

STOCHASTIC AIRSPACE SIMULATION TOOL DEVELOPMENT

*Seamus M. McGovern and Amanda Kalish, U.S. DOT National Transportation Systems Center,
Cambridge, Massachusetts*

Abstract

Modeling and simulation is often used to study the physical world when observation may not be practical. The overall goal of a recent and ongoing simulation tool project has been to provide a documented, lifecycle-managed, multi-processor capable, stochastic simulation capability enabling analysis of procedures and equipment for aircraft flight in the national airspace system (NAS). The tool is a Monte Carlo-based computer simulation program that contains the stochastic models of most components of the NAS, including navigation aids, surveillance systems, pilots, aircraft, air traffic controllers, and weather. It is also combined with discrete artifacts such as FAA database-supplied runway sizes and configurations, and obstacles. In addition, the tool possesses a fast-time airframe-type-specific kinematic aircraft model, a high-performance random number generator, a precise WGS-84-compliant elliptical earth model, and a graphical user interface integrating world-wide photo-realistic airport depictions and real-time three-dimensional animation.

This paper presents an overview of the software tool and its formulation, including programming languages, environments, and other development tools; selection, design, and implementation of the mathematical models; and reviews the verification and validation processes.

Background

Before any new flight operations are implemented, it is of interest to show that they will have no adverse effects on the safety of airline travel. To accomplish this, computer simulations may be used to predict the probability of aircraft separation violations. The tool discussed in this paper seeks to accomplish this using a variety of detailed models. One goal of this software tool has been flexibility in design to allow users to run simulations from one unified platform for various scenarios such as simultaneous converging instrument approaches, land and hold short

operations, and parallel approaches. Because of the complexity of the possible scenario, the tool utilizes a Monte Carlo simulation (a method that estimates possible outcomes from a set of random variables by running simulated events a large number of times and observing the outcomes) based on a number of aircraft and airport-specific parameters. During simulations, initial values, such as surveillance system performance standards and runway configuration, are declared and remain constant through all trials. Other variables, such as the specific aircraft types and their initial starting position, are randomly selected from a known distribution for each trial run.

As an example of using the tool for airspace safety studies during aircraft parallel approaches, one aircraft may be assigned to act as the "blunderer" and is commanded to turn towards the other aircraft on approach. All blunders are considered worst case blunders (a 30° turn towards the traffic stream of another parallel runway). In order to model a worst-case scenario, the secondary aircraft are then positioned with a critical alignment that maximizes the likelihood of being intercepted by the blundering aircraft.

Actions normally taken by the aircraft autopilot system, human pilot, and air traffic controller are stochastically modeled using data response data collected in other studies to enable accurate results. The tool determines the distance between aircraft at each step in time while modeling a number of stochastic processes including initial position of the aircraft relative to other aircraft, system delay, navigation system error, controller response time, surveillance system error, and endangered pilot/aircraft response. If at any point an aircraft comes within 500 feet of another aircraft, it is considered a test criteria violation (TCV) and the simulation begins the next run in its course.

The software tool detailed here consists of more than 80,000 lines of code and has an estimated level of effort of over 20 person-years of development.

Aircraft Models

The simulation tool requires aircraft motion to be accurately modeled to determine if a TCV is likely to take place in a given scenario. With a variety of options available [1], the tool is designed with three aircraft type-specific behavior models with the generic procedure sequence and information flow shown in Figure 1.

Kinematic Aircraft Model

The kinematic flight model operates on the principle that a full six-degrees-of-freedom (6DOF) model may not be necessary to accurately determine aircraft location in the simulation if the aircraft is operating within its approved flight envelope. The model [2] operates primarily on speed and heading, and at each time steps it computes its position. The kinematic model utilizes Vincenty algorithms because of their proven accuracy, compatibility with the World Geodetic System 84 (WGS-84), and quick implementation and computation speed on the computer. This method calculates a geodesic path and uses that route to compute its next location (latitude and longitude; altitude is calculated independently). Aircraft performance characteristics are type-specific using data collected from pilots flying corresponding Level D flight simulators.

