A Modeling Architecture for Capturing CNS Uncertainty and Prediction Errors in the NAS

Paul Cobb[1]
Seagull Technology Inc, Campbell, California, 95008

Dr. Karlin Roth[2]
NASA Ames Research Center, Moffett Field, California, 94035

and

Ari Stassart[3], Dr. George Couluris[4], Doug Sweet[5] and Dr. George Hunter[6]
Seagull Technology Inc, Campbell, California, 95008

The Airspace Concepts Evaluation System (ACES) is a distributed fast-time computer simulation of gate-to-gate flight operations throughout the National Airspace System (NAS). ACES is being developed by the NASA Virtual Airspace Modeling and Simulation project to assess future Air Traffic Management concepts of operation. ACES employs multi-trajectory modeling in an agent-based software structure to describe the behavior and interactions of Traffic Flow Management, Air Traffic Control, Flight Deck, and Airline Operational Control entities and associated communication, navigation, surveillance and weather information technologies. This paper discusses the ACES architecture for modeling Communications, Navigation and Surveillance (CNS) uncertainties and predictions and how those uncertainties propagate and affect agent decisions, delays and capacities within the NAS.

I. Introduction

Uncertainty in the National Airspace System (NAS) has multiple sources. Major sources of uncertainty are the various Communication, Navigation, and Surveillance (CNS) systems used throughout the NAS. Communications systems can have bandwidth limitations, range limitations, delays, and outages. Navigation systems can introduce errors due to the quality of the sensor(s) and dilution of precision effects. Likewise, surveillance systems can introduce position / velocity errors, have limited range, and can be susceptible to outages.

Predictions of the future state are also a significant source of uncertainty within the NAS. The accuracy of modeling future aircraft states can vary dramatically depending on the availability of data on the aircraft (e.g., current position and velocity, flight plan, aircraft performance, weight), environmental factors (e.g., winds, temperature, severe weather locations), and the quality of the trajectory model generating predictions.

[1] Technical Director, Air Transportation Systems, 1700 Dell Avenue, Campbell CA 95008, AIAA Member.
[2] Chief of Aerospace Operations Modeling Office, Airspace Systems, NASA Ames Research Center, Moffett Field CA 94035, AIAA Member.
[3] Technical Lead Software Engineer, Air Transportation systems, 1700 Dell Avenue, Campbell CA 95008, AIAA Member.
[4] Senior Scientist, Air Transportation systems, 1700 Dell Avenue, Campbell CA 95008, AIAA Member.
[5] Director of Air Transportation Systems, Air Transportation systems, 1700 Dell Avenue, Campbell CA 95008, AIAA Member.
[6] Senior Scientist, Air Transportation systems, 1700 Dell Avenue, Campbell CA 95008, AIAA Member.

In implementing this architecture we specifically addressed the following issues:

1) Introducing uncertainty and prediction errors
2) Tracking uncertainty and prediction errors
3) Propagating uncertainty and prediction errors
4) Categories of distribution of uncertainty and prediction errors
5) Measuring the effect of uncertainty and prediction errors
6) Replication of results

B. Introducing uncertainty and prediction errors

In order to provide for an accurate model of uncertainty within the NAS, the system must allow for multiple sources of information with their accompanying errors. Each agent has the capability to collect these different data sets from any source as well as to provide its own estimates and calculations for these data inputs. Each simulation run contains some accepted 'truth' data sets that all other data sets may be compared against. Uncertainty and prediction errors may be introduced either through the modification of the 'truth' data or via a separate lower fidelity model within the agent itself.

C. Tracking uncertainty and prediction errors

Each agent handles and tracks each of the 'truth' data sets and uncertainties to be employed in its own models. Errors introduced to the 'truth' data are easily tracked separately for each agent. However, new models with varying levels of fidelity track or serialize the complete data set generated by the independent model in order to compare against the 'truth' data.

D. Accumulation or propagation of uncertainties

Uncertainties accumulate or propagate throughout the system. For example, an airport Air Traffic Control (ATC) agent propagates a delay back along the flight path of an aircraft expected to arrive at some future time. In addition there is a sector overload along the same flight path which the flight will encounter. The airport Traffic Flow Manager (TFM) may be unaware of this sector overload and predicts a time of arrival that does not account for it, inducing an additional amount of error within the prediction. This requires that the system provide the capability to allow for the accumulation or propagation as well as recording the sources of these errors as they are introduced. The architecture additionally accounts for models or decision support tools that may or may not be able to negate the propagation of these errors via more accurate prediction algorithms or adjustment from independent data sources.

