A Computational Platform for Analyzing the Safety of the National Airspace System

Padmanabhan K. Menon[1], Bong-Jun Yang[2], Parikshit Dutta[3], Sang Gyun Park[4], Oliver Chen[5] and Victor H. L. Cheng[6]

[1,2,3,4,5,6]*Optimal Synthesis Inc., Los Altos, CA, 94022, USA*

menon@optisyn.com, jun.yang@optisyn.com, pdutta@optisyn.com, sang.park@optisyn.com, o.chen@optisyn.com, vcheng@optisyn.com

ABSTRACT

Safety analysis of complex systems can be carried out using computational models of the subsystems and their interactions. A modeling framework for analyzing the safety of the national airspace operations is described. The framework enables investigators to collaboratively formulate models of the subsystems of the National Airspace System and insert off-nominal operating conditions to assess their impact on the overall system safety. The software is designed to allow popular computational frameworks such as Python and Java to interact with it in a seamless manner. Interfaces to one of the popular commercial computational package are also provided. The software incorporates detailed aspects of the national airspace system infrastructure such as airport taxi ways and runways, approach-departure procedures, jet routes, air traffic control Sector and Center boundaries. It also includes simplified human performance models of controllers, pilots and ground vehicle operators. Performance data for over 400 different aircraft types, derived from a well-known database have also been incorporated. Airspace flight operations data is obtained from the Federal Aviation Administration data feed, and the weather data is derived from the National Oceanic and Atmospheric Administration web site. Two example applications of the software are given.

1. INTRODUCTION

Safety of operations is of paramount importance in the global air transportation system. In the United States (US), aviation is considered to be the *safest* mode of travel. Highly detailed analysis of aviation accidents by the National Transportation safety Board (NTSB) (NTSB, 2018), and the incident reporting system maintained by the Federal Aviation Administration (FAA) (FAA, FAA Accident and Inciden Data, 2018) and National Aeronautics and Space Administration (NASA) (NASA, 2018) have facilitated near-continuous improvement in the US aviation safety record since the middle of the 20th century. This enviable record has been achieved by continuously improving the certification methodologies, procedures and decision-support tools for airspace operations, surveillance and communication technologies, and aircraft maintenance and repair operations. In the decade spanning 2000 - 2010, the number of deaths per passenger-mile on commercial airlines in the US was around 0.2/10 billion passenger-miles (BTS, 2018), a remarkably small number.

Rapid growth of aviation in the US and worldwide since 2010 has further sharpened the focus on safety issues, since, even if the rate of incidents/accidents per passenger mile remained the same, the number of these events will continue to increase to socially unacceptable levels as the number of flight operations increase. Motivated by this factor, and the fact that a considerable amount of data on aviation operations is becoming available, NASA and the FAA have initiated research initiatives to consider methods for enhancing the system safety. System Prognostics (Roychoudhury, et al., 2015) has been identified as one of the technologies that has the potential for substantially improving the safety of the National Airspace System (NAS) by identifying safety-compromising situations in the system before they occur and adopt appropriate mitigation strategies. This program forms the foundation for the work reported in this paper.

This paper advances a computational framework in which factors impacting the safety of national airspace operations can be modeled and analyzed to assess emerging safety issues. This framework will be termed as the National Airspace Traffic-Prediction System (NATS) throughout this paper. NATS is implemented as a server-client software package that incorporates realistic models of three major subsystems: Equipment, Entities and Environment. The Equipment category includes aircraft, flight-deck automation equipment, ground vehicles, and surveillance and communication systems. The Entities category includes error models of all human operators involved in NAS operations such as pilots, air traffic controllers and ground vehicle operators. Thirdly, the Environment subsystem consists of airports with ramp, taxiways and runways, en-route and terminal area flight operations procedures, terrain, and weather. Any other subsystem models to be considered in the analysis can be modeled by the analyst and integrated with NATS under one of these three categories.

The three major components that make up the National Airspace operations are notionally illustrated in Figure 1. Since there can be over 6000 aircraft operations in the US NAS at any time instant, the fidelity of the models employed in system-level studies must necessarily be kept low enough to ensure that the system dynamics can be evaluated with available computational resources.

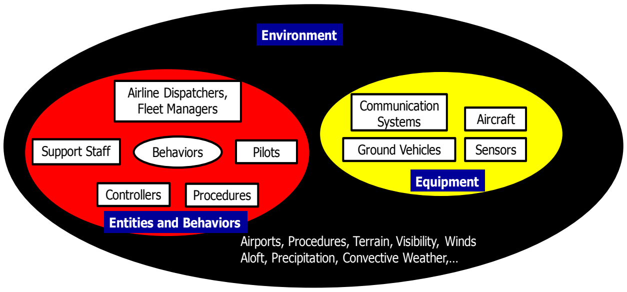

Figure 1. Subsystems modeled in the NATS computational platform.

Two distinct roles are envisaged for the NATS software. Firstly, it can be used by individuals or groups of investigators to carry out air traffic analysis in the NAS, and to develop prognostics algorithms. Secondly, the NATS software can be used as the traffic prediction and safety metric computing component of the real-time NAS-wide prognostics methodology being developed under the research supported by NASA under Contract No. NNX17AJ86A.

The NATS software is currently under development (stage-Alpha), and this paper should be considered as a report on the work-in-progress.

The following sections will outline some of the models developed currently, together with an overview of the software architecture. Two applications examples will then be described.

2. EQUIPMENT MODELS

As indicated in Section 1, the main equipment employed in the NAS is aircraft together with flight-deck automation system, ground service vehicles, communication systems, and surveillance systems. Although it may be desirable to incorporate high-fidelity models describing their nominal and off-nominal behaviors, this may not be feasible due to large number of subsystems that needs to be included, leading to unmanageable computational effort. Consequently, simpler models capturing the essential features of their operation are included. The following subsections provide additional details.

2.1. Aircraft and Ground Vehicle Models

High-fidelity aircraft models are generally described a set of 12 first-order nonlinear differential equations. However, six of these are not relevant to the system safety analysis, because they deal with the attitude motions of the aircraft in flight. Eliminating these equations from the aircraft dynamics is equivalent to assuming that the aircraft are always in "moment equilibrium". Under this assumption the aircraft dynamics can be described by six first-order nonlinear ordinary differential equations, commonly termed as the point-mass model(Napolitano, 2012). These equations generally require detailed knowledge about the aircraft engines, aerodynamics and mass.

