

# ADSIM+: A NEW DEVELOPMENT OF THE AIRFIELD DELAY SIMULATION MODEL

**Panta Lucic\*, Donghai He\*, Louai Adhami\*, Joseph Post\*\***

\*CSSI, Inc., 400 Virginia Avenue, SW, Suite 210, Washington, DC

\*\*Federal Aviation Administration, 800 Independence Ave., SW, Washington, DC

**Keywords:** *Airport, Terminal Area, Fast-Time Simulation, Capacity, Delay*

## Abstract

*This paper introduces ADSIM+; a fast-time airfield simulation modeling tool. ADSIM+ is an extension of the FAA's ADSIM simulation model that has been in use since the 1980s. ADSIM+ aims to enhance the ADSIM's airfield modeling characteristics and to extend them to include modeling of airport terminal area operations. ADSIM+ is designed to be a flexible tool aimed at modeling current terminal area/airport operational concepts as well as the changes proposed by NextGen.*

## 1 Introduction

Airport models are designed to address a large set of what-if questions stemming from the development and/or expansion of airports. In the design phase, decision makers need to consider a large set of airport related attributes and their interactions and evaluate numerous airport performance measures such as delays, capacity, safety, environmental impacts, and the cost-effectiveness of the proposed solutions. These attributes include airport elements (e.g., gates, taxiways, and runways), their methods of use (e.g., airport configurations and associated driving factors) and the airport users and their operations. In many instances these attributes are not independent (e.g., Metroplex systems).

Airport modeling and optimization, as a part of overall airport performance analysis, has been researched extensively. While analytical airside models (well known ones include the FAA Airfield Capacity Model [1] and the LMI Runway Capacity Model [1]) are designed to support strategic decisions, the simulation based airside models (well known ones include

ADSIM [2], RAMS [3][4], RDSIM [1], *runway Simulator (rS)* [5], SIMMOD [6], and TAAM [7]) are used to support airport design and operationally related decisions. Traditionally, analytical models are limited to a specific set of scenarios, and simulations are frequently costly to setup. Moreover, with the exception of the recently built rS, these models address either delay or capacity related questions. Since the complexity of the questions being addressed by airport models frequently lead to intractable detailed analytical solutions, using simulations to evaluate airport projects seems natural.

Many of the previously introduced models have a long history and have been modernized through the use of more flexible software languages. This however does not fully apply to ADSIM, which has started to be modernized only recently. This paper outlines the current effort to update ADSIM. The new model, ADSIM+ is envisioned to be an application capable of modeling current and future (e.g., NextGen) concepts of both airfield and terminal operations.

ADSIM+ focuses on the operations of aircraft inside the airport terminal area (including the airport surface) and has two modes of operations: delay and capacity. In the delay mode, the user defines the scope of the simulation (runways only, airfield, and/or the terminal area), the airport layout, schedule, airport runway configurations and the aircraft separation rules. In the capacity mode, the user specifies a fleet mix instead of a schedule. While the delay mode produces detailed statistics for various entities involved in the simulation, the capacity mode produces a capacity envelope.

## 1.1 Background

The Airfield Delay Simulation Model (ADSIM) [2] has been developed as the FAA's in-house tool for detailed modeling of airfield operations for over 30 years. ADSIM is a discrete event (Monte-Carlo) simulation of airport operations. ADSIM has been in use for a long period of time and thus has been validated extensively.

ADSIM and its sibling, RDSIM [1] divide airport and terminal airspace modeling as follows: RDSIM's focus is from the immediate airport vicinity (only a portion of terminal airspace) down to the runways; while ADSIM's focus is wider and additionally includes surface movements.

Airport layouts can be digitized for input into the model as a link-and-node structure, additionally complex runway relationships can be defined. ADSIM currently can model up to 8 runways.

ADSIM supports "What if" analyses on the impact of new runways, taxiways, high speed exits, new gate configurations, runway closure analysis, and introduction of new large aircraft (e.g., A380) at airports. The model calculates travel time, delay, and flow rate data to analyze components of an airport, airport operations, and operations in the adjacent airspace.

ADSIM outputs include: hourly (or quarter-hourly) delays caused by each user-specified ATC separation; arrival and departure delays by hour (or quarter-hour), by aircraft class, by runway, by airfield components, and by airfield operations including runway crossing and taxiway queues; hourly (or quarter-hourly) flow rates and travel times. These metrics can be aggregated by type of operation or time period.

The learning curve of ADSIM could be considered moderate; prior experience with other simulation models is an advantage. ADSIM is actively maintained by the FAA in-house; enhancements to the model are made on an as needed basis. ADSIM is free for FAA users and no special licenses are required.