E. Categories of uncertainty distribution

Estimated Flight States or Predicted Flight Trajectories, or estimated states of environmental data such as winds, may be widely distributed to the ACES NAS Agents or may be limited to one or a few NAS Agents. The architecture is capable of supporting the following distributions of estimated states:

Global – the need to represent common uncertainty for all NAS Agents (e.g., wind predictions)
Group / Region – the need to represent common prediction for a type of NAS Agent or set of NAS Agents (e.g., same trajectory model for the ARTCC ATC Conflict Detection & Resolution)
Individual – the need to tailor model specifically to an single NAS Agent (e.g., evaluate a effect of a Decision Support Tool at a single location or region)

F. Measuring the effect of uncertainty and prediction errors

The design necessarily allows for the ease of comparison and measurement of the levels of uncertainty. While uncertainties may be tracked as an error attached to the 'truth' data, some systems may contain separate lower fidelity models that must be compared against the 'truth' data in order to measure the level of uncertainty. This approach allows for a Monte Carlo style capability for analyzing and determining the best possible combination of varying components within the overall NAS infrastructure.

G. Storage and replication of results

The architecture provides for the capability to record and reintroduce the same errors at specific points within the simulation. This is done in order to reproduce earlier results or introduce new errors within the system and review the combined results of these uncertainties.

II. Introduction to ACES architecture

The National Aeronautics and Space Administration (NASA) has established the Virtual Airspace Modeling and Simulation (VAMS) Project to examine proposed operational concepts for increasing the capacity of the NAS. As part of this effort, VAMS is developing the Airspace Concepts Evaluation System (ACES) to simulate NAS operations. ACES is a fast-time, computer simulation of local, regional and nationwide factors covering aircraft flight from gate departure to arrival.

ACES' overarching objective is to provide a flexible NAS simulation and modeling environment that can assess the impact of new NAS tools, concepts, and architectures, including those that represent a significant departure from the existing NAS operational paradigm. To meet this objective, ACES utilizes the High Level Architecture (HLA) and an agent-based modeling paradigm to create the large scale, distributed simulation framework necessary to support NAS-wide simulations. HLA is a set of processes, tools and middleware software, developed by the Department of Defense, to support "plug-and-play" assembly of independently developed simulations. For ease in integration and efficient runtime execution of the simulation, the ACES simulation framework employs agent-based modeling. The ACES agent-based processes simulate airspace and aircraft operations.[1,2,3]

The ACES architecture is designed to accommodate models of each operational component of the NAS. ACES contains models for Air Traffic Management (ATM), encompassing Air Traffic Control (ATC) and Traffic Flow Management (TFM) operations; aircraft dynamics; and en route winds. The modeling accounts for airspace and airport designs and procedures, including airport visual flight rules (VFR) and instrument flight rules (IFR). Agents represent present-day NAS operations and include Air Traffic Control System Command Center (ATCSCC), the en route Air Route Traffic Control Center (ARTCC), Terminal Radar Approach Control (TRACON), Airport Traffic Control Tower (ATCT), and aircraft and pilot entities. [1,2,3]

The agents are autonomous entities, each simulating operations according to algorithmic and data processing logic defined by its model based on information exchanged with other agents and supporting constructs. In the ACES modeling concept, the unifying factor is the aircraft trajectory. Each of the basic NAS agents performs modeling functions that operate on flight trajectories. AOC agents conduct pre-takeoff flight planning to define four-dimensional (4-D) desired/requested trajectories, post-takeoff flight following to coordinate trajectory flight plan revisions, and flight schedule revisions based on delay, diversion and cancellation assessments. The TFM agents conduct local, regional and nationwide flow constraint assessments and determinations based on flight plan, traffic surveillance, meteorological and airspace and airport status and constraint information. The ATC agents conduct trajectory intervention assessments and resolutions based on traffic situation surveillance, procedures, separation rules and TFM constraints, trajectory state and intent, meteorological and aircraft performance information. The aircraft trajectory simulations are based on trajectory state estimate and intent, planned trajectory, ATC clearance, aircraft performance and meteorological information. [1,2,3]

Each agent has information describing a trajectory and performs action on the trajectory based on this information. But each agent does not necessarily have the same information as another, and none may know the true trajectory state of a flight. The ACES design enables each agent to maintain separate or different trajectory data and trajectory management logic, hence implementing a <u>multi-trajectory modeling</u> concept. ACES also maintains a model of trajectory truth for each flight. Hence ACES would have the ability to model effects of trajectory estimation errors concurrently with the modeling of agent operational processes. The degree of accuracy and fidelity with which the agents operate on trajectories depends on the technologies and functional capabilities of the NAS operational concept being simulated. In ACES these are represented by the logic encoded into each agent and associated modeling parameters.[3]

III. Introducing uncertainty and prediction errors

Uncertainty in the NAS has multiple sources. For instance, there is uncertainty due to the various communication, navigation, and surveillance systems used throughout the NAS. Communications systems can have bandwidth limitations, range limitations, delays, and outages. Navigation systems can introduce errors due to the quality of the sensor(s). Surveillance systems can introduce position / velocity errors, have limited range, and are susceptible to outages.

