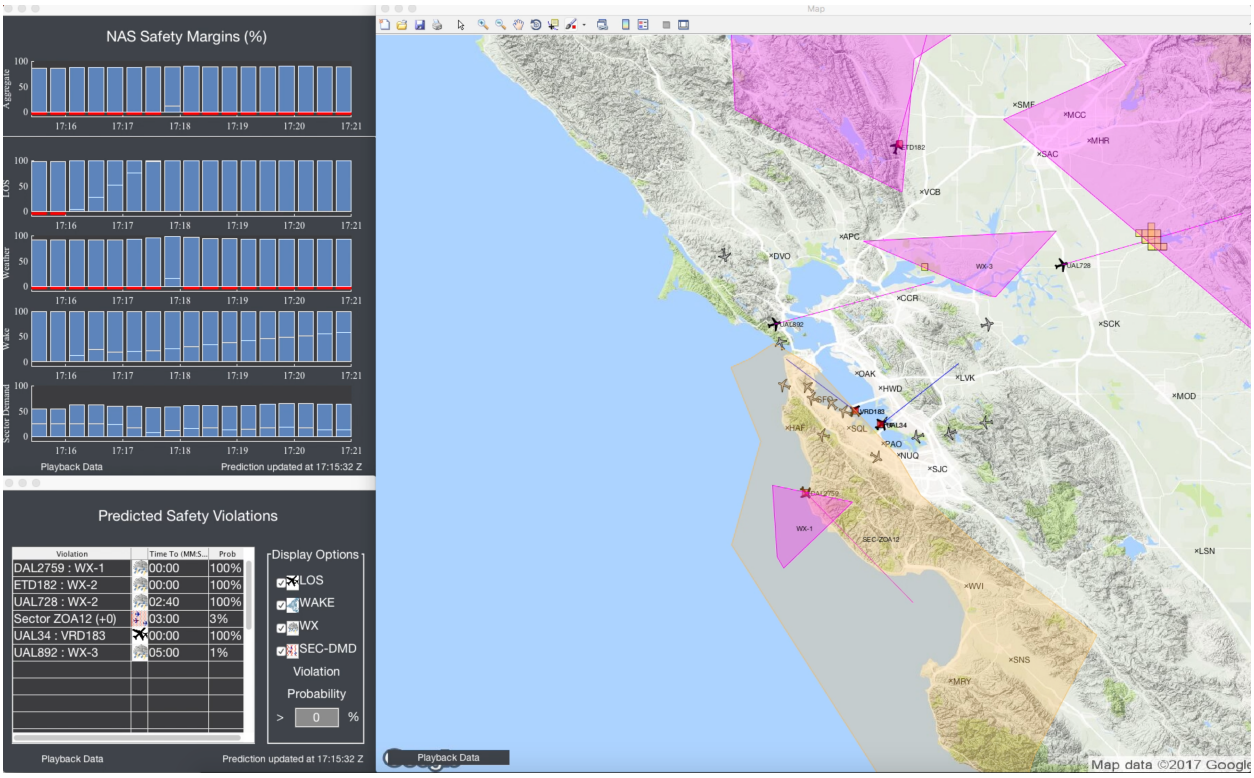

Figure 4. RTSM display, showing safety margins, predicted events, and regions of interest.

A sector over-demand (SEC-DMD) event occurs when a sector contains as many or more than its default capacity of aircraft. For an airspace with S sectors, there are S potential events. The single metric is the aircraft count in a given sector, and the margin (as a percentage) for sector s is computed as:

$$\mathbf{margin}_{\text{SEC-DMD}_s}(\mathbf{x}) = \frac{C_s - \mathbf{m}_{\text{SEC-DMD}_s}}{C_s}, \tag{16}$$

where SEC-DMD$_s$ is the over-demand event for sector s, and C_s is the default capacity for sector s. We define also a meta-event in this case.