ADSIM currently does not have a Graphical User Interface (GUI) and data input is performed through direct editing of input (text) files. The output consists of a set of output (text)

reports. Additionally, ADSIM has a limited animation capability and a limited interactive setup and error diagnostics/debugging capability. It is written in a programming language that does not allow for quick software augmentations.

To address these limitations a major overhaul of the ADSIM model was undertaken, this included re-architecting ADSIM into a modern software language (i.e., C++), implementing a GUI (using Qt) and enhancing its error diagnostics/debugging capability. These changes are being implemented in a manner that not only preserves the original ADSIM capabilities, but also expand its modeling capabilities to include modeling of NextGen concepts throughout the terminal area.

## 1.2 Objective

The FAA is modernizing the Airfield Delay Simulation Model (ADSIM). To date, ADSIM is the FAA's best in-house tool used to model airfield operations. Nevertheless, ADSIM is written in FORTRAN 66 and thus is hard to maintain and allows for limited expansion of its capabilities. The main goal of this effort is to port the existing application into a modern programming language (e.g., C++ and/or Java) in a way that will allow:

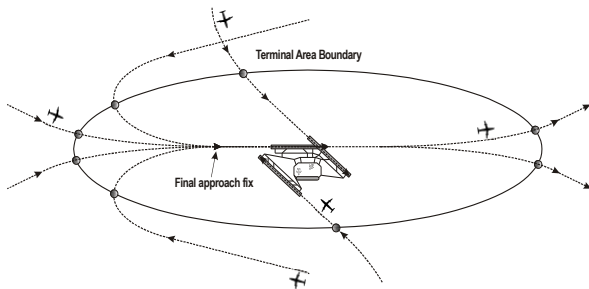
- Easier application maintenance;
- Shorter development time for the modification and expansion of its capabilities.

The new (Advanced) Airfield Delay Simulation Model (ADSIM+) will provide sufficient flexibility such that it could model advanced concepts of airfield operations and will have wider focus; it will be able to model terminal area operations. Additionally, in contrast to ADSIM, which runs only under the Linux operating system, ADSIM+ is being developed as a cross-platform application.

## 2 ADSIM+ Concept of Use

ADSIM+ will focus on the operations of aircraft inside the airport terminal area, which could be roughly described as a three-dimensional cylindrical region of space with a radius of

approximately 50 nautical miles surrounding the airport surface (see Fig. 1).



**Fig. 1. Airport terminal area**

When arriving at an airport, an aircraft passes an entry point of the airport terminal area. While approaching a runway, the aircraft passes a few distinct points: an outer marker (a final approach fix), a middle marker and an inner marker. Once on the ground, the aircraft decelerates and takes an appropriate exit (possibly a high-speed exit if one exists) and taxis to a gate. Aircraft parked at gates are serviced (e.g., unload and load passengers/freight, refueling). Once service is completed, the aircraft is ready to taxi out along a designated taxipath to the assigned departure runway. The departure is completed when the aircraft takes off from the runway and reaches the airport terminal area boundary. This is a simplified description of a landing-and-takeoff (LTO) cycle. The aircraft operations scheduled for an airport could be provided as a chronologically ordered list of LTO cycles.

Because the simulation could be bounded by time ( $t_{begin}$ ,  $t_{end}$ ), an initial state of the simulation could be provided by the user to identify the state of the simulation at the time  $t_{begin}$ . Otherwise, it would be assumed that at time  $t_{begin}$  the system is empty. The initial state could be provided with a schedule that includes only departures for some aircraft which implies that the aircraft are already at the airport at  $t_{begin}$ . Furthermore, touch-and-go (TGO) operations can be modeled.

For each operation, the schedule includes properties of the flight (e.g., arrival time, preferred runway) and a designator for the aircraft category, which depicts properties of the aircraft during flight. ADSIM+ will model aircraft movements as realistically as possible

by taking into consideration properties of corresponding aircraft categories specified by the user. The following two options were considered for modeling of the terminal area movements:

- Arrival/departure route (3D) will be defined as the user's input (terminal area route definition) and speed will be provided for each aircraft category through a set of altitude dependent speed bands;
- Full implementation of an aircraft performance model.

Modeling of aircraft performance based solely on the requirements specified in the arrival/departure procedure definitions and the user input in the aircraft category definition is relatively simplistic and easy to implement. However, this option may require the user to input more data than minimally necessary, is a relatively crude approximation of reality, and limits the simulation outcome to only basic metrics (e.g., arrival/departure delay).

An aircraft performance model (e.g., BADA [8]) on the other hand will reduce the user's input. The user will need only to associate an aircraft type to each flight. This allows more realistic modeling of terminal area operations (e.g., takes into consideration an aircraft altitude to adjust climb/descent rates and speed), which provides more simulation capabilities (e.g., verification of some of the aspects of flyability of a terminal area route), and will accommodate the assessment of more performance indicators (e.g., fuel burn). However, this model is more difficult to implement and would require ADSIM+ to have a set of "system" data (e.g., BADA could be part of it). Further, it may require licensing.