Predictions of the future aircraft state are also a source of uncertainty within the NAS. The accuracy of modeling future aircraft states can vary dramatically depending on the availability of data on the aircraft (e.g., current position, flight plan, aircraft type, performance and weight) and environmental factors (e.g., winds, temperature, severe weather locations) and the quality of the trajectory model generating predictions.

In addition to uncertainties, quantities within the NAS may be computed, estimated or predicted separately by different agents. For instance, an aircraft state is estimated by the ATC using surveillance and it is estimated by the

flight deck navigation system. Therefore, our design allows quantities to have multiple sources and uncertainties. We focus on the aircraft trajectory and flight plan as these are the most common and usually the most important sources of error and uncertainty.

We divide the uncertainty sources into three categories:

1) Intention prediction error (Surface and Airborne),
2) Trajectory modeling error,
3) Initial condition error.

Intention prediction errors occur when the trajectory predictor is modeling a different intent than the AOC or flight crew intent. The AOC and flight crew intent changes with time so intent modeling may be correct at one point in time and incorrect at another. For instance, ATC may have the correct planned gate departure time early in the day for a flight. If the flight is delayed due to mechanical problems then the ATC predicted gate departure time accrues error. When the problem is resolved and the ATC predictor is updated, then the error is resolved. The following tables list typical sources of uncertainty within the system.

Table 1. Typical intention prediction error sources in the surface phase

Error source
Flight cancellation
Preflight route change (eg, SWAP)
Gate departure delay
Taxi delay due to air traffic or weather
Taxi time variation due to surface traffic

Table 2. Typical intention prediction error sources in the airborne phase

Error source
Congestion delay or reroute
CD&R
Storm reroute
Clear Air Turbulence avoidance
Expedite flight
Preferred routing
Non amended FP
Discrepancy in route data

Trajectory modeling errors occur when the predictor does not correctly model the way the aircraft flies its trajectory. The most significant source in this category is usually error in the wind forecast. Table 3 lists trajectory modeling error sources.

Table 3. Typical trajectory modeling error sources

Error source
Speed variations
Speed change maneuver model error
Turn maneuver model error
Climb / descent model error
Wind forecast error

Initial condition errors occur when the state of the aircraft is not measured accurately. Table 4 lists initial condition error sources.

Table 4. Typical initial conditions error sources

Error source
Navigation errors
Surveillance and tracking errors

Tables 1 – 4 list a wide range of error and uncertainty sources. Many of them contribute only minor levels of trajectory prediction error. For ACES Build 3 we implemented and demonstrated two of the above error sources: the wind forecast error (Table 3) and gate departure delay error (Table 1).

In this implementation we focus on the truth and uncertainty datasets for the aircraft flight data. Specifically, we track the initial scheduled flight plan, updated flight plans (based on route changes), the 'true' flight state, the estimated flight state and the predicted flight state. These were previously defined above.

The models that we specifically address are uncertainty with wind errors (Uncertainty generated by different sources using different wind measurements for flight time adjustments), gate delay errors (Uncertainty generated by different sources having varying gate delays) and the estimated state for errors of the surveillance type. We describe the implemented models below as well as describe some specific concepts used throughout this paper.

H. Specific Sources of Uncertainty

The following represent a few simple error uncertainty models that are being used to test out the ACES architecture. These tests exercise all of the basic functionality of the uncertainty architecture and provide a basis for creating more complex models of uncertainty.

1. Wind Forecast Errors

The model for introducing the first type of error is a wind time bias. The ACES flight agent uses actual wind times, but the rest of the agents use winds with a 30 minute time bias. For example, if the simulation time is noon, then the Flight agent is using winds for 12:00 PM while ARTCC ATC and the rest of the agents are using winds for 12:30 PM. The pre-processed flight data contain this wind forecast error.

2. Gate Departure Delays

The second model of uncertainty is gate departure delay. Each flight has a randomly generated gate delay. To make this repeatable a random seed is required. This design implements a uniform distribution gate delay model with delays between 0 to 10 minutes. Other algorithms that could be implemented are an exponential model with a cutoff time, or a constant slope model. ACES implements the gate delay uncertainty algorithm for departing flights.

3. Surveillance Errors

The third model of uncertainty involves generating two surveillance errors. Figure 1 shows an example of an estimated state error. In this example, the estimate state of the flight is southwest of the current flight position. For purposes of demonstrating the estimated state error, the current true state generated by ACES during every 1 minute update is used to generate two current estimated states by offsetting the position of the flight by 0.25 NM and 0.5 NM in a due northerly direction from the current position.