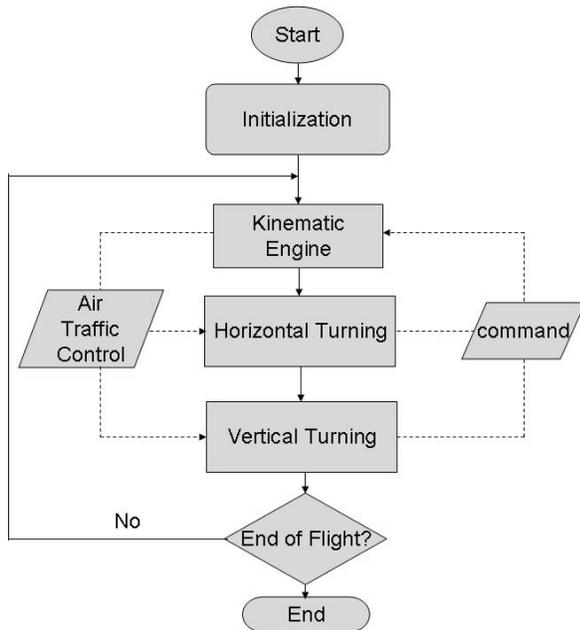


Figure 1. Aircraft Procedure Sequence and Information Flow

The kinematic equations are called at each time step to calculate the position, calling on inputs including the aircraft's current state, desired (commanded) state, and simulator clock time. The aircraft state includes its current latitude, longitude, altitude, angle of bank, heading, indicated airspeed, and several rates. It is assumed that each phase of flight (e.g., approach) maintains a constant speed through implicit thrust adjustments.

Kinetic/Kinematic Hybrid Aircraft Model

The kinematic flight model is able to attain a high level of realism, however this can be further enhanced by determining the vertical velocity and acceleration using EUROCONTROL's Base of Aircraft Data (BADA) model, an industry accepted kinetic-energy-based flight model. When combined with the kinematic model, this new hybrid model makes use of the BADA algorithm to determine the vertical velocity and gives results for determining vertical acceleration. These results were subsequently checked by comparing them with pre-existing test distributions and an additional third-party flight simulator. The hybrid model uses kinematic algorithms for all other functions and parameters. This hybrid model utilizes C code to access parameter data from BADA's collection of approximately 100 airplanes. Using an energy balance, the BADA model is used to determine vertical velocity during climb, cruise, and descent.

High-Fidelity, Six-Degrees-of-Freedom Aircraft Model

Finally, a high-fidelity aircraft model was desired for integration with the software tool. While capable of running in fast time (i.e., faster than real time and as directed by the simulation tool) its processing time is not as fast as that of the kinematic or the hybrid models. The high-fidelity aircraft model library consists of numerous commercial aircraft typical of most airport fleet mixes. The model is derived from the full 6DOF analysis of flight, with total forces and moments calculations performed at discrete points along the aircraft and is validated using certification flight test procedures [3].

Human Models

Various human factors (e.g., reaction times) were acquired from rigorous testing experienced pilots and air traffic controllers in simulated environments.

Pilot (Kinematic Aircraft Model)

The pilot actor is modeled using either a human or an automatic flight system. The pilot can evade, blunder, or maintain a heading. The pilot uses the information from the navigation aids to stay on an assigned heading. The model also maintains a glideslope to the touchdown point.

A blunder is modeled by that aircraft's pilot turning the aircraft to a fixed 30-degree heading change. The aircraft that the blunderer moves towards becomes the evader aircraft and, when commanded by the aircraft controller, executes a 90-degree turn away from the blunderer. Human pilot parameters and distributions along with those for an example heavy aircraft include

- Roll rate: 5°/s (3°/s for heavy or super aircraft)
- Angle of bank: bounded Gaussian distribution for all aircraft with $\sigma = 3^\circ$, $\mu = 17^\circ$, $\min = 15^\circ$, and $\max = 20^\circ$
- Rate of vertical speed (i.e., vertical acceleration in feet/minute/second) example given by Johnson S_B distribution with $\gamma = 0.225$, $\delta = 0.927$, $\lambda = 394.848$, $\varepsilon = 88.072$, $\min = 123.6$, and $\max = 462.6$
- Vertical speed (feet/minute) example given by Johnson S_B distribution with $\gamma = 2.592$, $\delta = 2.185$, $\lambda = 4526.482$, $\varepsilon = 1400.0$, $\min = 1442.8$, and $\max = 3108.9$
- Approach indicated airspeed (KIAS) example given by Johnson S_U distribution with $\gamma = -0.144$, $\delta = 0.887$, $\lambda = 1.06$, $\varepsilon = 142.0$, $\min = 137.2$, and $\max = 149.4$

Pilot (High-Fidelity Aircraft Model)

The high-fidelity pilot is significantly more involved, as it is modeled as a control system, with a proportional-integral-derivative (PID) loop. One prime example of the differences between kinetic and high-fidelity model pilot lies in adjusting to a desired speed. The kinetic pilot will achieve the

target speed (with random error) quickly and maintain it, whereas the high fidelity pilot will climb to the speed and adjust as necessary, using the aircraft instruments for feedback.

Air Traffic Controller

The air traffic controller monitors each airplane's position to determine if it is within 500 feet of the no-transgression zone (NTZ). If the plane ventures within 500 feet of the NTZ, the controllers direct other airplanes to execute an evasive maneuver (a 90° avoidance turn) away from the direction of the blundering aircraft.

In addition, the S_B Johnson distribution models the time required for an air traffic controller to respond to a blunder. A blunder is initiated when an aircraft starts to turn without instruction from an air traffic controller. If the time elapsed from the start of the blunder is equal to that determined by the Johnson distribution then the air traffic controller actor responds by directing the other airplanes to execute a 90° avoidance turn away from the blundering aircraft.