ADSIM+ incorporates several methods to maintain realistic separation between aircraft. Wake vortex separation is supported in addition to ADSIM's legacy separation methods which were implemented to maintain the traffic separation required by dependent (parallel or crossing) runways. This includes the ability to model crossing yet independent runways (Land and Hold Short [LAHSO]-type operations).

ADSIM+ also supports the modeling of ground operations occurring at gates, taxiways and runways.

ADSIM+ will allow the user to input the airport layout as well as the terminal area trajectories. The user will be able to use the following means to input an airport layout:

- GUI via a set of dialogs and drawing tools that could import airport drawings and allow the user either to identify different components of the airport or to trace over them
- A specific XML format
- Currently, we plan to support data imports from:
  - An Aeronautical Information Exchange Model (AIXM) formatted text file;
  - The NACO database (terminal area routes).

ADSIM+ will allow the user to input various levels of detail regarding airport usage such as:

- Specifying for some or all of the LTO cycles the entry point into the terminal area, the arrival runway, the runway exit, the taxipath to the gate, the arrival gate, the departure gate and the taxipath from the arrival gate to departure gate (if necessary), the taxipath to departure runway, the departure runway, and the exit point from the terminal area;
- Providing only some of the data listed above per LTO cycle and allowing ADSIM+ to determine the remainder (e.g., let ADSIM+ select the gate for an operation based on airline and aircraft category, choose the runway based on the user's defined distribution, or use a runway exit based on aircraft performance);
- Using airport runway configurations which could be activated based on a predefined schedule (a time-driven change) and/or changed based on *local weather*<sup>1</sup> conditions.

The user's ability to provide some of the input data in an aggregate sense exists in the current ADSIM and will be preserved in ADSIM+ (e.g., the runway exit selection). ADSIM+ will allow the user to specify additional aggregated inputs (e.g., runway assignment).

ADSIM+ will have an entirely new layer of aircraft separation functionality to maintain separation between aircraft in the terminal area. Given that it's main purpose will be to maintain the separation of airborne aircraft, this layer could be quite different from the current functionality. There will be three possible options to maintain separation between aircraft within the terminal area: speed control, path stretches (vectors, fanning, trombone routes), and holding patterns. Modeling aircraft movements in the terminal area of an airport could increase the amount of data that the user is required to provide to run the model (e.g., the user will need to define paths from all entry points of a terminal area to the runway thresholds for the arrivals and paths from the runways to the exit points for the departures, to define the holding patterns, and/or the path-stretch logic).

ADSIM+ is envisioned to be flexible enough to allow the user to run the simulation utilizing only the data available (e.g., if the user does not provide data on the terminal area routes, ADSIM+ will consider the airfield only). The user could provide a definition of only the runways, the airfield, or the entire terminal area to run the simulation.

ADSIM+ has two running modes:

1. Airport simulation (delay mode);
2. Airport runway capacity analysis (capacity mode).

In the airport simulation mode, ADSIM+ will simulate movement of aircraft so that the statistical properties, such as delays, queuing length of runways and average waiting time can be estimated. The entire airfield is modeled as a First In First Out (FIFO) queuing network model with blocking. A simple car-following model is developed to ensure more realistic ground movements of aircraft are simulated. To run the simulation mode, the user will provide a schedule, terminal area properties, airport

---

<sup>1</sup> Weather driven change in the runway configurations requires development of a capability to read in and process a specific segment of the weather data

## ADSIM+: A NEW DEVELOPMENT OF THE AIRFIELD DELAY SIMULATION MODEL

layouts, fleet (aircraft categories) properties, separation rules, and airport runway configurations.

In the capacity mode, ADSIM+ simulates aircraft arrivals and departures while taking into consideration only airport runways. To run the capacity mode, the user will need to specify a fleet mix. When operating in this mode, ADSIM+ determines the airport runway capacity only, which means that no consideration will be given to the taxiways, gates and/or terminal area routes.

ADSIM+ preserves the simulation randomization capabilities of ADSIM. Additionally, it will allow the user to randomize:

- A set of parameters that may influence aircraft performance (e.g., speeds);
- The spacing between arrivals and departures;
- Departure runway entry points.

As mentioned earlier, ADSIM+ has a GUI to let the user set up a study, review results (reports), and review a visualization of the simulation (which could be 2D or 3D interactive “show-as-you-go” animation or playback).