ACES employs an agent infrastructure in order to provide a simple mechanism for providing independent models within the system. Each agent has access to activities that provide the actual

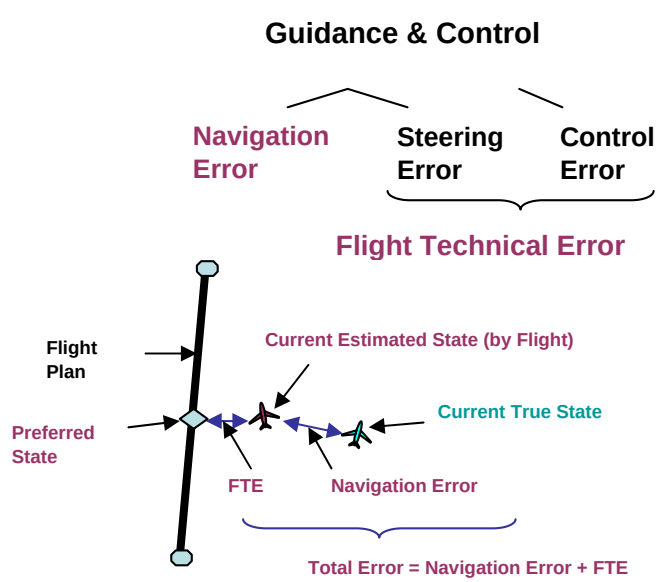

Figure 1. Estimated flight state example (Flight perspective)

modeling code. These activities are executed at defined intervals and operate on the data provided by the agents. Each agent may then also publish and subscribe data via messages within these activities as well. The agents communicate via these messages. This allows the ACES developer to easily introduce new sources of uncertainty and track or modify any changes to the data based on the modeled uncertainties.

IV. Descriptions of concepts within the ACES uncertainty architecture

To support the understanding of this architecture, it is important that a common set of terminology is used. This helps to avoid confusion and misunderstanding of the ACES uncertainty modeling concepts.

I. Flight Plan

A set of sequential time or position-based flight events. Typical data that would be included in a flight plan would be the origin and destination airport, the various fixes or waypoints along its planned flight path and cruise altitude and speed.

J. Static (Flight) Data

A set of data about a flight that does not change during the execution of the simulation. Typical data that would be included are flight identification, ETMS identification, airline leg number, stream class, airline, airline flight number, airline tail number, departure airport, arrival airport, and various scheduled times (departure gate, takeoff, departure fix, arrival fix, landing, arrival gate).

K. True Flight State

The state of an aircraft at a given time without any error. By definition, the true flight state must be a past or current state of the flight. To illustrate, Figure 2 shows the plot of one of AC 1's true state parameters, namely the altitude of the flight, at $t = t_3$. Between time t_1 and t_3, the true flight state is shown in the dark solid line. Beyond time t_3, a predicted altitude is shown, but it is <u>not</u> part of the true flight state data set at time $t = t_3$. This is not to say that a future predicted flight state could not be the same as the eventual true flight state at a time in the future (say at time $t = t_4$ in the example in Figure 2). For a predicted flight trajectory to represent aircraft true states in the future, the predicted flight trajectory must account for all interactions with other NAS agents and be based on the current True Flight State and true environmental data for the predicted time frame. Note that future flight trajectory predictions cannot be verified until the simulation is run and the true flight state is measured.

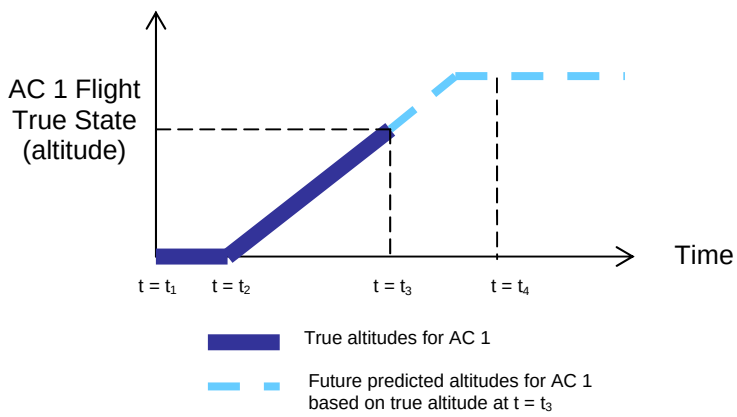

Figure 2. True flight state example

L. Flight Data Set (Class)

The set of data used to characterize each individual flight. The Flight Data Set is an abstract concept that includes the complete set of data used to describe each individual flight. As Figure 3 illustrates, it includes static data which do not change during the course of the flight. It also includes true flight state data for the current state of the flight at a given time. In addition, it may include one or more estimated flight states representing the current flight state. For future predictions of the aircraft's state, the FDS includes one or more sets of flight plan data and predicted flight trajectory data. The FDS may contain multiple flight plans for a single flight and may contain multiple predicted flight trajectories. The number of different estimated current flight states and the number of flight plans and predicted flight trajectories depends on how many independent assessments of the aircraft state are required for a specific simulation.