Since it is nearly impossible to get accurate data for every aircraft that are operating in the NAS, a further simplification is made in the analysis. The simplification consists of employing the three differential equations describing the position kinematics of aircraft, together with rate of climb, rate of descent, and airspeed or Mach number bounds from well-known databases such as BADA (Eurocontrol, 2018). This database was assembled by Eurocontrol using actually observed trajectories in the European airspace.

With these simplifications, the equations of motion for an aircraft are given by:

$$\dot{h} = f(h, A), \qquad \gamma = \sin^{-1}\left(\dot{h}/V\right) \; V_{min} \leq V(h, A) \leq V_{max},$$

$$\dot{\lambda} = \frac{1}{(R_e + h)}\{V cos\gamma cos\chi + W_N\}, \tag{1}$$

$$\dot{\tau} = \frac{1}{(R_e + h)cos\lambda}\{V cos\gamma sin\chi + W_E\}$$

The definitions of the variables in these equations are indicated in Figure 2.

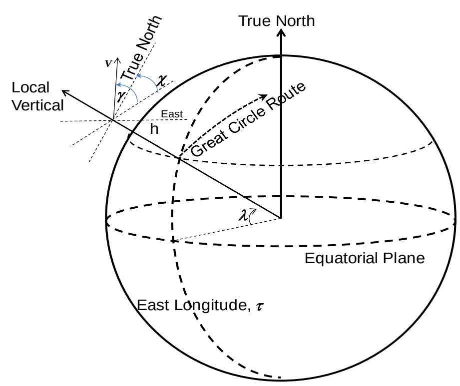

Figure 2. Definition of the variables in the aircraft kinematic equations of motion.

The control variables in this model are the airspeed V, flight path angle γ or the altitude rate $f(h, A)$, and the course angle χ. Note that the limits on the climb and descent rates and the airspeed are specified in BADA for over 400 Aircraft types. In the present NATS framework, these variables are chosen by the human pilot and/or flight deck automation to follow flight plans approved by the controller. The North component of the wind W_N and the East component of the wind W_E are obtained from the National Oceanic and Atmospheric Administration (NOAA) weather data.

In order to enable the investigation of potential accidents and incidents that can occur in each flight phase, the aircraft operations in the NAS have been separated into 11 major flight regimes. These are: stationary at the gate, pushback, taxi to runway, takeoff, climbout, climb-to-cruise, cruise, initial descent, approach, landing, taxi to gate. The software is designed such that the salient motion characteristics of aircraft in historic accidents and incidents in each of these flight regimes and their impact on NAS operations can be formulated and analyzed.

The motion of the aircraft on the ground and the motion of ground vehicles can be simulated by eliminating the altitude dynamics, and assuming that the flight path angle γ is zero. Moreover, the wind has negligible effect on aircraft motion when it is on the ground. With these simplifications, the equations of motion for the aircraft moving on the ground, and the equations of motion for the ground vehicle, are:

$$\dot{\lambda} = \frac{V cos\chi}{(R_e + h)}$$
$$\dot{\tau} = \frac{V sin\chi}{(R_e + h)cos\lambda} \tag{2}$$

The control variables employed by the pilot or the ground vehicle are the speed V and the course angle χ used to move along the ramp, taxiways, and up to the runway.

The aircraft and ground vehicle dynamic equations are integrated using the first-order Euler integration method.

2.2. Navigation and Flight-Deck Automation Systems

The main navigation aids currently used by commercial aircraft in the NAS are the Global Positioning System (GPS), the Inertial Navigation System and the Instrument landing System (or Microwave Landing System). Navigational errors can cause the aircraft to deviate from specified flight procedures, potentially leading to unsafe operating conditions. In order to assist in the investigation of the effect of these errors on flight safety, the NATS software allows the user to introduce both deterministic and random errors into the aircraft position and velocity components and in the sequence of operations.

Modern commercial aircraft are operated with the aid of flight deck automation systems. These include the autopilot and the autothrottle settings accessible through the Mode Control Panel, with the Flight Management System (FMS) providing trajectory tracking, fuel management and other higher-level automation functions. Pilots access the FMS functionality through the control display units.

Simplified representations of these automation systems are provided in the NATS software. For instance, using the aircraft flight plan, the FMS subsystem will generate the course angle required to track the series of latitude-longitude waypoints using the formula (Bilimoria, Sridhar, Grabbe, Chatterji, & Sheth, 2001):

$$\chi = tan^{-1}\left\{\frac{sin(\tau_{i+1} - \tau)\, cos\lambda_{i+1}}{sin\lambda_{i+1} cos\lambda - cos\lambda_{i+1}\, sin\lambda\, cos(\tau_{i+1} - \tau)}\right\} \tag{3}$$

In this expression, (λ, τ) is the current latitude-longitude location of the vehicle, and $(\lambda_{i+1}, \tau_{i+1})$ is the latitude-longitude location of the next waypoint in the flight plan. The altitude changes required by the flight plan are implemented using the BADA data for the specific aircraft type.

The autothrottle function is simulated in NATS by selecting the airspeed from the BADA corresponding to a specific aircraft type and flight regime, and using these to integrate the equations of motion.

Just as in the case of the navigation system, deterministic and random error components or operational errors can be introduced into the automation system outputs to investigate their effects on the system safety.

Next generation aircraft are likely to have additional automation available on the flight deck such as airborne self-separation, and tools for trajectory-based operations. NATS software is being designed to enable the investigation of the potential error sources in these systems, and their impact on the NAS safety.

2.3. Communication and Surveillance Systems

Most of the communications between the controller and the pilot in the present air traffic control system are achieved over very high frequency/ ultra-high frequency (VHF/UHF) radio. The contents of the communications typically involve flight plan modifications, changes in cruise altitudes, speed and heading advisories, and potential weather deviations. NATS provides functions for introducing communication errors and to assess their impact on the NAS safety. Moreover, the effect of the terrain on communications between aircraft and between aircraft and the ground can also be assessed in NATS, as will be discussed in Subsection 3.5.