ADSIM+ saves interim simulation states to allow for the post-simulation calculation of various statistics, generating reports and/or animating runs. ADSIM+ will let the user validate input data before starting a simulation run (e.g., verify connectivity between taxi-segments along a taxipath and the flyability of the terminal area routes).

### 3 Development Environment Evaluation

This section outlines evaluation of development environments for the ADSIM+ simulation model. The following properties were considered:

- Reliability
- Efficiency
- Flexibility.

Reliability is the most critical requirement of the development platform. A reliable platform can significantly reduce the time for debugging and the maintenance cost of

ADSIM+ in the future. A reliable platform has to be well tested and professionally maintained.

Efficiency is another major concern in ADSIM+. ADSIM+ will be a stochastic discrete-event simulation model. Therefore, all outputs and analysis will be based on Monte Carlo simulation. The precision of Monte Carlo simulation strongly depends on the number of simulation replications, whereas it is expected that the variance of output results decreases as the number of simulation replications increases. Therefore, a statistically significant output requires a sufficiently large number of simulation replications. To conduct a statistical analysis as accurately as possible, the ADSIM+ model has to be as efficient as possible. The efficiency of a simulation model is strongly correlated to two aspects: the system design and its implementation. Furthermore, careful selection of the development language can significantly improve the efficiency of the final simulation model. C++ is the most efficient modern computer language, which is normally faster than Java. Therefore, C++ is the top choice in terms of efficiency.

Flexibility of the development environment is also very important for ADSIM+. Since one of the requirements calls for ADSIM+ to be a cross-platform application (at least Windows and Unix/Linux need to be supported), the development environment has to have the ability to support both of these platforms. ADSIM+ should be able to be ported to other platforms either directly or by recompiling the original source code, with trivial modification. Java is a well-known cross-platform language, whereas a GUI developed in C++ is usually highly platform dependent. However, there are several excellent cross-platform C++ libraries that allow a platform-independent GUI to be written in C++, such as wxWidgets<sup>2</sup> and Qt<sup>3</sup>. To meet these requirements, the following environments were considered:

- Java (for both GUI and Kernel)
- Java and C++ (Java for GUI and standard C++ for Kernel)
- Cross-Platform C++ GUI Library

---

<sup>2</sup> <http://www.wxwidgets.org/>

<sup>3</sup> <http://qt.nokia.com/>

- wxWidgets (for GUI) and standard C++ for Kernel
- Qt (for GUI) and standard C++ for Kernel.

A summary of the key factors that influenced the selection of the development environment platform is depicted in Table 1.

**Table 1. Comparison of potential solutions**

	Java (GUI and Kernel)	Java (GUI) C++ (Kernel)	wxWidgets (GUI) C++ (Kernel)	Qt (GUI) C++ (Kernel)
<b>Platform</b>	Windows Unix/Linux Mac	Windows Unix/Linux Mac	Windows Unix/Linux Mac	Windows Unix/Linux Mac
<b>License</b>	Free	Free	Free	Open Source: Free Proprietary: Fee
<b>Compiler</b>	Sun JDK (free)	Sun JDK (free) Visual Studio (Windows, not free) G++ (Linux, free)	Visual Studio (Windows, not free) G++ (Linux, free)	Visual Studio (Windows, not free) G++ (Linux, free)
<b>Reliability</b>	Very Good	Good	Good	Very Good
<b>Flexibility</b>	Very Good	Very Good	Very Good	Very Good
<b>Efficiency</b>	Good	Good	Very Good	Very Good

The final selection was to develop the Kernel in C++ and the GUI in Qt.

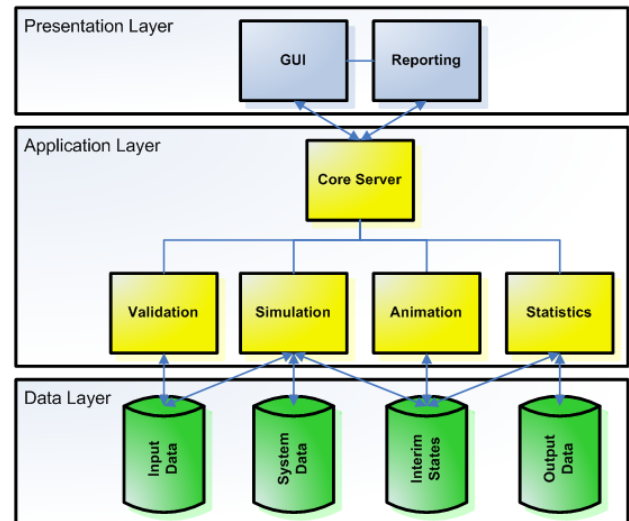
#### 4 ADSIM+ Design Description

This section depicts the basic properties of the ADSIM+ design along with some remarks on the current software implementation.