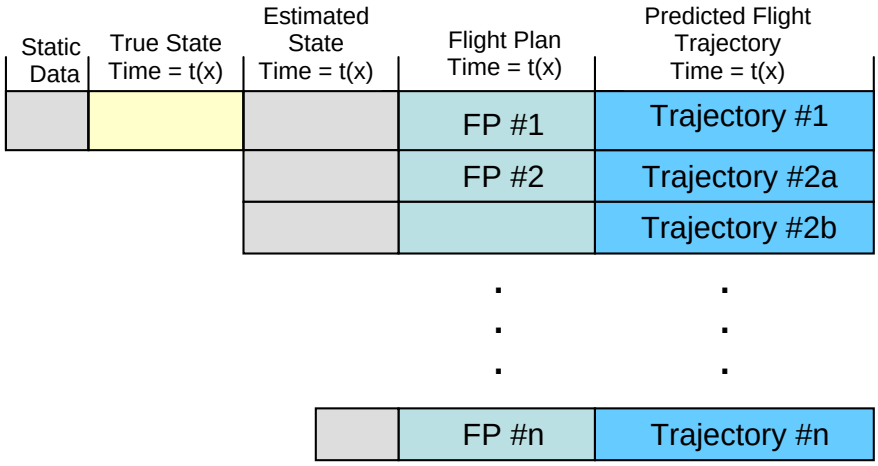

Figure 3. Flight data set illustration

V. Tracking uncertainty and prediction errors

This architecture builds a framework for allowing each agent to access and track the 'truth' data sets as well as data sets containing uncertainty. This can be done in one of two ways, either to track the truth data and the difference between the uncertainty and the 'truth' or keep them as two complete separate entities with the uncertainty data being a combination of truth and uncertainty. Using either path allows us to track sources of uncertainty as well. Each new source of uncertainty may allow for a completely new instance of the data set, which is an addition to the truth data set, or a compilation of the truth and all other uncertainty sources. While this will most likely increase the memory footprint and reduce performance of the system, this is a necessary element to allow the researcher to determine the magnitude and scope of the uncertainties and delays introduced by each source.

These sources of error and induced errors are tracked within the simulation in several ways. Each agent may obtain its' own copy of the data through calculation based on input data that may or may not contain errors or via subscription to data that has been produced by other agents. Agents may 'publish' their data allowing any 'subscribing' agent to access and use that data. Each agent then may collect multiple instances of a data item. It is this mechanism that allows each agent to track its own data as well as data obtained from other agents. All data that are communicated as messages between agents are then recorded for later data analysis with the name of the agent producing the data and any other relevant information desired by the developer. Additionally any internal or private data may be recorded at the request of the developer. These data are then available via database queries for analysis and comparison. In the Build 3.0 implementation of ACES we have provided an Application Programming Interface to allow any agent to calculate, publish and subscribe the flight plan and flight trajectory data.

One of the examples that we track here is the uncertainty found through varying levels of winds used by different agents. In this case the flight computes its trajectory using the actual winds over the course of its flight. All other agents compute the trajectory using predicted winds. Each agent holds its own copy of the flight trajectory and bases its predicted entry/exit and arrival times on these winds. This causes flights time changes based on wind direction both in the positive and negative direction. This in turn causes discrepancies within expected boundary crossing times and arrival times. TFM planning is then further affected by inaccurate sector counts and airport arrival rates. This type of run and analysis then allows

In the same manner we also demonstrate tracking gate delay errors. Each agent obtains a randomly generated gate delay offset that is then used for determining the flight times. This is done calculating a random delay, adding it to the scheduled flight plan of the system and then calculating the new flight trajectory based on the new flight plan. The system then retains and records both the original flight plan and trajectory and the modified trajectory. These data are then published to other agents for use in further modeling.

We also track the estimated state. The estimated state provides an additional set of data with its own uncertainties that may be independent from a received flight state and the predicted flight state. This is described further below.

M. Estimated Flight State

The state of an aircraft, including uncertainty, at a given time. For the purposes of this design, the estimated flight state refers to current (or past) flight states and does not refer to future predicted flight trajectories. Figure 1 illustrates the estimated flight state and the various error sources, from the flight perspective. The aircraft to the far right represents the true flight state of the aircraft at the current time (labeled as the *current true state*). The desired state at the current time, from the aircraft perspective, is represented in Figure 1 as the *preferred state*. The first error source that is introduced is the navigation error, representing the error in the aircraft position sensors. The second source of error is the Flight Technical Error (FTE) that represents the error in controlling the aircraft (by either steering or control system inaccuracies). We therefore could represent the current estimated flight state, from the flight perspective, by modeling these error sources and including them in the Flight model.