Dependent surveillance of the most of the aircraft in NAS is currently achieved through a network of ground-based tracking radars interacting with Mode C transponders onboard aircraft(FAA, FAA Advisory Circular, 2005). FAA has mandated that by January 1, 2020, every aircraft operating in the NAS that are currently required to carry Mode C transponders, must be equipped with Automatic Dependent Surveillance-Broadcast (ADS-B) Out capability. The ADS-B system derives position estimates with the aid of GPS satellites, augments the data with aircraft-derived speed, heading and vertical speed information which is then broadcasted. The aircraft state estimates are highly accurate and are available at much higher sample rates than the radar.

However, the system is currently susceptible to jamming, which can significantly degrade their performance. Moreover, the position estimates can contain significant error under certain GPS satellite-constellation configuration relative to aircraft.

NATS software provides functions for modeling these and other known susceptibilities to assess their impact on the system safety.

3. ENVIRONMENT

Environmental factors have contributed to several well-known accidents in the NAS. Functions are provided in the NATS for modeling the nominal and off-nominal behavior of the system under the influence of environmental factors such as adverse terrain and weather. Some aspects of these environmental factors are discussed in the following subsections.

3.1. Airports

Airports are central to flight operations in the NAS. NATS software includes highly detailed gate-taxiway-runway layouts for 55 major airports in the US. The models are given in the node-link form, facilitating the formulation of various accidents and incidents reported by the NTSB. One of the examples, given in the later portion of this paper will illustrate the use of these functions for setting up a taxi route from a specified gate to a runway. Variations on this example can be used to set up some of the accidents reported in the NTSB aircraft accident reports.

In order to simplify the design of taxi routes, NATS includes a function that can be used to determine shortest-distance (Abiy, Pang, & Khim, 2018) taxi routes from gates to the runways and vice versa. The taxi routes constructed in this manner can be appended to the flight plans from the FAA data feed to create gate-to-gate aircraft trajectories from origin to destination airports.

3.2. Arrival and Departure Procedures

Instrument procedures for arrival and departure to and from airports are prescribed by the FAA and form an important part of the NAS environment. These procedures are generally given by charts, an example of which is given in Figure 3 for approach to Runway 28L at the San Francisco International Airport (KSFO).

The FAA Coded Instruments Flight Procedure (CIFP) database (FAA, Coded Instruments Flight Procedures, 2018) provides access to the complete list of published procedures at every major airport in the NAS. NATS software provides access to these procedures through several interface functions for use by the investigators.

The main reason to include these procedures in the NATS software is that the flight plans in the data feed from the FAA may not include terminal area and arrival departure procedures. However, these must be available in order to complete the flight plan required predict aircraft trajectories for use in safety analysis.

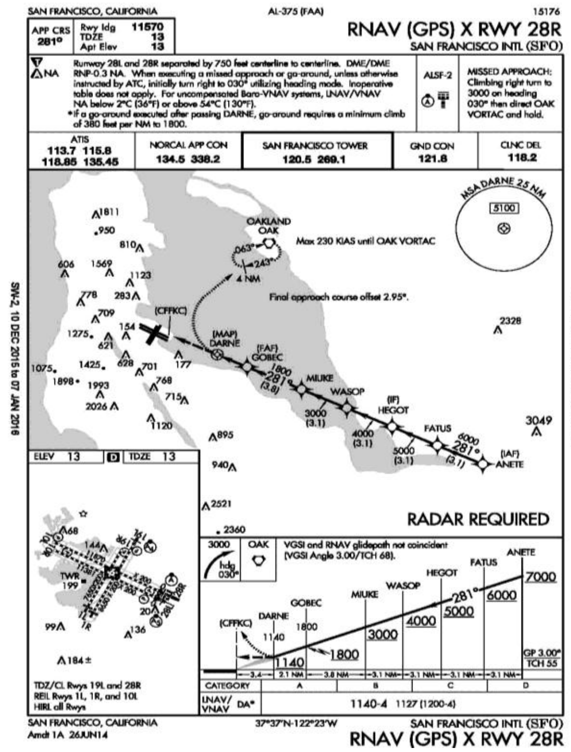

Figure 3. Approach procedure into Runway 28R at the San Francisco International Airport (Source: FAA).

3.3. En-route Airspace

As in the case of arrival and departure procedures, NATS provides functions for accessing the complete list of jet routes and waypoints in the NAS, available in the CIFP database. These waypoints can be used to verify flight plans in the FAA traffic data feed, and to design valid flight plans for introducing test aircraft in the NAS simulations. Moreover, this data will be used by the controller and pilot models described in following sections to create weather-avoidance routes.

3.4. Weather

Weather is one of main sources of uncertainties in the NAS. Typical weather models of interest in safety assessments are the en-route winds, convective weather, winds in the terminal area, crosswinds and tailwinds at the runways.

NATS provides functions to download weather data from the NOAA web site (NOAA, 2018), and make it available for use in trajectory simulations. The weather data can also be used to simulate controller functions such as aircraft rerouting in severe weather. In addition to the strategic rerouting actions, the controllers may also issue tactical flow control advisories to the aircraft in the terminal area, primarily by vectoring the aircraft around weather, of by holding them in standard pattern near the Terminal Radar Approach Control (TRACON) metering fixes.

As an example, Figure 4 and Figure 5 show routes with and without rerouting to avoid an en route severe weather region, shown as a shaded polygon in these figures. These routes are between Las Vegas (KLAS) and Minneapolis-St. Paul (KMSP), and between Phoenix (KPHX) and New York John F. Kennedy (KJFK) airports.

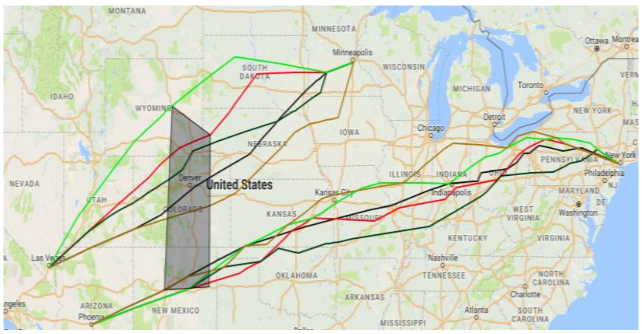

Figure 4. Flight plans before weather-avoidance rerouting.