This particular scheme is modeled with a Johnson S_B distribution with a low truncation of 3.0 seconds, and a high truncation of 30.0 seconds.

Statistics

Along with the aircraft and human interactions (pilot and air traffic controller), the tool is reliant on realistic models of all other components involved with airport operations including the surveillance systems, navigation systems, and environmental conditions [4]. Understanding these models requires a short introduction to the statistical methods and tools used.

The mathematical modeling within tool takes advantage of uniform, Gaussian, Johnson S_B , Johnson S_L , Johnson S_U , and triangle distributions. All errors are modeled as randomly distributed according to given probability density distributions. Some procedures used to determine the error are numerical; however, some are simple enough to have analytical solutions. For example, the error for the uniform and triangle cases can be determined exactly. However, the Gaussian and Johnson S_B , S_L , and S_U distributions generally require numerical methods to determine the error.

The uniform distribution, used in surveillance and variables, uses a one-dimensional probability density function

$$P(x) = \frac{1}{\text{length}} \quad (1)$$

where the standard deviation can be given by

$$\sigma = \frac{\text{length}}{2} \frac{1}{\sqrt{3}}. \quad (2)$$

After normalization, the random number generator picks a fraction in the set [0, 1] as a multiplier to obtain

$$\text{error} = \text{fraction} \times \text{length}. \quad (3)$$

The Gaussian distribution is used to compute the error of a normally distributed variable. The probability density function is given by

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], \quad (4)$$

where μ is the mean and σ is the standard deviation. A numerical method is used to obtain values from the normally distributed variables.

Johnson S_B distribution is used to model the time for an air traffic controller to respond to a blunder. A Johnson S_L distribution is used to model the human pilot reaction time delay for receiving air traffic control directives, and the Johnson S_U distribution is used to select the indicated airspeed.

The triangle distribution is used to model variables where only approximate data is available.

Surveillance System Models

The surveillance systems are used to provide aircraft location within some error distribution. Since the simulation program is developed with the intent of being as generic as possible, a variety of the more common surveillance system and their associated errors are included.

ASR-9/ASR-11

Airport Surveillance Radar model 9 (ASR-9) short range radar and ASR-11 long range radar are used to track the position of the aircraft and provide the locations to the air traffic controller actor. The surveillance system parameter values are based on radar specifications and field data for radars in Southern California and the Northeast collected by the Lincoln Laboratory at the Massachusetts Institute of Technology. There are five different types of sensor-error types: registration, range, azimuth, data dissemination quantization, and uncorrelated scan time.

Registration is classified as a bias (and maintained as a constant error the entirety of a run), while azimuth errors are classified as jitter (electronic noise, modeled with normal distribution and updated for each time step). Range errors can be both bias and jitter. Data dissemination quantization error is updated at the start of each run and is accounted for by rounding the aircraft position to the radar's nearest resolvable range and azimuth position. The uncorrelated sensor scan-time error accounts for the mechanical sweep of the radar (since the actual radar does not image the entire 360° simultaneously). Secondary surveillance radar errors are added to the total range error.

For example, Figure 2 illustrates a possible scenario for the difference between actual and sensed aircraft location due to the radar azimuth error ϕ (range errors are not shown in this example for clarity).

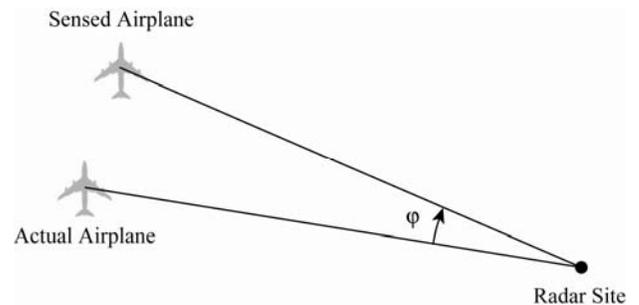


Figure 2. Radar Error Model Example

ADS-B

Automatic Dependent Surveillance – Broadcast (ADS-B) is a surveillance system offering more precision and additional information

(such as weather and traffic information). Aircraft receive GPS signals and use them to determine their ownship position. This is then combined with other data and broadcast to other aircraft and to air traffic control.