Over all event categories, we define also a final meta-event that is detected when any of the considered events is detected. The margin is computed as the average of the individual meta-event margins. In a sense, this event represents anything unsafe happening in the given airspace region, and the associated margin represents an overall measure of the region's safety.

IV.B. User Display

An overview of the current prototype RTSM display is shown in Fig. 4. In the upper-left, the safety margins are shown, with the top plot being the overall safety margin, and the remaining being the safety margins for the different event categories (i.e., associated with the meta-events). Both the average safety margin (top of the blue bar) and the minimum safety margin over all constituent events (the white line in the middle of the bar, or the red line at the bottom if zero). The leftmost bar is the safety margin for the current time, and the following bars are a prediction of how the average and minimum safety margins will evolve in the future. So for example, we can see that overall safety is high, but there are some safety margins that are zero, meaning events are being predicted for that time. For the current time (leftmost bar) this is due to LOS and weather, and in the future (bars to the right) this is due mainly to weather events.

The bottom-left window shows the predicted events. Currently, we see that three different event types are predicted (weather, sector over-demand, and LOS).[c] Information on the specific event predicted (e.g.,

[c]Note that the safety thresholds were increased (e.g., a larger than standard aircraft separation was used) in order to have multiple events occurring in order to explain the display. In reality, if operations are safe, there should be much less occurring.

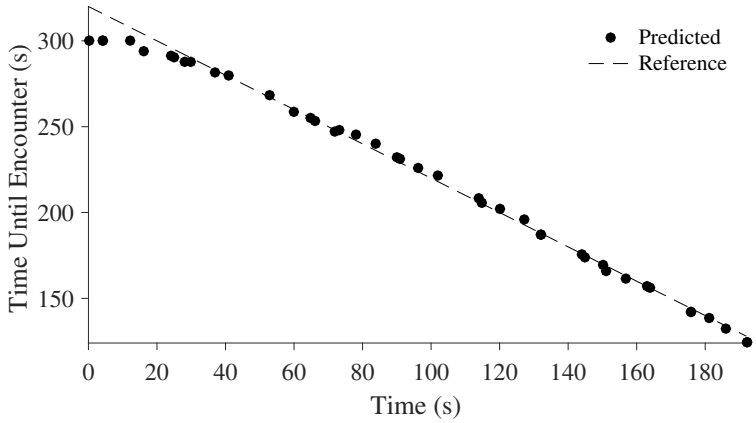

Figure 5. Predicted mean time until weather encounter, as a function of time.

the aircraft involved in a LOS), the (average) predicted time until the event, and the probability of the event occurring within the prediction horizon (here, 5 minutes) are all displayed. For example, there is a 100% probability of a weather encounter occurring between DAL2759 and WX-1 in 0 minutes, i.e., the event is detected as happening right now. In 3 minutes, RTSM predicts a sector over-demand for ZOA12 with a small (3%) probability. A user can select which event types to display and also filter by probability, so that the user is not overwhelmed with many low-probability events.

The map display shows event information geographically. Regions of convective weather are shown with magenta-colored polygons, and sectors of predicted over-demand are shown as orange-colored polygons. All known aircraft in the region of airspace are shown. If an aircraft is associated with a particular event, it is colored black, its predicted trajectory is shown, and the predicted location of the aircraft when the event is to occur is shown using yellow, orange, and red grid squares, which roughly indicate the probability of the aircraft being in a specific grid square (yellow is lowest, red is highest, and the probability intervals associated with the different colors are configurable). This gives a user a visual indication of where an event is to occur to aid in planning activities to prevent the event. For example, UAL728 is headed towards a region of convective weather, and the yellow-colored area is where the aircraft is expected to be when it gets an unsafe distance from the weather (for demonstration purposes, we use 5 nautical miles here instead of the typical 20 nautical miles).[d]

IV.C. Sample Results

To demonstrate the prediction capability, consider a convective weather encounter (UAL728 with WX-2 in Fig. 4). At each time point, the future aircraft and weather trajectories are predicted, and based on these, the distribution of encounter times and the probability of encounter within the prediction interval of 5 minutes are computed.