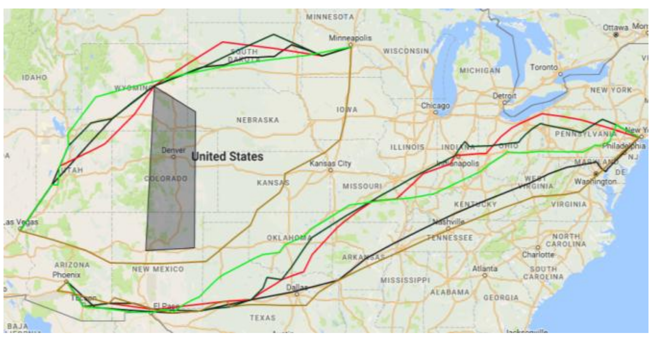

Figure 5. Modified flight plans avoiding convective weather.

3.5. Terrain

The terrain over the regions of the NAS places severe operational constraints on flight operations. Firstly, the terrain can pose direct hazards to aviation, by requiring higher navigation precision to achieve safe arrivals and departures at certain airports. Secondly, the terrain can limit the line-of-sight between various regions in the NAS, making it difficult to carry out VHF/UHF communications between controllers and pilots. As an example, Figure 6 illustrates the effect of the terrain on line-of-sight communications in the vicinity of the Salt Lake City international airport (KSLC). NATS provides functions to obtain the terrain height above sea level, given any latitude-longitude pair within the contiguous US. These functions can be used to model the impact of the terrain on navigation accuracy, and to model accidents such as controlled flight into terrain.

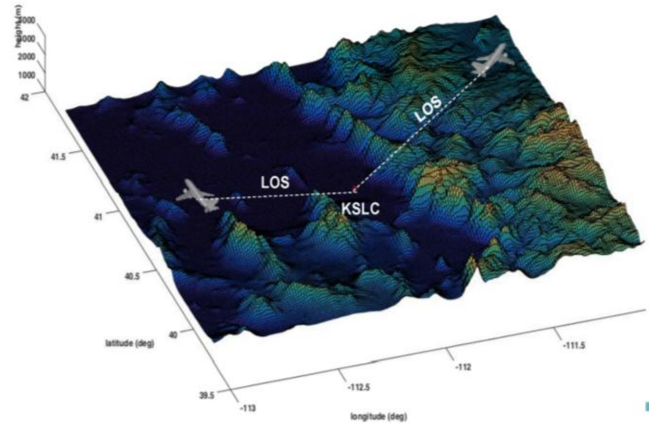

Figure 6. The effect of the terrain on line-of-sight communications in the vicinity of Salt Lake City international airport (KSLC).

4. ENTITY MODELS

Human errors have contributed to several accidents and incidents in the air transportation system. All the human operators in the NAS such as pilots, ground vehicle operators and pilots are grouped under the "Entity" umbrella in the NATS software.

NATS software incorporates one of the well-known human operator models, the Human Error Template (HET) (Stanton, et al., 2010) that classifies human errors into 11 groups, and one user-defined group. For a given task, the HET model classifies the potential errors into the following categories:

- Failed to execute
- Task execution incomplete
- Task executed in the wrong direction
- Wrong task executed
- Task repeated
- Task executed on the wrong interface element
- Task executed too early
- Task executed too late
- Task executed beyond requirements
- Task executed less than the requirements
- Misread information
- Other (user-defined).

Air traffic control advisories, combined with the flight plans form the basis for the human pilots to interact with the aircraft through flight deck automation systems such as the autopilot, autothrottle and the FMS. This process is susceptible to the introduction of several potential errors or faults into the system. For instance, pilots can compromise system safety by the incorrect execution of controller advisories, incorrect FMS data entry, incorrect settings on the Mode Control Panel, and the incorrect selection of flight modes.

Recently, the HET model was employed by two of the researchers in the present paper (Park & Yang, 2016) to investigate the effects of mode confusion on flight safety. One such a mode-confusion example is listed in Table 1.

Table 1. Example of a "Task Executed Too Late"

Event Time/ Pilot Action	Resulting Aircraft behavior
At 175 sec. (around 8000 ft.), the pilot toggle the vertical speed button	The vertical speed wheel lets the pilot set the vertical speed.
At 195 sec., the pilot set the target vertical speed at -1000 feet per minute (fpm).	The target vertical speed is set to -1000 fpm.
At 250 sec. (around 6,800 ft.), the pilot sets the target altitude to 7000ft.	In this particular aircraft, if the target altitude is set before descent to the target altitude, the aircraft would capture the target altitude (nominal). However, in the present example, the target altitude was set after the target altitude had passed, and the aircraft continued to maintain the vertical speed mode. This can lead to a safety-compromising situation if the pilot is not aware of this particular mode of operation near the ground.

The test was set up in the NASA Multi-Aircraft Control System (MACS) simulator (Prevot, et al., 2006) incorporating the Mode Control Panel in a modern airliner. Altitude histories corresponding to this example is given in Figure 7. Note that the mode confusion error resulted in the aircraft continuing to descend below the desired level-off altitude, potentially leading to unsafe operation. Additional details on this example are available in the Reference (Park & Yang, 2016).

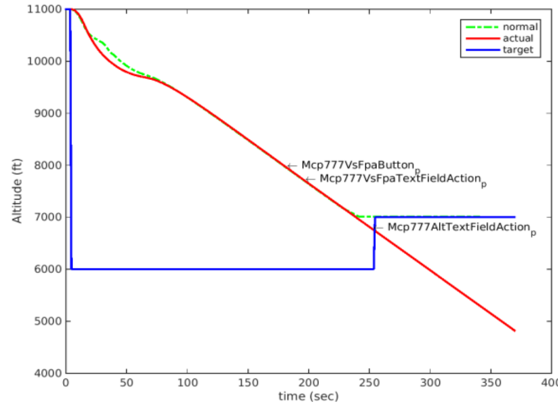

Figure 7. Altitude profiles for the nominal and too-late mode selection by the pilot.