Another example of estimated flight state is shown in Figure 4. Figure 4 shows the estimated flight state from the perspective of an ATC agent that is using surveillance (e.g., radar, ADS-B) to assess the state of the flight.

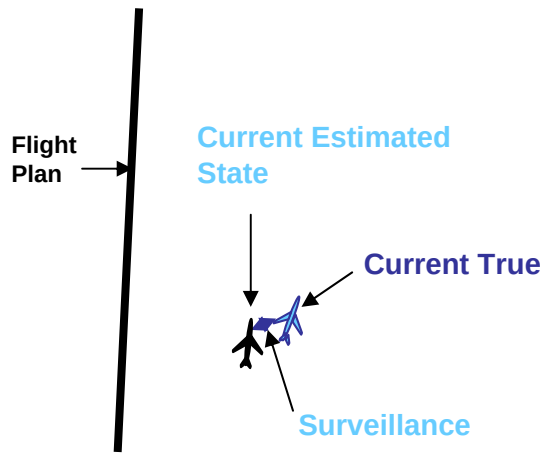

Figure 4. Estimated flight state example (ATC perspective)

VI. Accumulation or propagation of uncertainties

This architecture allows uncertainties to propagate and accumulate within the system. As agents introduce new errors into the data and then publish these data to other agents, these errors may begin to grow, dependent upon the errors introduced by each agent. This can be handled by the system in a few different ways. Each agent may subscribe to data that contain errors and then introduce its own uncertainties within the data or the system may create a new instance of the data based on the subscribed data or other inputs. In this manner we allow the researcher to model and track the propagation of uncertainties as they would be represented within the actual NAS. As well agents may modify the data based on better knowledge to allow the researcher to model decision support tools or more accurate information obtained by the agent.

For example the flight plan could be modified with some added delay at the originating airport. The updated flight plan would be published or propagated to other agents who would then base the flight trajectory on these data. If the delay at the airport was increased or decreased by some amount and these data was not immediately published to the other agents, the downstream TRACON and Centers would have incorrect timings for boundary crossing and arrival times. This in turn could cause further delays within the system based on other factors such as sector loads.

The system records the changes to the flight plan as well as the actual takeoff times, so that the researcher could modify the models to remove the error and then compare the delays generated by uncertainties within the takeoff time.

VII. Categories of uncertainty distribution

The architecture must be able to handle uncertainties at varying levels within the system. To do this we introduce three levels of uncertainty distribution, Global, Regional or Group and Individual. Global uncertainty examples include items that can affect uncertainties in all NAS Agents such as winds. Regional or Group uncertainties represent more localized sources such as local weather, similar or common radar systems within a center and communication latencies between a ARTCC TFM and the center's TRACON TFMs. Individual uncertainty consists of uncertainties tailored to particular models, such as the radar at specific terminal areas or airports.

The first uncertainty implementation demonstrates a Global uncertainty within the system. All winds are affected and each agent may have differing results dependent upon which version of the wind data it is using. The Gate Delay Error is an example of either a Group or Regional or on a Global scale depending on how it is implemented. Adding random gate delays to a single airport gives us a Regional effect. The Estimated state example demonstrates uncertainty on an individual agent basis. This type of uncertainty can also be applied to a group of aircraft or the entire NAS.

VIII. Measuring the effect of uncertainty and prediction errors

In order to compare and measure the effects of uncertainty within the NAS, the system records not only flight trajectories, but also all flight plans, boundary crossings, surface and airborne delays, flight restrictions implemented by the various TFM and ATC agents as well as any other message published between agents. This allows the developer to easily compare flight times for specific configurations and measure any delays produced by these errors.

This system records all messages to a SQL database as separate tables. SQL queries exist to process these data for comparison. Additional post processing SQL queries may also be written to easily compare any data message easily. The system also provides a simple Application Programming Interface (API) that allows the developer to output any additional data existing within the models.

This allows the researcher to 'search' for specific solutions with the desired results via a Monte Carlo style series of runs. A researcher might desire to determine the most optimal configuration for a specific airport under certain operating conditions. By running many simulations, each with a slightly different terminal configuration the researcher could compare the overall delays and determine the 'best' solution for a particular desired result.

A further example of measuring the effect of uncertainty is to determine the reduction in the amount of delay that a new radar system provides. The model would be updated to introduce the current amount of uncertainty in place within the system for an older piece of equipment. The model would then be updated to represent the smaller errors produced by the new equipment. The delays produced by each run could then be compared to show the improvement within the system provided by the new equipment. Additionally the system could be configured to represent a mixture of old and new equipment to simulate the staged introduction of new systems.