Fig. 5 shows the predicted mean time until the weather encounter at each time of prediction. Note that along the x-axis, the times provided are relative to the first time of prediction, which is assigned a value of 0. The true time of weather encounter is unknown, because the aircraft leaves the region being monitored before the event happens. The true time until encounter will decrease with a slope of 1, and this is shown with the reference line assuming a true event time of 320 s. We can see that after the first 20 s, in which the prediction is very uncertain (see the probability in Fig. 6), the predictions are very consistent, and after 40 s, are associated with a probability near 100%. That is, about 5 minutes before the event is thought to occur, RTSM consistently predicts the time of the event with high confidence.

The safety margin as computed within a 15 minute interval is shown in Fig. 7. Here, we see that on average, the safety margin is quite high, however there are specific instances of unsafe events causing the minimum safety margin to become low in some points. In the first few minutes, there are both LOS and weather events occurring, resulting in a low margin. About 5 minutes in, a sector demand event occurs,

[d]Note that the weather polygons used here are for demonstration purposes only and do not reflect actual convective weather from this data set, which explains why some aircraft are inside the polygons or headed directly into them.

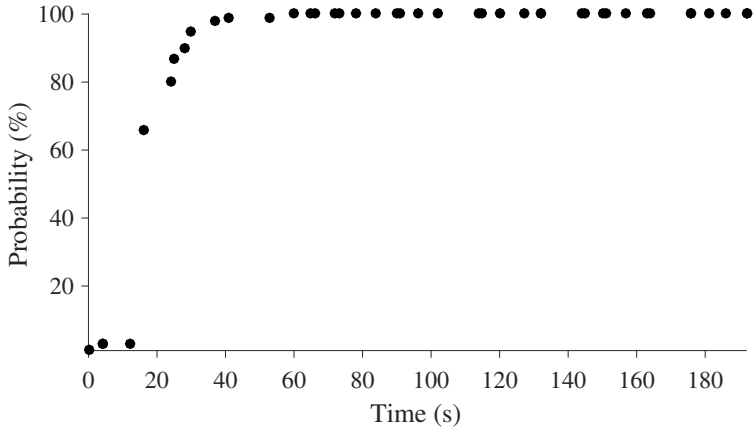

Figure 6. Probability of weather encounter in the next 5 minutes, as a function of time.

resulting in the low margin, followed by additional weather events. In the remaining time, there are some sporadic LOS events contributing low margins.

IV.D. Discussion

Although the predictions in Fig. 5 are very accurate, not all predictions achieve the same level of accuracy, for several reasons. First, in order to make accurate aircraft trajectory predictions, flight plans are needed. However, through the current data sources used as input, flight plans are not always available. In the case that the flight plan for an aircraft is not available, we assume that within the prediction horizon the flight will maintain its current speed, heading, and altitude. This assumption is not always valid, especially for aircraft near the airport, where departure and arrival procedures are likely to result in changes to all three of these parameters. Thus, the trajectories will not be accurately predicted, and may result in false positives and false negatives with respect to the prediction of events.

Second, the data sources are not always reliable. In the examples abvove, the region considered was a 50 nautical mile radius from SFO. Ideally, we would like to get data for aircraft (i) as soon as they are on the runway, so that they can be tracked throughout their takeoff and departure procedure, and (ii) as soon as they fly into our region of interest from outside the area. Sometimes, the first data obtained for aircraft is when they are somewhere in the middle of the region of interest, and this can result in sudden event predictions. Overall, the predictions made by RTSM are only as reliable as the data coming into the system.

V. Conclusions

The real-time safety monitoring framework is a model-based framework for quantifying safety in the NAS as expressed through safety metrics and margins, and for both monitoring the current state of safety and predicting the future state of safety while taking into account uncertainty. RTSM was demonstrated on real NAS data from the SFO region, demonstrating the kind of information that can be presented to a user of RTSM and showing the fidelity of the predictions that can be achieved.