As shown in Fig. 2, ADSIM+ has a three-tiered architecture. The layers of this architecture are referred to as the following:

- Presentation Layer
- Application Layer
- Data Layer.

Each layer only communicates directly with adjoining layers. The Presentation Layer only communicates with the Application Layer by requesting services via the Application Layer's Core Server module. In turn, the modules of the Application Layer communicate with the Data Layer by querying its data modules.



**Fig. 2. ADSIM+ Layers and Subsystem Modules**

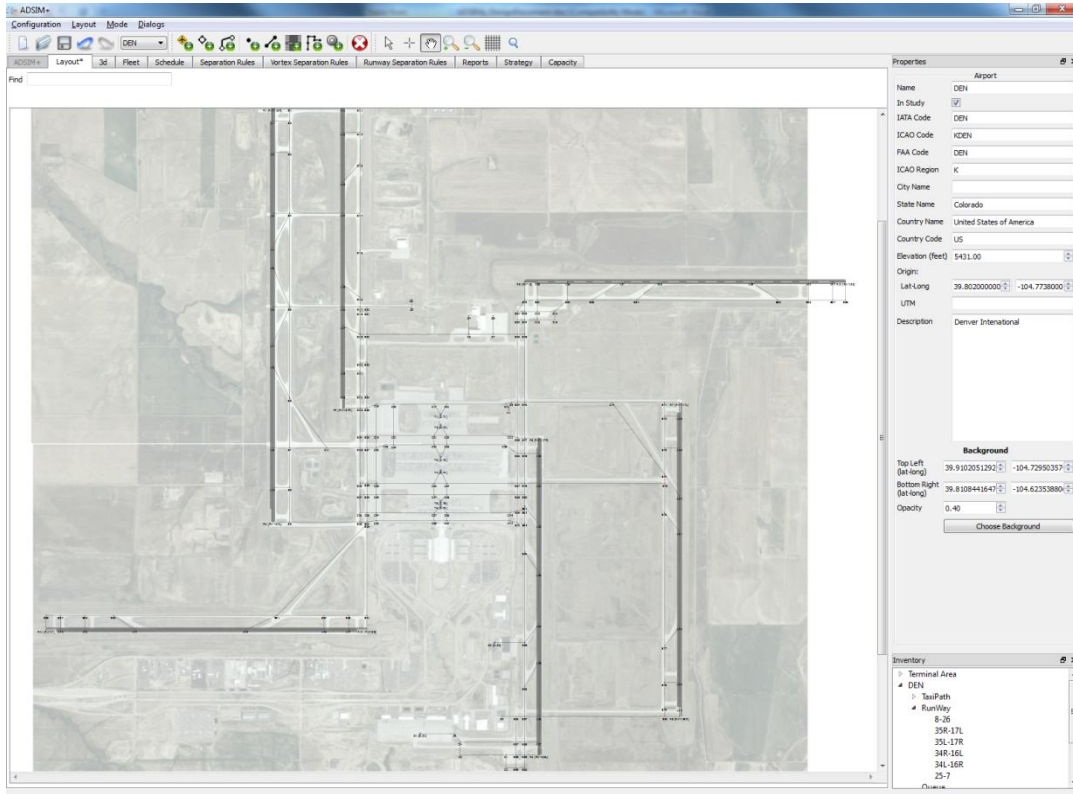
The Presentation Layer contains the following two modules:

- **Graphical User Interface (GUI).** The GUI provides users with graphical interaction and visualizations. It is responsible for all of the windows used for entering input parameters, controlling program execution, displaying simulation animations and viewing reports. The user-issued commands through the manipulation of displayed graphical elements are translated into service requests which are sent to the Application Layer's Core Server. The functionality of this module does not exist in the legacy ADSIM.