In addition to modeling the behavior human pilots, the HET framework can also be used to model controller errors. Following the present mode of air traffic control operations in the NAS, NATS assumes that the controllers are responsible for generating advisories for separating the aircraft in flight and on the ground, sequencing and scheduling arrivals and departures in the terminal area, and convective weather avoidance. They accomplish these functions by monitoring the relative positions of aircraft and velocity vectors of aircraft in their displays, assisted by an array of decision support tools to issue the advisories.

Just as in the case of flight deck automation, NATS can be used to assess the safety impact of errors and faults introduced by controller decision support tools by assuming ideal behavior of the controllers.

NATS currently incorporates algorithms for accomplishing the controller tasks using the aircraft state data. HET model can be used to modulate the behavior of these algorithms to simulate controller decision errors. The safety implications of such errors can then be analyzed by propagating the aircraft trajectories forward in time using the NATS software.

5. SAFETY METRICS

NATS software provides functions for evaluating various safety metrics based on the NAS aircraft states. Several NAS safety metrics have been identified in the recent literature (Roychoudhury, et al, 2015). The following are some of the desirable characteristics of the safety metrics.

1. They must be sufficiently intuitive for widespread adoption in routine operations
2. They must capture the essential threats to the NAS safety
3. They must be computable using traffic data, expected system characteristics, and conditional

probability density functions (PDF) derived from historic incidents/accidents databases

4. They must be adaptive to the future evolution of NAS.

The computation of these (Roychoudhury, et al, 2015) and other safety metrics require not only the current aircraft states and their temporal evolution, but also the conditional PDFs of adverse events occurring, given the evolution of the aircraft state vectors in the NAS.

As an example, consider the predicted proximity of the aircraft to certain icing regions in the NAS forecast by NOAA. If the conditional PDF of icing on the aircraft, given its proximity to the icing conditions, were available, the safety hazard posed by them on NAS can be assessed. In certain cases, these conditional PDF can be estimated from NTSB, FAA and NASA accident/incident databases. However, due to the extreme rarity of these events, in most cases, these conditional PDFs will need to be estimated using Monte Carlo simulations.

As another illustrative example, (Andrews, Welch, & Erzberger, 2001) discusses the use of fault trees for the comprehensive estimation of conditional PDFs for next-generation aircraft separation concepts.

In addition to the list of safety metrics outlined in Reference (Roychoudhury, et al, 2015), NATS software incorporates some of the FAA metrics such as the Operational Errors (OE), Operational Deviation (OD), Pilot Deviation (PD), Runway Incursion (RI), Near Midair Collisions (NMACs), Vehicle and Pedestrian Deviations (VPDs). Additional metrics that may be included are FAA/Eurocontrol safety metric-Aerospace Performance Factor (APF), and the NASA trajectory-based complexity (TBX) metric.

6. NATS SOFTWARE

The computational modules incorporated in the current version of the NATS software (Version Alpha 1.0.0) are shown in Figure 8.

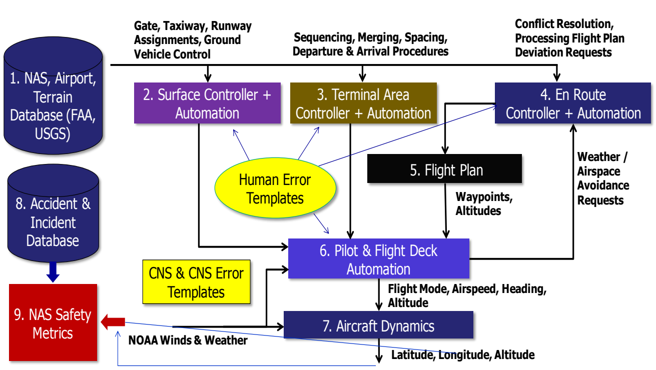

Figure 8. Computational modules implemented in the NATS software.

NATS software provides functions that can be used to select the integration step size, start and stop time instants, start, pause, resume, and stop controls. It also allows the user to specify the simulation data output format and to carry out collaborative simulations.

The NATS software is based on a client-server framework in which the all the computational functionality is implemented on a server hosted in the cloud, with the client accessing this functionality through a set of Java™(Oracle, 2018), Python™(Python Software Foundation, 2018) or MATLAB®(The MathWorks Inc., 2018) interfaces. A notional block diagram of the NATS software is given in Figure 9.

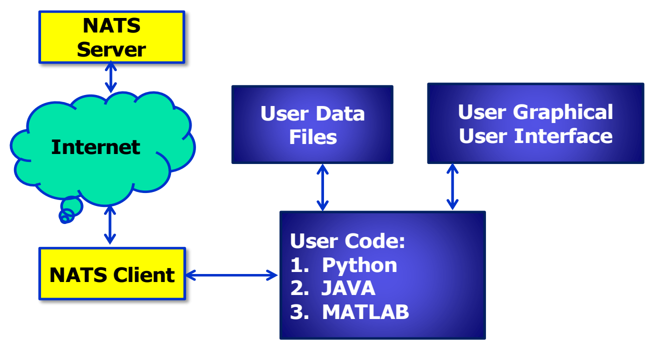

Figure 9. NATS software architecture.

Compute intensive portions of the NATS software are coded in C++ for speed and flexibility, and the client-server interface is implemented in Java, Python and MATLB. In addition, the NATS server software is being coded in a way amenable to the use of high-performance computing platforms, such as Graphical Processing Units, in order to handle the need for Monte Carlo simulations of the entire NAS.

7. NATS SOFTWARE USAGE EXAMPLES

Several application examples have been developed to serve as tutorials to the NATS software users. The following sections will present two of these to illustrate some of the NATS software capabilities.

7.1. Taxi Route Planning

The ground operations of a departing aircraft include pushback from the gate to the ramp, taxi to the runway, and upon clearance by the air traffic controller, takeoff from the specified runway. Reverse sequence of ground operations is performed by arriving aircraft.

En route flight plans from the FAA traffic data feed currently do not include taxi routes. However, realistic prediction of the traffic for prognostics requires aircraft taxi plans, as well as the runways from which the aircraft will be operating. NATS provides a set of functions that can be used to create interactive code for taxi way design and for the selection of approach-departure procedures. This example demonstrates a Python function that use of the taxi route design interface.