ADS-B is modeled using a Gaussian distribution with the characteristics shown in the Tables 1 and 2 (note some of the data is given in meters m and some in nautical miles NM; some of the original data is also given in kilometers). The 95% bound r_{95} and the standard deviation (St. Dev.) of the estimated position uncertainty (EPU) σ_{EPU} are related by

$$\sigma_{EPU} = \frac{r_{95}}{\sqrt{2|\ln(0.05)|}} \quad (5)$$

Table 1. Characteristics of ADS-B

2-D Gaussian Distribution	
NAC _p	11 to 1
Mean	$\sigma_{EPU} \cdot \sqrt{2} \cdot \sqrt{\pi}$
Standard deviation	σ_{EPU} from Table 2
Bounds	95% from Table 2
Bounds	None
1-D Uniform Distribution	
Mean	180 degrees
Bounds	0 and 360 degrees
Time Delay	
Time delay	3.2 seconds

Table 2. Bound and Standard Deviation Values for Each Navigation Accuracy Category for Position (NAC_p)

NAC _p	95% bound (r_{95})	St. Dev. (σ_{EPU})
11	3 m	1.23 m
10	10 m	4.09 m
9	30 m	12.3 m
8	0.05 NM	0.0204 NM
7	0.1 NM	0.0409 NM
6	0.3 NM	0.123 NM
5	0.5 NM	0.204 NM

4	1 NM	0.409 NM
3	2 NM	0.817 NM
2	4 NM	1.63 NM
1	10 NM	4.09 NM

PRM

Precision Runway Monitor (PRM) is a precise, high-speed radar system developed to monitor simultaneous parallel instrument approaches for runways less than 1,525 m from each other.

PRM is modeled as shown in Table 3.

Table 3. PRM Characteristics

1-D Gaussian Distribution	
Mean	0
Standard Deviation	$\sigma = 50$ ft
Bounds	$\pm 2\sigma$
1-D Uniform Distribution	
Mean	180 degrees
Bounds	0 and 360 degrees
Time Delay	
Update Rate (scan time)	1 sec
Processing Delay	0.5 sec
Display Delay	0 sec
Detection	
Detection Probability	99%

Navigation Systems

Navigational aids (navaids) broadcast a signal that enables the aircraft avionics to determine the position of the aircraft relative to the navaid. In the tool, this information is then combined with the known navaid location to obtain the aircraft location.

ILS

Instrument Landing System (ILS) errors are modeled using the International Civil Aviation Organization (ICAO) collision risk model (CRM) modified with a correction factor. Initial lateral and vertical position errors are determined using the CRM, and then the aircraft model is directed down

along a calculated glideslope and towards the runway touchdown zone. A correction factor CF is applied to the lateral error according to

$$CF = \frac{1 + d/a}{1 + d/b} \quad (6)$$

where d is the distance of airplane from threshold, a is the localizer to the threshold distance, and b is 3,800 m. The lateral error is first determined from the CRM, and then multiplied by CF to obtain the modified lateral error.

VOR

The VHF Omni-Directional Range (VOR) system broadcasts a signal that the airplane uses to determine its bearing from the VOR station. The VOR model positions the airplane using a one-dimensional continuous probability distribution applied to the original airplane position in a horizontal plane. This VOR angular error is modeled using a Gaussian distribution with a mean of zero, standard deviation of 1.5 degrees, and bounds at ± 4.5 degrees.

Figure 3 shows an example of the VOR error model. The initial airplane position and the final airplane position (after repositioning using the model error) are shown. The angle θ corresponds to the $+4.5$ degree bound and the angle ϕ corresponds to a random angular error variant of a few degrees as determined from the Gaussian distribution. Note that all final airplane positions will lie within the angles $+\theta$ and $-\theta$. Given the distance from the airplane to VOR to be R , the lateral displacement error is $d = \phi \cdot R$.

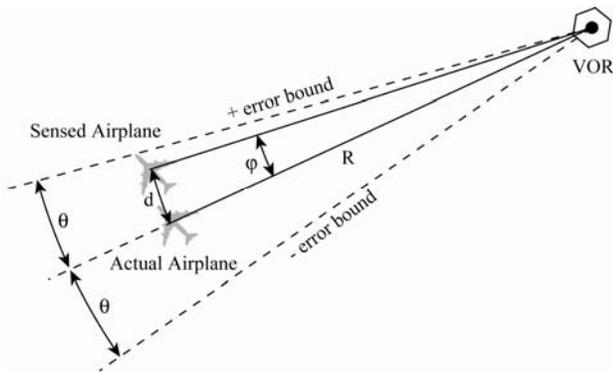


Figure 3. VOR diagram

DME/DME

Distance Measuring Equipment (DME) is a radio navigation system that measures distance by timing the propagation delay of VHF or UHF signals. As a navigation system, DME/DME uses the slant ranges between the airplane and two DMEs to compute the location of the airplane.

The DME/DME error model works as illustrated in Figure 4.