Currently, RTSM focuses on a small set of safety events and metrics to assess the NAS. Ongoing work involves adding to this list. Further, depending on the user of RTSM (e.g., pilot vs. air traffic controller), a different set of safety events and metrics may be useful and current work involves customizing the framework to different users. Current work also involves development of a more distributed, scalable implementation of the framework.

This work focused on predicting how the state of the NAS evolves given assumed operator behavior. If this results in an undesirable level of safety/risk then it implies different behavior is required by the operators. In this way, our system can be used within a decision-making framework. RTSM can be used to then predict and evaluate the safety/risk of alternative behaviors. Ongoing work involves integrating this kind of information into automated decision-making algorithms.

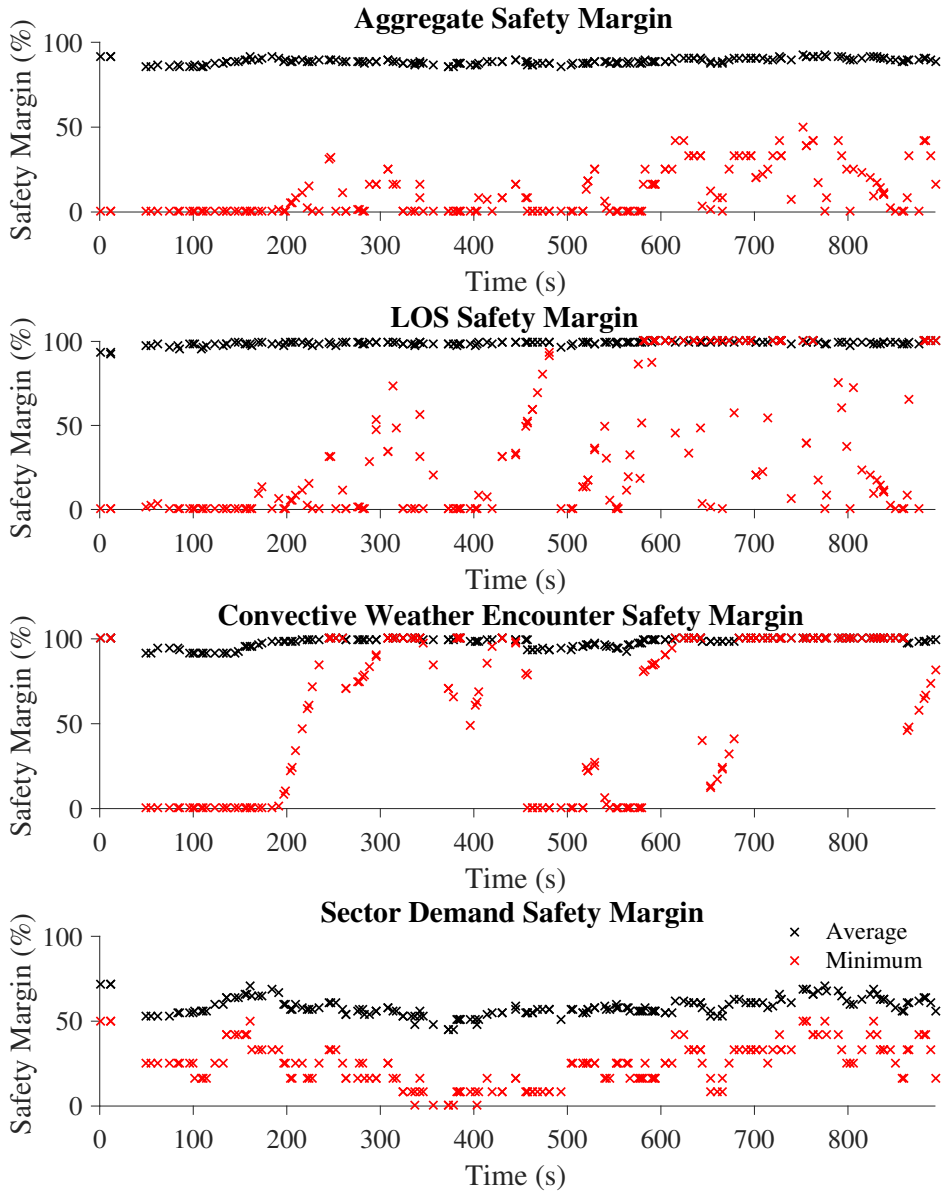

Figure 7. Estimated safety margins, as a function of time.

Acknowledgments

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

[1]Flight Safety Foundation, "Decision-Making (Operator's Guide to Human Factors in Aviation)," 2010, https://www.skybrary.aero/index.php/Decision-Making_(OGHFA_BN).

[2]Albright, J., "Big Sky Redefined," *Business and Commercial Aviation*, March 2017, pp. 24–31.

[3]Croft, J., "Who is in charge of safety at amsterdam schiphol?" *Aviation Daily Apr 24, 2017*, April 2017, pp. 4–5.

[4]General Aviation Joint Steering Committee, "Flight Risk Assessment Tools," 2008, https://www.faa.gov/news/safety_briefing/2015/media/SE_Topic_15-08.pdf.

[5]AOPA, "Flight Risk Evaluator," 2010, https://www.aopa.org/go-fly/flight-tools.

[6]Federal Aviation Administration (FAA), "Safety Management System (SMS)," 2016, https://www.faa.gov/about/initiatives/sms/.

[7]Smith, B., "Safety Net Inventory," Tech. Rep. To appear as NASA/TM, NASA Ames Research Center, Moffett Field, CA, USA, June 2017.