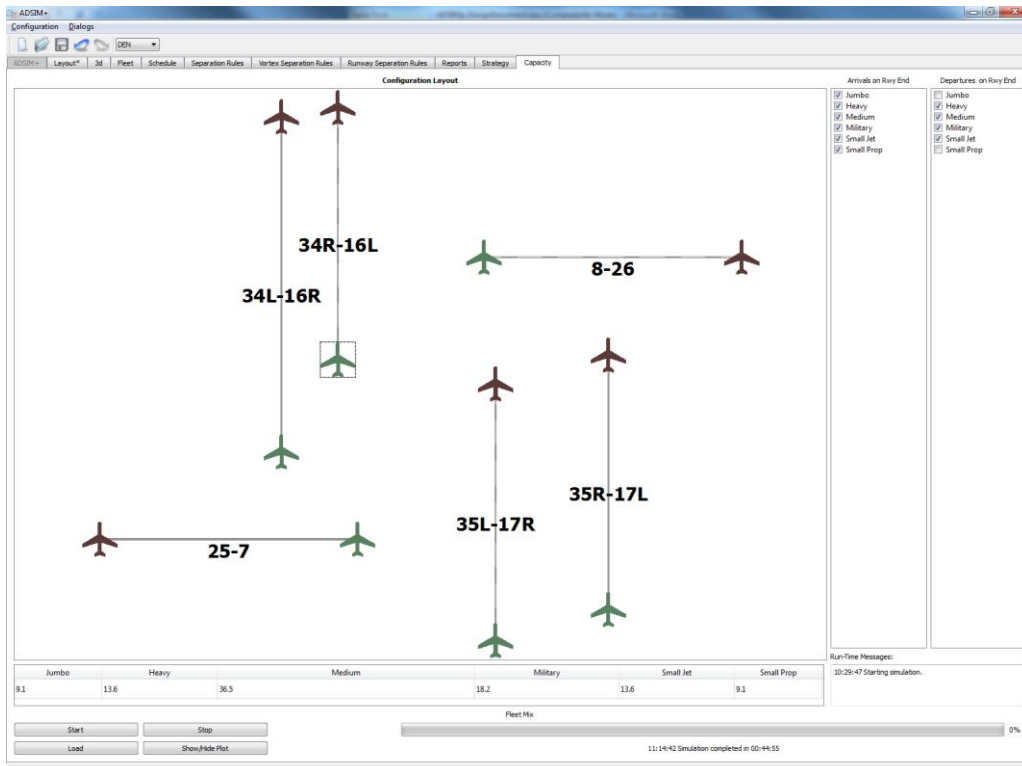
The ADSIM+ Layout tab, which allows the user to define airport and terminal area characteristics (e.g., arrival/departure routes, runways, gates, taxiways) as well as to view 2D animation, is depicted in Fig. 3. As can be seen from the figure, ADSIM+ allows the user to import a background map that can help in defining some of the airport properties (the user can trace over the imported map while defining, for example, runways, taxiways and gates). In its final stage, the GUI will allow the user to import properties of airports and/or terminal areas. The ADSIM+ capacity mode tab is depicted in Fig. 4. The user is able to define the runway configuration for which the capacity is being evaluated along with a fleet mix. The user also is able to specify a set of

# ADSIM+: A NEW DEVELOPMENT OF THE AIRFIELD DELAY SIMULATION MODEL

separation rules that apply at the times when capacity is evaluated.



**Fig. 3. ADSIM+ GUI**



**Fig. 4. ADSIM+ Capacity Mode**

- **Reporting.** The Reporting module contains the functionality necessary to generate reports, such as how each report is organized, which data elements are shown and how each field is formatted. Functionality for this module exists in the legacy ADSIM to generate text files. A sample report generated while running ADSIM+ in the delay mode is depicted in Fig. 5. A sample result of the capacity mode (a capacity envelope) is depicted in Fig. 6.

The Application Layer contains the following five modules:

- **Core Server.** The Core Server provides an interface for the Application Layer which receives user requests, directs user commands to the appropriate subsystem module(s), and returns results to the Presentation Layer.
- **Validation.** The Validation module evaluates input data as it is entered by the user into the Input data module. After data entry is complete, the Validation module evaluates input data sets as a whole for internal consistency. Validation prevents incomplete or inconsistent data from being modelled.
- **Simulation.** The Simulation module runs the event-driven simulation model of terminal area operations using input from the Input data module and storing the resulting sequence of events and model states in the Interim States data module. Depending on the user's selection, this module will run in the delay or capacity mode. While the focus of the delay mode is a terminal area, or an airport, the focus of the capacity mode is a specific runway configuration of an airport.
- **Animation.** The Animation module transforms event-driven model states into interpolated time-driven model states suitable for fast-time animation of airport operations which can be viewed concurrently with a simulation run or during the analysis of a previous simulation run.

The functionality of this module is not integrated in the legacy ADSIM. Presently, ADSIM+ will allow 2D animation (various sample rates) concurrently with a simulation run. It also will allow 2D and 3D animations of previous runs.