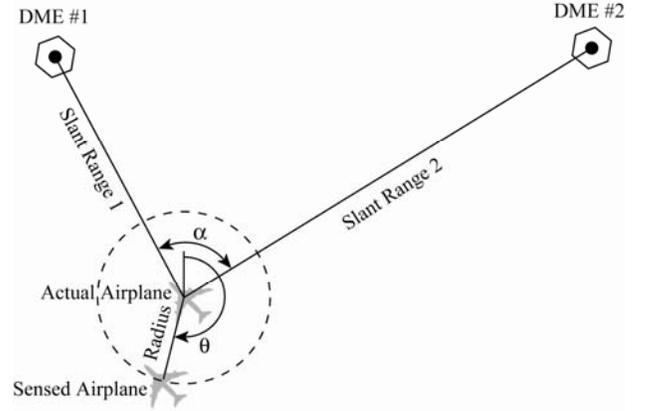


Figure 4. DME diagram

As shown, the DME/DME model repositions the aircraft using two one-dimensional distribution-based errors (radius and angle) in the horizontal plane. This is modeled with a one-dimensional Gaussian distribution for radial error with the standard deviation given by

$$\sigma_{DME/DME} = \frac{\sqrt{\sigma_{1,air}^2 + \sigma_{1,sys}^2 + \sigma_{2,air}^2 + \sigma_{2,sys}^2}}{\sin(\alpha)} \quad (7)$$

where $\sigma_{1,sys} = 0.05$ NM and $\sigma_{2,sys} = 0.05$ NM. $\sigma_{1,air}$ is computed by finding the greater of 0.085 or 0.125% times the slant range distance between the first DME and the airplane; $\sigma_{2,air}$ can be found in a similar manner. α is the angle between two slant range lines. For DME/DME, the airplane is repositioned randomly through the use of a uniform distribution on a circle whose radius is the absolute value of the Gaussian random number with zero mean and standard deviation $\sigma_{DME/DME}$.

LAAS/WAAS

The Local Area Augmentation System/Wide Area Augmentation System (LAAS/WAAS) makes corrections to the GPS signal and is capable of achieving an accuracy of ± 6 m.

The error model used calculates the stochastic lateral, longitudinal, and vertical errors. The vertical error is a Gaussian distribution with zero mean, standard deviation of 2.25 m, and bounds of ± 6 m. The vertical error variant is added to the airplane altitude. The lateral and longitudinal errors are also Gaussian distributions, with a zero mean, standard deviation of 1.5 m, and bounds of ± 4 m.

RNP

The Required Navigation Performance (RNP) system is modeled with a single Gaussian distribution that is used for the lateral and longitudinal errors. Also, the longitudinal error is only used to produce an equivalent vertical displacement and is not used to produce a longitudinal error in the horizontal plane. In addition, the ILS ICAO CRM vertical error is added in order to account for the pilot portion of the flight technical error. The lateral and longitudinal errors are modeled using the Gaussian distribution with a mean of zero, standard deviation of σ_{RNP} (from 0.060 NM to 0.170 NM depending on RNP level) and bounds of $\pm 3 \sigma$.

GPS/RNAV

The Global Positioning System/Area Navigation (GPS/RNAV) model positions the aircraft according to the parameters in Table 4 using one distribution-based error (lateral) applied to the original aircraft position. It also models the position with one distribution-based error (longitudinal) applied to the original position in the horizontal plane. The longitudinal error is applied to the vertical. After re-positioning, the aircraft heading is directed on the approach course.

Table 4. GPS/RNAV Characteristics

Lateral/longitudinal distribution type	single Gaussian (the same value is used twice)
Mean	zero nautical miles
Level	0.07 nautical miles
Standard deviation	0.039888 nautical miles
Bounds	$\pm 3 \sigma$ nautical miles

Environmental Models

Coordinate System

The simulator uses WGS-84 to describe the aircraft's position. Altitude is reported in feet at mean sea level with positive values above sea level and negative values below. All calculations involving distances along the curved earth surface are carried out in nautical miles. For long flight distances, round-earth considerations help to provide a much higher degree of accuracy when compared to flat earth models.

Standard Atmosphere

The atmospheric pressure is modeled as dependent on the vertical (z) direction with the equation

$$\frac{dp}{dz} = -\rho g \quad (8)$$

through the manipulation of Newton's second law applied to an inviscid fluid with density ρ and gravity assumed constant.

The model for temperature, in degrees Celsius ($^{\circ}\text{C}$) stems from the ICAO-defined standard atmosphere with a temperature lapse rate of $6.5^{\circ}\text{C}/\text{km}$ ($1.99^{\circ}\text{C}/1,000$ ft) from sea level to 11 km (approximately 36,000 ft). At this altitude, a linear temperature dependence can be defined as

$$T = T_0 - Bz \quad (9)$$

where B is the temperature lapse rate. Assuming

$$\rho = \frac{P}{RT} \quad (10)$$

with the previously given temperature dependence, the pressure equation becomes

$$P = P_a \left(1 - \frac{Bz}{T_0} \right)^{g/_{RB}} \quad (11)$$

where P_a is the pressure at sea level. Note that $g/_{RB}$ becomes 5.46 and is dimensionless. It is also assumed that isothermal and isobaric lines are tangent to the earth's surface everywhere.