[8]Center, V. N. T., "Traffic Flow Management System (TFMS)," Tech. Rep. VNTSC-TFM-RM-010, U.S. Department of Transportation, Cambridge, Massachusetts, 2011.

[9]Spirkovska, L., Roychoudhury, I., Daigle, M., and Goebel, K., "Real Time Safety Monitoring: Concept for Supporting Safe Flight Operations," *17th AIAA Aviation Technology, Integration, and Operations Conference*, June 2017.

[10]Roychoudhury, 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, December 2015.

[11]Roychoudhury, I., Spirkovska, L., Daigle, M., Balaban, E., Sankararaman, S., Kulkarni, C., Poll, S., and Goebel, K., "Predicting Real-Time Safety of the National Airspace System," *AIAA Infotech@Aerospace Conference*, January 2016.

[12]Roychoudhury, I., Daigle, M., Goebel, K., Spirkovska, L., Sankararaman, S., Ossenfort, J., Kulkarni, C., McDermott, W., and Poll, S., "Initial Demonstration of the Real-time Safety Monitoring Framework for the National Airspace System Using Flight Data," *16th AIAA Aviation Technology, Integration, and Operations Conference*, June 2016.

[13]Daigle, M., Sankararaman, S., and Roychoudhury, I., "System-level Prognostics for the National Airspace," *Annual Conference of the Prognostics and Health Management Society 2016*, October 2016, pp. 397–405.

[14]Bilmoria, K. D., Banavar, S., Chatterji, G. B., Sheth, K. S., and Grabbe, S., "FACET: Future ATM concepts evaluation tool," *3rd USA/EuropeATM R&D Seminar*, June 2000.

[15]Julier, S. J. and Uhlmann, J. K., "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, March 2004, pp. 401–422.

[16]Daigle, M., Saha, B., and Goebel, K., "A Comparison of Filter-based Approaches for Model-based Prognostics," *2012 IEEE Aerospace Conference*, March 2012.

[17]Sankararaman, 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.

[18]Sankararaman, S. and Daigle, M., "Uncertainty Quantification in Trajectory Prediction for Aircraft Operations," *AIAA Guidance, Navigation, and Control Conference*, January 2017.