7.1.1. Airport Surface Layout

The NATS software incorporates the application programming interface, `AirportInterface`, for handling functions related to the airports, taxiways, and runways. Airport related functions are available for 55 major airports in the US. For example, Figure 10 shows the KSFO layout. The layout is defined by a set of nodes (denoted by teal dots) and links (denoted by teal lines). The runways are displayed in thicker grey lines.

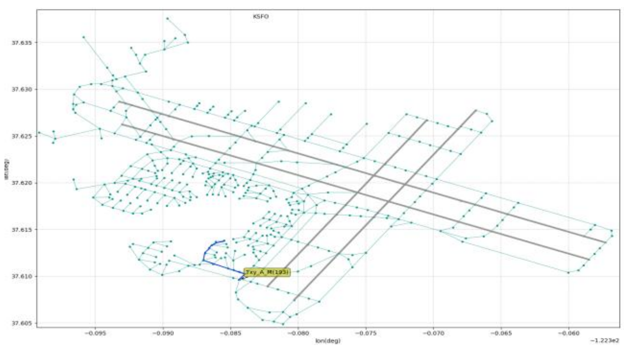

Figure 10. Airport layout and a departure taxi plan design from Gate G-01-001 to a runway 01L at the San Francisco International Airport (KSFO)

If desired, the airport layout can be uploaded to Google Earth or Google® MyMaps™ (Google, 2018) as shown in Figure 11. In addition to a detailed display of airport layouts with terminals and surrounding land marks, Google MyMaps interface allows the user to add or remove nodes and links.

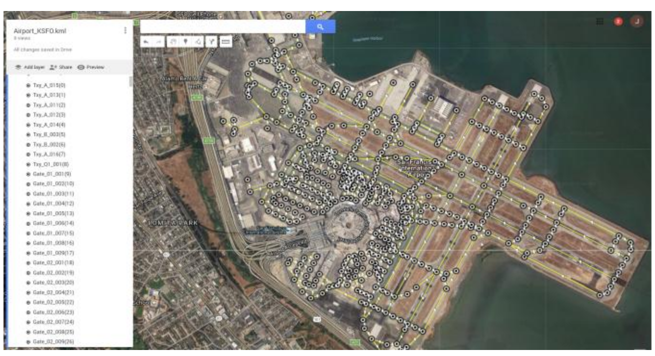

Figure 11. San Francisco airport layout on Google MyMaps.

7.1.2. Route Design

As motivated earlier in this paper, the primary objective in designing taxi routes is to enable the simulation of aircraft motion on the airport surface. The taxi plan starts from a gate and ends at a runway threshold for a departing flight and starts at a runway exit and ends at a gate for an arriving flight. NATS software provides several functions that can be used to develop user-defined codes for developing taxi routes. The results from a Python code based on these functions are discussed in this section.

Figure 10 and Figure 12 show some of the interactive features of a user-designed Python code used in the taxi route design. The user begins the route design process by clicking on the gate node from which the taxi plan begins in the interactive figure and continues to click through the nodes that make up the taxi route. A double click at the runway entrance node completes the taxi plan.

```
jun NATS_Client_20180402 $ python dep_taxiplan_example.py
==============================================================
  National Airspace Trajectory-Prediction System(NATS) Client

  Optimal Synthesis Inc.
==============================================================

Connected to NATS Server (localhost:2017)
acs:(u'ULI13-2553603',) are loaded
 AC:ULI13-2553603 departs from KSFO
total 42 CENTERs are read
Gate_02_001(18) will be added to the route.
Ramp_02_001(118) will be added to the route.
Ramp_02_002(117) will be added to the route.
Ramp_02_003(119) will be added to the route.
Ramp_02_004(120) will be added to the route.
Ramp_02_005(124) will be added to the route.
Txy_H_003(123) will be added to the route.
Txy_A_004(230) will be added to the route.
Txy_A_M(193) will be added to the route.
```

Figure 12. Screen output when the user clicks at the nodes

The taxi plan design for an arrival flight follows the exact same process, carried out in a reverse sequence. It starts with a mouse click at a runway exit node and ends with a double click at a selected gate. For example, Figure 13 shows a user-designed arrival taxi plan for the JFK International Airport (KJFK), shown as solid blue line.

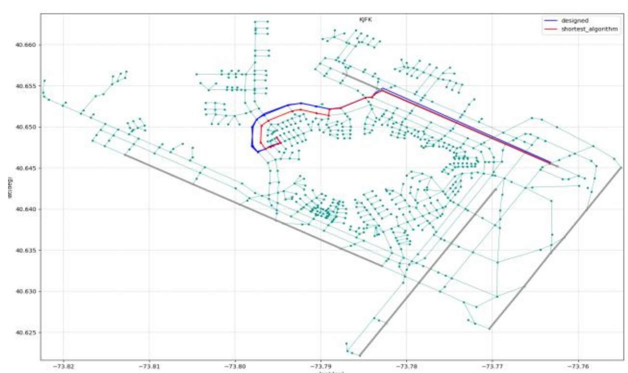

Figure 13. User-designed arrival taxi plan (in blue) in comparison with that generated using the NATS Shortest Path Algorithm (in red) from Gate-08-31A to Runway 31R.

As an alternative to such interactive taxi route design process, NATS software also provides the Shortest Path A-Star Algorithm (Abiy, Pang, & Khim, 2018) for taxi route design. An example of the shortest arrival taxi route computed using the NATS function is displayed in red in Figure 13.

7.1.3. NATS Airport Interface Functions

Table **2** shows a list of NATS functions for the retrieval of airport layouts and the design of taxi plans. Once the taxi plan is integrated with the aircraft flight plan, its trajectory can be generated by the NATS software using the equations of motion.

Table 2. NATS Functions for the Airport Interface

Function Objective	Relevant NATS Functions
Retrieves a departure /arrival airport for a flight	`getArrivalAirport`, `getDepartureAirport`
Retrieves airport layouts	`getLayout_node_map` `getLayout_node_data` `getLayout_links`
Design shortest-path taxi plans and retrieve them	`getSurface_taxi_plan` `generate_surface_taxi_plan` `setUser_defined_surface_taxi_plan` `get_taxi_route_from_A_To_B`
Set and get taxi speeds	`getTaxi_tas_knots` `setTaxi_tas_knots`

7.2. Monte-Carlo Simulation of the En Route Airspace

Off-nominal performance of aircraft, navigation, communication systems, and inaccurate execution of flight procedures by human pilots are some of the factors that can compromise NAS safety. As motivated in Section 5, the conditional probability densities of safety compromising events or situations, together with the instantaneous prediction of aircraft states can be used to estimate the safety of the overall air transportation system.