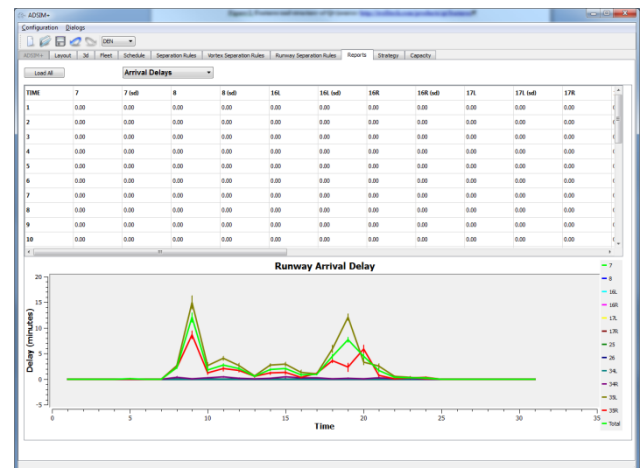


Fig. 5. Sample of ADSIM+ Reports Tab

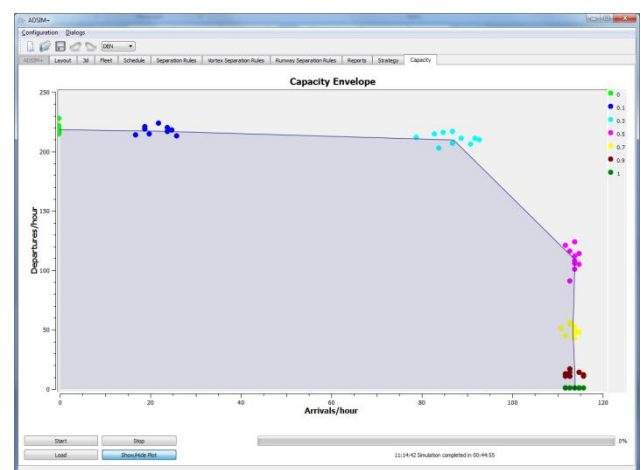


Fig. 6. Sample of ADSIM+ Capacity Envelope

- **Statistics.** The Statistics module computes statistical variables on the dataset stored in the Interim States and stores the results in the Output data module. Thus, this module is responsible for the content of the reports (samples are depicted in Fig. 5 and Fig. 6).

The Data Layer contains the following three modules:

- **Input Data.** The Input data module provides access to and storage for the simulation input parameter values such as the terminal area and airport(s) layout, schedule, aircraft separation rules, and resource load balancing strategies.



## ADSIM+: A NEW DEVELOPMENT OF THE AIRFIELD DELAY SIMULATION MODEL

- **System Data.** The System data module provides access to and storage for the default system parameter values for aircraft performance, the airport layout, and 3D aircraft models. Presently, the 3D aircraft models are the only system data available.
- **Interim States.** The Interim States data module provides access to and storage for the various model states throughout a simulation run. The user can access this data outside of the GUI and generate additional reports as necessary.
- **Output Data.** The Output data module provides access to and storage for the output data which can be displayed in a GUI, appear in a report or be used as an input to other applications.

### 5 Model Verification and Validation

ADSIM+ is partially derived from the legacy ADSIM model, which has been validated extensively. Still, ADSIM+ will need to undergo a significant verification and validation process. In the verification phase we are comparing simulation results obtained in ADSIM+ runs to those obtained in the legacy ADSIM. Although both models are discrete-event Monte-Carlo simulations and both are based on the link-node description of the airport resources (e.g., runways, gates, taxiways), there are modeling differences. Thus, a statistical comparison between results of the models is being conducted. To be successful, the statistical comparison will need to include multiple scenarios. Thus far the verification of ADSIM+ has only been performed on a limited set of scenarios. The initial comparison was based on flow rates and average delays for each runway and operation type (arrival/departure) and time bin. The comparison between ADSIM and ADSIM+ flow rates based on a single scenario (DEN airport) are depicted in Fig. 7 and Fig. 8. Both arrival and departure flow rates show a good agreement between ADSIM and ADSIM+ outputs.

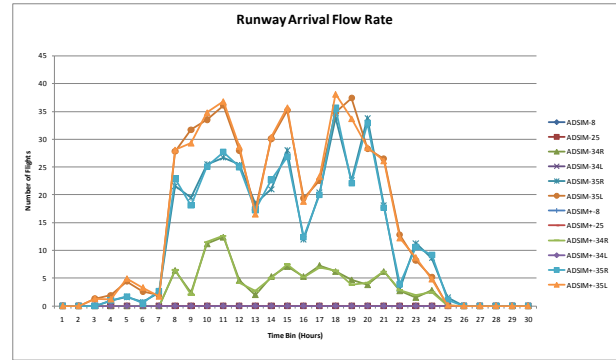


Fig. 7. Runway arrival flow rates (DEN scenario)

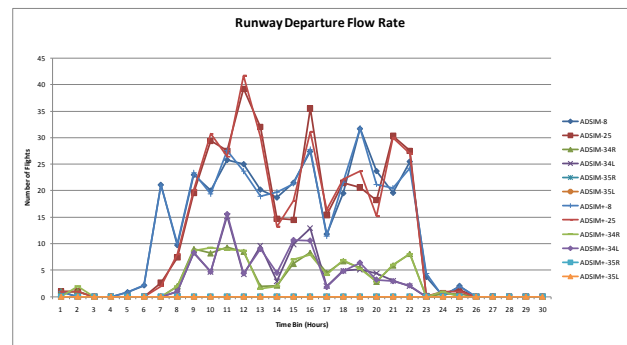


Fig. 8. Runway departure flow rate (DEN scenario)

The comparison between ADSIM and ADSIM+ runway delay outputs for the same scenario for arrivals and departures is depicted in Fig. 9 and Fig. 10 respectively. These results also exhibit a good agreement between ADSIM and ADSIM+ outputs.