Wind

A simple two-dimensional wind, independent of time, is included in the simulation. It consists of a wind speed and direction constant at all altitudes that is tangent to the ground. The wind is resolved into its north and east directional components at all altitudes

Software Engineering

The software design enabled a fully documented computer program that can be easily operated by an end-user with little programming knowledge.

Software Design and Architecture

An object-oriented approach to design and coding was chosen for this application. Object-oriented programming is composed of self-service *objects* containing all the information needed to manipulate their own structure. Each object is capable of acting independently and interacting with objects. The object-oriented design allows for rapidly integrating new objects (e.g., adding a new navigation system to the navigation class) since it will inherit the properties of the governing class.

Development Tools

The programming language was chosen to be C++ because of its performance and familiarity within the programming community and due to its ability to interface with external mathematical libraries and packages. Specifically, the tool is written with GNU GCC (an open source compiler collection) and documented with Doxygen (an open source documentation generator for C++ and other languages). The open source, cross-platform

graphical user interface (GUI) library, wxWidgets, was used for all GUI work.

Simulator Time Step

The kinematic aircraft's physics model equations are independent with respect to absolute simulator time. This means that at each time-step of the simulator, the model equations are called from the beginning and run through completely. It is possible, therefore, to vary the time-step between calls. The main simulation program sends the kinematic flight model the current simulator time. The time between the previous and current time is calculated (this is, in fact, the only number necessary). This current time-step helps determine the change in position (e.g., change in latitude and longitude) and hence the aircraft's current position. The default time-step is currently 50 milliseconds, so an aircraft with an approach speed of 122 kias (such as a twin-turboprop) yields an approximate position step of ten feet. Similarly, an aircraft with an approach speed of 150 kias (such as a medium jet) yields an approximate step of thirteen feet.

Graphical User Interface

In the interest of minimizing the training required for a casual user to run simulations with the program, a GUI has been implemented since the intent is for the software to be relatively easy to use by a person who is not a programmer. To accomplish this, the software has been designed with the capability of being run entirely from a GUI. The intuitive GUI allows the user to quickly set up and initiate a run of simulations without programming or compiling.

Once the data parameters have been entered in the extensible markup language- (XML) based scenario file, the results of any runs can be displayed in the GUI (see Figure 5) that allows the user to track a single trial and monitor the aircraft's movement in latitude and longitude compared to the runways and altitude over time.

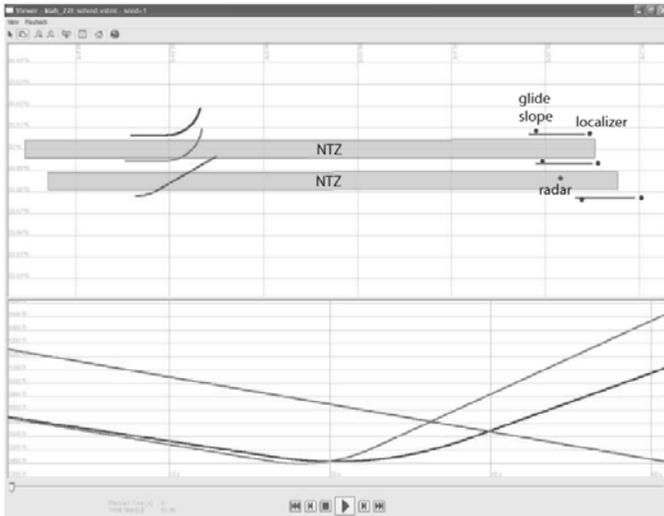


Figure 5. GUI Screenshot

The raw numerical data is also collected and written to disk file. In addition to the graphical tracking seen in Figure 5, there is a capability to overlay rendered motion tracks in 3-D on to real world images using an automatically generated kml file.



Figure 6. KML File Animation Screenshot

Software Testing

The process of software testing is to make sure that the tool meets all technical requirements, works as expected, and can be implemented repeatedly with the same characteristics. This process is broken primarily into verification and validation tests.

Since it is crucial that the simulation software perform at the highest standards, each segment of the program was first thoroughly tested as a module and then after integration to ensure its veracity. The software testing was performed by personnel certified in testing, while the associated software documentation allows for third-party testing.

Software Verification

The verification process checks to ensure the tool produces results consistent with itself (e.g., symmetric runways produce similar results) and to test this, a battery of trajectory, hypothesis, and closest distance tests were performed [5].

The symmetric runway geometry is created by modifying the locations of runways, localizers and glide slopes, and radar in the scenario files. In the symmetric scenario file, the latitudes of the runway ends are made to be the same so that the runways are placed on lines of latitude. The longitudes for the east and west ends of the runways are also equalized, respectively, so that the runways are aligned at the same line of longitude. In addition, localizer and glideslope antennas, and radar sites are purposely placed and aligned at runway centerlines. With the symmetric runway geometry and same initial conditions of the aircraft, the output trajectories and the TCV rates for the east and west bound simulation should be equal.