The present example is motivated by the hypothesis that since the safety of the en route airspace depends on the controller's ability to resolve conflicts, and since the number of controller errors is correlated with the number of conflicts they need to resolve, the safety of en route airspace can be assessed by determining conditional PDF of the number of conflicts under various perturbations. A limited version of this conditional PDF estimation in a Monte Carlo simulation is presented in this section.

NATS software provides a set of "get" and "set" functions for flight plans, states and inputs for every aircraft in the NAS simulation. These functions can be used to formulate comprehensive Monte Carlo simulations to estimate conditional PDFs. The general approach is to generate random sample values of the aircraft flight plans, states and inputs from assumed PDFs, "set" these values for the aircraft under study, and propagate the simulation for a desired number of time-steps. The simulation can then be paused, desired values extracted using "get" functions, followed by the setting of new values, and the simulation can be resumed to the next time instant or event.

The Monte Carlo simulation example given in this section considers a hypothetical 80 aircraft traffic scenario over the contiguous US for the duration of four hours. The objective is to demonstrate the assessment of the conditional PDF of conflicts occurring in the en route airspace when the one or more aircraft operates in an off-nominal manner.

The present example considers trajectory perturbations for three aircraft in the 80 aircraft traffic, the first one taking off from San Jose to Louisville (KSJC→KSDF), the second one departing from Denver to Louisville (KDEN→KSDF), and the third flight from Newark to Los Angeles (KEWR→KLAX).

In the present example, these aircraft are subject to uniformly distributed perturbations in their departure times in the range [0, 240] seconds, cruise speed variations distributed as zero-mean Gaussian PDFs with 10% standard deviations and variations in latitude components at each of their flight plan waypoints distributed as zero-mean Gaussian PDFs with 1% standard deviation. These constitute a total of 60 random variables being perturbed in the present Monte Carlo simulation.

The perturbations in departure times can occur due to adverse traffic conditions, weather and equipment malfunctions at departure airports; while the cruise speed variations normally occur due to en route winds. The flight plan waypoint perturbations can occur due to tactical weather rerouting by the controllers, navigation errors in the aircraft avionics and onboard automation/pilot-induced errors, uncorrected by the en route air traffic controllers.

In the present hypothetical example, under the nominal case, the first aircraft encountered one conflict with other aircraft en route, while the other two encountered two and one conflicts, respectively, with other aircraft en route. The traffic simulation had a total of 7 conflicts, the three between other aircraft in the simulation. The present definition of the conflicts is from the FAA. According to the FAA, an en route conflict is declared if any two aircraft approach each other closer than 5 nautical miles at the same altitude, or if they approach less than 1000 feet altitude at the same horizontal position.

Results for 1000 Monte Carlo iterations are presented in Figure 14. Since 60 variables are being perturbed in the Monte Carlo simulation, with a simple sampling strategy, statistically significant results may require 1000 or more iterations per variable, or a total of 60,000 iterations in the present example. Thus, the histogram given in this figure should be considered an intermediate result.

However, the figure shows the type of results that could be generated using Monte Carlo simulations. For instance, this intermediate result shows that the present nominal case is more of an anomaly, and en route airspace appears to be safer under the perturbations introduced in this example.

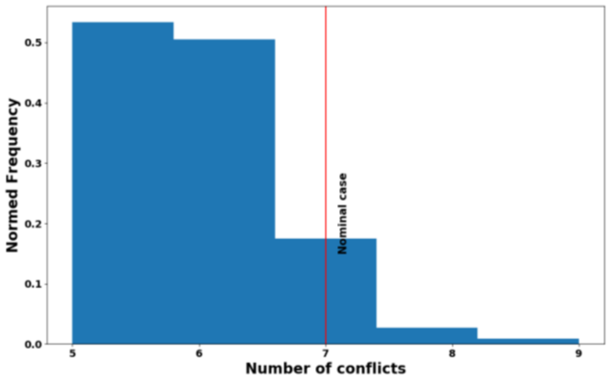

Figure 14. Intermediate Conditional Histogram of aircraft conflicts en route, predicted by the hypothetical 80-aircraft NAS traffic Monte Carlo simulation under state, flight plan and departure time perturbations for three aircraft.

8. CONCLUSIONS

This paper described a software package for analyzing the safety of the National Airspace System. It allows the user to model every phase of aircraft operations such as taxi, takeoff, climb-to-cruise, descent-to-land and taxi to gate. It also allows the introduction of human, equipment and environmental errors, faults and perturbations for the assessment of conditional probability density functions corresponding to various adverse events that can occur in the National Airspace System. Two examples on the use of the software were described.

In addition to being useful for off-line analysis, a real-time version of the software can be used for computing the conditional probability densities of various safety-compromising traffic situations using real-time traffic data feed. This capability will be useful for implementing real-time prognostics concepts.

ACKNOWLEDGEMENT

This work was carried out under Arizona State University Subcontract No. ASU-18-275, under the NASA Prime Contract No. NNX17AJ86 with Dr. Kai F. Goebel serving as the Technical Monitor. Professor Yongming Liu is the Principal Investigator on the Prime Contract at the Arizona State University.

NOMENCLATURE

A	Aircraft type (B-737, A320…)
$f(.)$	Aircraft type dependent climb/descent altitude rates given as a function of altitude.
h	Altitude
R_e	Radius of earth
V	Aircraft velocity
V_{min}	Minimum permissible aircraft velocity
V_{max}	Maximum permissible airspeed
W_E	Component of the ambient wind in the East direction
W_N	Component of the ambient wind in the North direction

REFERENCES

Abiy, T., Pang, H., & Khim, J. (2018). (May 03) *Dijkstra's Shortest Path Algorithm*. Retrieved from BRILLIANT: https://brilliant.org/wiki/dijkstras-short-path-finder/

Andrews, J. W., Welch, J. D., & Erzberger, H. (2001). Safety Analysis for Advanced Separation Concepts. *Air Traffic Control Quarterly, 14*(1), 5-24.