The following describes how the system uses various predicted flight states and current estimated states for comparison in these cases.

N. Predicted Flight Trajectory

A 4-D predicted future state of the aircraft at a specific time. A predicted flight trajectory is derived from a flight plan using a trajectory generation model (e.g., MPAS) and an initial state of the aircraft at a specific time. A predicted flight trajectory can be represented in a

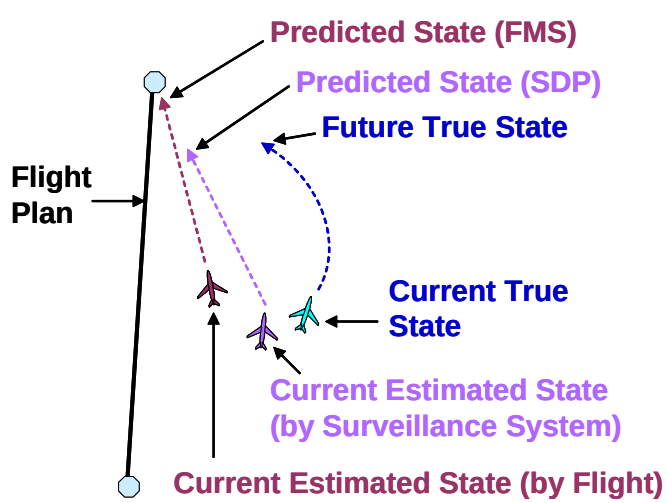

Figure 5. Multiple predicted flight trajectory example

number of ways, depending on end user needs. It can be represented as a set of 4D points along the predicted trajectory to perform conflict detection and resolution or it could be represented as crossing times at specified locations to perform sector loading calculations.

Predicted flight trajectories are characterized by the initial flight state, the look-ahead timeframe, the frequency of updates, the form of the trajectory component (e.g., 4D points, facility crossing times), environment (wind, temperature) and the trajectory model (including A/C performance model data) used to predict future flight states. The uncertainty associated with a given predicted flight trajectory depends on all of these. For example, the initial state of the aircraft used for a given predicted flight trajectory could be accurate (i.e., very close to the true flight state) or it can have significant error (i.e., based on an estimated flight state with significant uncertainty). Figure 5 shows how the different aircraft state estimates and predictions can differ from the respective true values. The flight will be using its Flight Management System (FMS) to predict future intent and facilities will be using Surveillance Data Processing (SDP) to generate predicted trajectories.

4. Predicted Flight Trajectory Examples

Figure 6 illustrates various levels of uncertainties that can be introduced through the use of different predicted flight trajectories. At the time marked NOW in Figure 6, the ARTCC is evaluating the various aircraft depicted by either light blue or dark circles. The light blue circles represent aircraft not within the facility. For these aircraft, the ARTCC has limited knowledge (flight plan only) of their current state since they are not measuring their position directly. The dark circles represent aircraft that are within the facility and whose current state is directly measured by the facility. For these aircraft, the ARTCC has improved current state knowledge (current state thru radar, but current plus intent thru ADS-B).