To validate the consistency of the tool with random parameters, including aircraft type, aircraft initial position, maximum bank angle, vertical speed, vertical acceleration, approach air speed, and radar errors, TCV rates for the east- and west-bound scenarios were examined by Monte Carlo simulations with the geometry of symmetric runways. Monte Carlo simulations were run four times with 50,000 runs for each scenario with different random seeds. The hypothesis test with the Student's *t*-distribution (or Student's *t*-test) was used to determine if the results are the same statistically.

The results of the Student's *t*-test show that with six degrees for freedom, the critical value of the Student's *t*-test is $t_{cv} = 2.4470$. The test statistic " t^* " is calculated by using the simulated TCVs. The test statistic is almost at the center of the critical values, indicating that the TCVs of the 1E scenario are statistically the same as the TCVs of

the 1W scenario at the confidence level of 0.025. The conclusion is then reached that the program with random parameters is self-consistent for the simulations of the east- and west-bound scenarios.

Model Validation

In the validation process, the results generated by the tool are compared to those produced in a previously conducted study [6] to check for consistency between the two programs.

The first method compared the TCV rates from both programs [5]. TCVs were generated by using the tool for 50,000 runs per scenario with evading. To compare the TCV rates to the previous study, the blunder “at-risk” TCV rate of 90 percent was used as a divider for scaling the data as was done for the benchmark results [6]. Figure 7 (a) – (b) shows the comparison between the TCV rates for all scenarios generated by the tool and the benchmark, where each scenario consists of 50,000 simulation runs (a run is defined as a simulation with parameters what are randomly set). Some of the input parameters were not available in the earlier document and have been derived from third-party sources. Overall, good global agreement has been achieved for the TCV rates between the tool and the benchmark; however, local discrepancies between scenarios are noticeable, particularly for the West Right-Center blunders. The discrepancies could be due to the differences between the models and input parameters. Possible reasons for the mismatch may include

- the use of a models the earth as an ellipsoid of revolution according to the WGS-84 model rather than a flat-earth model
- some subtle input parameters may not be the same
- some functions, algorithms, or logic may be modeled differently (e.g., radar sweep vs. random distribution; etc.)
- normal statistical variation could be expected in the benchmark results