Bilimoria, K. D., Sridhar, B., Grabbe, S. R., Chatterji, G. B., & Sheth, K. S. (2001). FACET: Future ATM concepts evaluation tool. *Air Traffic Control Quarterly, 9*(1), 1-20.

BTS. (2018). (July 31) *Fatalities*. Retrieved from Bureau of Transportation Statistics: https://www.bts.gov/content/transportation-fatalities-mode

Eurocontrol. (2018). (April 30) *Base of Aircraft Data*. Retrieved from http://www.eurocontrol.int/services/bada

FAA. (2005). (September 16) *FAA Advisory Circular*. Retrieved from Aircraft Surveillance Systems and Operations: https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC%20120-86.pdf

FAA. (2018). (April 30) *Coded Instruments Flight Procedures*. Retrieved from https://www.faa.gov/air_traffic/flight_info/aeronav/digital_products/cifp/download/

FAA. (2018). (April 30) *FAA Accident and Inciden Data*. Retrieved from https://www.faa.gov/data_research/accident_incident/

Google. (2018). (May 03) *Google MyMaps*. Retrieved from My Maps Help: https://support.google.com/mymaps

Napolitano, M. R. (2012). *Aircraft dynamics: From modeling to simulation.* Hoboken, NJ: John Wiley and Sons.

NASA. (2018). (April 30) *NASA Aviation Safety Reporting System*. Retrieved from https://asrs.arc.nasa.gov/search/database.html

NOAA. (2018). (April 30) *Aviation Weather Center*. Retrieved from https://www.aviationweather.gov/

NTSB. (2018). (April 30) *NTSB Aviation Accident Report*. Retrieved from https://www.ntsb.gov/investigations/AccidentReports/Pages/aviation.aspx

Oracle. (2018). (May 04) *JAVA*. Retrieved from JAVA: https://www.java.com/en/

Park, S. G., & Yang, B.-J. (2016). Simulating Flight-Deck Mode-Confusion Incidents using Human Error Templates. *AIAA Modeling and Simulation Technologies.* San Diego, CA: AIAA 2016-1184.

Prevot, T., Smith, N., Palmer, E., Mercer, J., Lee, P., Homolar, J., & Callantine, T. (2006). The Airspace Operations Laboratory(AOL) at NASA Ames Research Center. *AIAA Modeling and Simulation Technologies.* Keystone, CO: AIAA-2006-6112.

Python Software Foundation. (2018). (May 03) *Python*. Retrieved from Python: https://www.python.org/

Roychoudhury, I., Spirkovska, L., Daigle, M., Balaban, E., Sankararaman, S., Kulkarni, C., Poll, S., & Goebel, K. (2015). *Real-Time Monitoring and Prediction of Airspace Safety.* Moffett Field, CA: NASA.

Stanton, N. A., Harris, D., Salmon, P. M., Demagalski, J., Marshall, A., Waldmann, T., Dekker, S., & Young, M. S. (2010). Predicting Design-Induced Error in the Cockpit. *Journal of Aeronaustics, Astronautics and Aviation, Series A, 42*(1), 001-010.

The MathWorks Inc. (2018). (May 03) *MATLAB*. Retrieved from Math. Graphics. Programming: https://www.mathworks.com/products/matlab.html

BIOGRAPHIES

P. K. Menon is the Chief Scientist and the Chief Operating Officer at Optimal Synthesis Inc. He holds a Ph. D. degree in Aerospace Engineering from Virginia Polytechnic Institute and State University, a Master of Aerospace Engineering degree from the Indian Institute of Science, and a Bachelor of Mechanical Engineering degree from the National Institute of Technology. He has published over 200 papers at international conferences and peer-reviewed archival papers and has served as the Associate Editor for the Journal of Guidance, Control, and Dynamics, and the IEEE Transactions on Control Systems Technology. Dr. Menon is a Fellow of the AIAA and a Senior Member of the IEEE.

Bong-Jun Yang is a senior research scientist at Optimal Synthesis Inc. He obtained his Ph.D. in aerospace engineering and an M.S. in applied mathematics in 2004 from the Georgia Institute of Technology, Atlanta, GA. He has published papers in the area of dynamic system modeling and control, air traffic management, hybrid system formulation for human-machine interactions. He is a senior member of the AIAA and a member of IEEE control system society.

Parikshit Dutta is a research scientist at Optimal Synthesis Inc. He obtained his Ph.D. from Texas A&M University, College Station, TX in 2011. He was an associate researcher at INRIA from 2011-2013 and a postdoctoral fellow at Duke University from 2013-2014. He has published several articles in AIAA and IEEE journals and conferences on uncertainty prediction methods.

Sang Gyun Park is a research scientist at Optimal Synthesis Inc. He received a Ph.D. in Aerospace Engineering and ab M.S. in Mathematics from the Georgia Institute of Technology (2014). He also holds an M.S. (2004) and a B.A. (2002) degrees in mechanical and aerospace engineering from the Seoul National University, South Korea. He worked as a research engineer at the Hyundai Heavy Industries (2007-2009) and the Hyundai Mobis (2004-2006). He has published several articles in AIAA and IEEE journals and conferences.

Oliver Chen is a research engineer at Optimal Synthesis Inc. He obtained his MS degree in Computer Science from the George Washington University, Washington, DC and a BS degree in Civil Engineering from the National Cheng Kung University, Taiwan. He has over 10 years of experience in designing, developing and maintaining large scale web-based and user-focused application systems.

Victor Cheng is the President of Optimal Synthesis Inc. He received his Ph.D. and MS degrees in Electrical Engineering and Computer Sciences from University of California – Berkeley, and his BS degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute. He has served as Associate Editor of the IEEE Transactions on Control Systems Technology and the Journal of the American Helicopter Society. He has also served on the AIAA Technical Committee on Guidance, Navigation, and Control, the AIAA Program Committee on Aerospace Traffic Management, and the American Helicopter Society Technical Committee on Avionics and Systems. Dr. Cheng is an Associate Fellow of the AIAA.