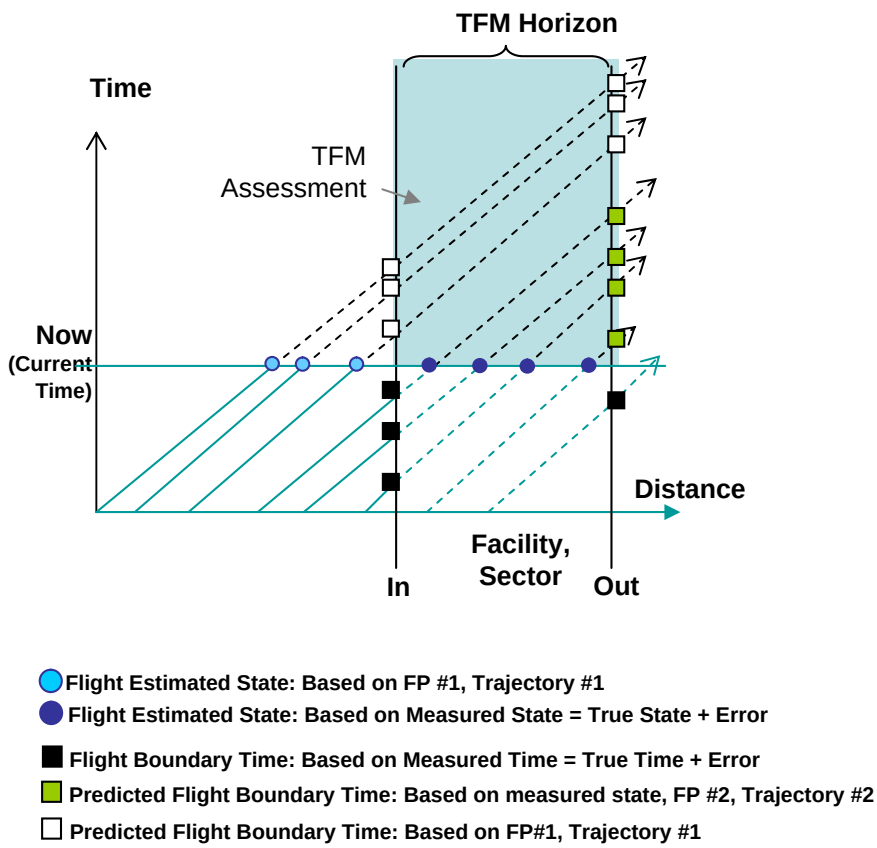

Figure 6. Levels of Uncertainty

The predicted boundary crossing times for the flights described above are derived from two different predicted flight trajectories. The first trajectory is based on flight plan derived initial conditions and a flight plan (FP #1) which may not include all past or planned changes to the flight path prior to entering the facility. The second trajectory is based on measured initial conditions for aircraft within the facility and an improved flight plan (FP #2) which contains within-facility intent, resulting in a more accurate prediction of facility exit times.

IX. Storage and replication of results

A deterministic simulation approach is used for ACES. This allows us to record and recreate specific scenarios with small varying inputs in order to determine the impact of these changes across large or small areas of interest. The system provides a simple interface to save any messages published by any of the agents. Additionally the

system provides an interface to allow developers to record private or non-message data within the system. The ACES front end provides a graphical interface that allows the developer to pick and choose what messages are recorded for ease of analysis. The development environment provides a straightforward Application Programming Interface that allows the developer to add new private data items to this interface as well for simple storage and analysis of the data.

ACES also provides API calls to introduce random errors into the data and record the random seeds for these runs in order to replicate the data at a later date. This allows researchers to investigate more complex comparisons of data. A researcher can develop an experiment where specific uncertainties are applied to one or more segments of the system and record the changes to the overall system. These individual 'random' uncertainties can then be applied in tandem in order to determine the difference with the combinations of induced uncertainties.

X. Conclusion

This paper presents the issues behind introducing Communications, Navigation and Surveillance uncertainties and predictions into a deterministic simulation, an architecture for handling these issues and the results from the implementation of this architecture. Our architecture provides a method to introduce these uncertainties and to propagate and affect agent decisions, delays and capacities within the NAS. The addition of the capability to model and track uncertainty within the ACES system provides an invaluable addition to its capabilities. This will allow ACES to more accurately model the current NAS as well as model future implementations and provide cost benefits analysis to systems that contain varying levels of uncertainty. ACES can show the benefits of future NAS communications, navigation and surveillance systems through the analysis of the decrease in delays provided by these new systems.

Acknowledgments

The ACES simulation system was developed for the NASA Virtual Airspace Modeling and Simulation Project under the Air Traffic Management System Development and Integration Contract. The development team, which is led by the Raytheon Company, includes Seagull Technology, SAIC, Intelligent Automation, Inc., TITAN, Metron Aviation, Booz-Allen Hamilton and others.

While selected references are sited, we emphasize that this effort reflects the common knowledge and expertise accumulated over 30 years of collective work in airspace, airport, flight dynamics, traffic demand and computer modeling conducted by many, many individuals and organizations, far too many to identify one-by-one. Much of this previous work has been sponsored by FAA, NASA, DOD and various other public and private members of the aviation and computer science communities.

References

1. Aronson, J. S., Manikonda, V., Peng, W., Levy, R. and Roth, K., "An HLA -Compliant Agent-Based Fast-Time Simulation Architecture For Analysis Of Civil Aviation Concepts,", Simulation Interoperabili ty Standards Organization (SISO), 2003 Spring Simulation Interoperability Workshop, Paper 03S -SIW-098, Orlando, FL (April 2003).

2. Sweet, D. N., Manikonda, V., Aronson, J. S., Roth, K., Blake, M., "Fast-Time Simulation System For Analysis of Advanced Air Transportation Concepts ," American Institute of Aeronautics and Astronautics Modeling and Simulation Technology Conference, AIAA Paper 2002-4593, Monterey, CA (August 2002).

3. Couluris G. J., Hunter, C. G., Blake, M., Roth, K., Sweet, D., Stassart, P., P hillips, J., Huang, A., "NATIONAL AIRSPACE SYSTEM SIMULATION CAPTURING THE INTERACTIONS OF AIR TRAFFIC MANAGEMENT AND FLIGHT TRAJECTORIES", American Institute of Aeronautics and Astronautics Modeling and Simulation Technology Conference, AIAA Paper 2003-5597, Monterey, CA (August 2002).