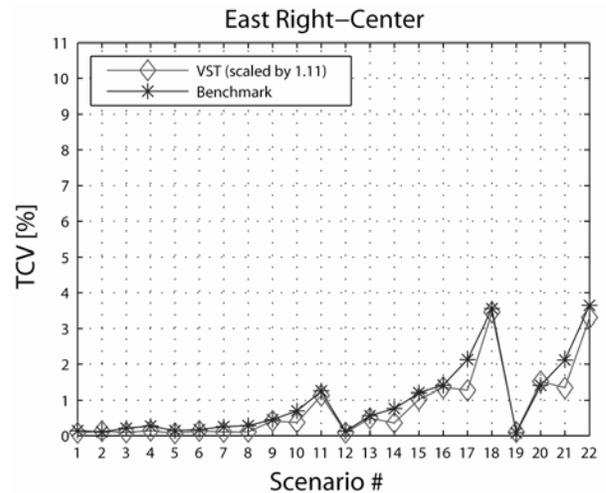


Figure 7 (a). TCV Rates for the East Right-Center Scenario

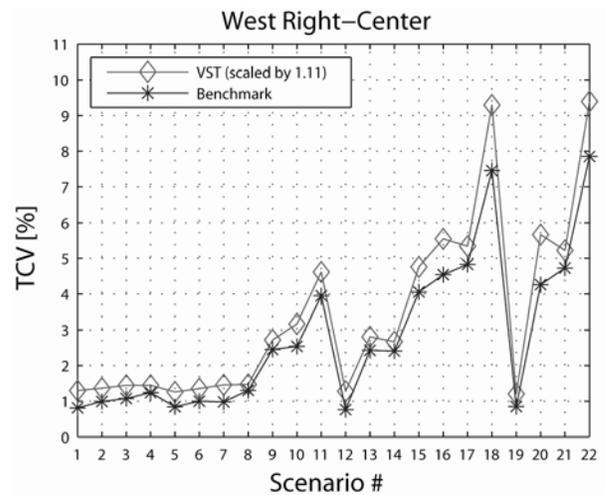


Figure 7 (b). TCV Rates for the West Right-Center Scenario

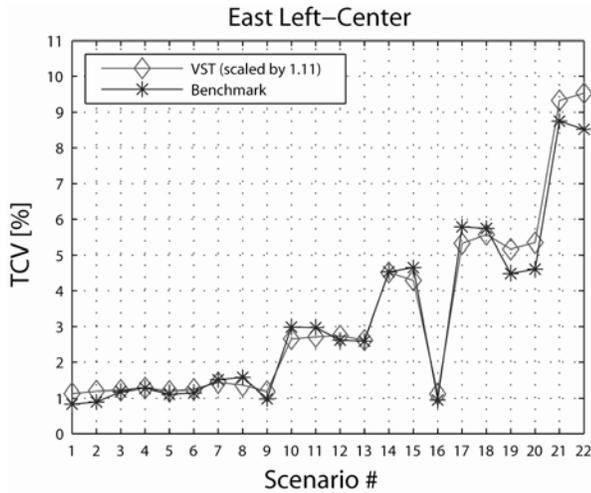


Figure 6 (c). TCV Rates for the East Left-Center Scenario

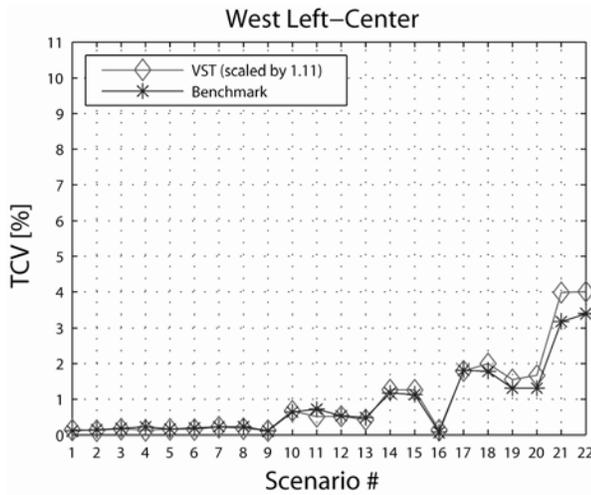


Figure 7 (d). TCV Rates for the West Left-Center Scenario

The second validation tool is the Pearson product-moment correlation coefficient (PMCC) [5], which is a common measure of the correlation between two variables, where 0.90-1.00 is considered “very highly correlated.” Table 5 shows the computed Pearson PMCC values between the tool and the benchmark data. The limitation of this validation is that it is not sensitive to a constant offset since the mean value is removed from the data.

Table 5. Pearson PMCC Values

	Right-Center Blunder	Left-Center Blunder
East Bound	0.9833	0.9904
West Bound	0.9706	0.9945

The Pearson PMCC value is determined with the following equation:

$$r = \frac{\sum_{s=1}^S [(X_s - \bar{X})(Y_s - \bar{Y})]}{\sqrt{\sum_{s=1}^S (X_s - \bar{X})^2} \sqrt{\sum_{s=1}^S (Y_s - \bar{Y})^2}} \quad (12)$$

where \bar{X} and \bar{Y} are the mean values and S is the size of the data set.

Summary

With modeling and simulation taking a larger and larger role of evaluating situations, procedures, and technology, this paper reviewed some efforts in modeling various interacting components in the national airspace system. Specifically, an overview was presented detailing a software tool and its formulation, including programming languages, environments, and other development tools; selection, design, and implementation of various mathematical models; and the verification and validation methodology.

References

- [1] McGovern, S. M. and S. B. Cohen, 2007, “Survey of Contemporary Aircraft Flight Dynamics Models for Use in Airspace Simulation,” Proceedings of the 2007 SPIE Defense and Security Symposium, International Conference on Modeling and Simulation for Military Operations II, April 10-12, 6564, Orlando, FL, SPIE, CD-ROM.
- [2] McGovern, S. M., S. B. Cohen, M. Truong, and G. Fairley, 2007, “Kinematics-Based Model for Stochastic Simulation of Aircraft Operating in the National Airspace System,” Proceedings of the IEEE/AIAA 26th Digital Avionics Systems Conference, October 21-25, Dallas, TX, IEEE/AIAA, CD-ROM.
- [3] McGovern, S. M., 2007, “Categories for Classification of Aircraft Flight Model

Validation,” Proceedings of the IEEE/AIAA
26th Digital Avionics Systems Conference,
October 21-25, Dallas, TX, IEEE/AIAA, CD-
ROM.

- [4] Fairley, G., 2007, Stochastic Simulation
Physical and Statistical Models, DOT-RITA-
RVT-4B-07-05-1.1, Cambridge, Massachusetts.
- [5] Zhang, Y., 2009, Stochastic Simulation
Validation Results, DOT-RITA-RVT-72-09-05-
1.1, Cambridge, Massachusetts.
- [6] McCartor, G. R. and S. Ladecky, 2005, Safety
Study Report on Triple Simultaneous Parallel
Instrument Landing System and Area
Navigation/ Required Navigation Performance
Approaches at George Bush Intercontinental
Airport, DOT-FAA-AFS-440-16, Oklahoma
City, OK, FAA.

Acknowledgements

This material is based on work supported
under RITA/VNTSC #FA7GB1 FD4RN.

Disclaimer

Any opinions, findings, and conclusions or
recommendations expressed in this paper are those
of the author.

*28th Digital Avionics Systems Conference
October 25-29, 2009*