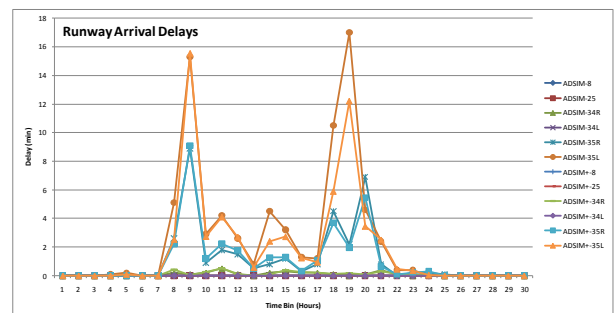


Fig. 9 Runway arrival delays (DEN scenario)

We will be using more scenarios to complete a statistical verification of ADSIM+ results. Additionally, validation of ADSIM+ results will be done by using a (large) set of real-life data obtained through ETMS/ASDI, PDARS, and/or ASDE-X data feeds.

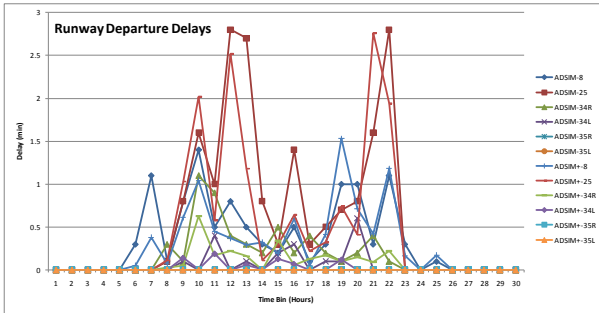


Fig. 10. Runway departure delays (DEN scenario)

## 6 Summary and Future Steps

This paper outlines the current state of ADSIM+ development. Compared to other airport simulation models, ADSIM+ is envisioned to be a trade-off between the more simplistic (e.g., rS) and elaborate (e.g., TAAM) models. ADSIM+ focuses on the operations of aircraft inside the airport terminal area and has two modes of operation: delay and capacity. In contrast to the delay mode, which considers the entire terminal area, the capacity mode considers a specific airport configuration.

Future steps include:

- Continuation of the verification-validation process;
- Enhancement of the System Data (e.g., aircraft performance characteristics and separation rules);
- Augmented input error checking.

## 7 References

- [1] Odoni, A.R., Bowman, J., Delahaye, D., Deyst, J.J., Feron, E., Hansman, R.J., Khan, K., Kuchar, J.K., Pujet, N., Simpson, R.W., Existing and required modeling capabilities for evaluating ATM systems and concepts, Technical Report, International Center for Air Transportation, MIT, Cambridge, MA, 1997.
- [2] Monk, H., Holladay, R., Morris, J., Airfield Delay Simulation Model (ADSIM) User's Guide, DOT/FAA/PM-84/2, U.S. Department of Transportation, Federal Aviation Administration, WHTC Atlantic City Airport, New Jersey, 1984-2007.
- [3] RAMS: Reorganized ATC Mathematical Simulator, MIT AATT Model Project, 1996. <http://web.mit.edu/aeroastro/www/labs/AATT/review/s/rams.html>

- [4] RAMS Plus Simulation Solutions, ISA Software, 2010. <http://www.ramsplus.com>
- [5] Barrer, J.N., Kuzminski, P., Swedish W.J., Analyzing the Runway Capacity of Complex Airports, AIAA 5th Aviation, Technology, Integration, and Operations Conference, Arlington, Virginia, 2005.
- [6] How Simmod Works (SIMMOD Manual), FAA, [http://www.tc.faa.gov/acb300/how\\_simmod\\_works.pdf](http://www.tc.faa.gov/acb300/how_simmod_works.pdf)
- [7] TAAM Reference Manual for Version 2.4, Preston Aviation Solutions Pty Ltd, 2005.
- [8] User Manual for the Base of Aircraft Data (BADA) REVISION 3.8, EEC Technical/Scientific Report No. 2010-003, 2010.

## 8 Contact Author Email Address

Panta Lucic, [plucic@cssiinc.com](mailto:plucic@cssiinc.com)

## 9 Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS2010 proceedings or as individual off-prints from the proceedings.