A Queuing Framework for Terminal Area Operations

Monish D. Tandale*, Sai Vaddi†, Sandy Wiraatmadja‡ and Victor H. L. Cheng§.

Optimal Synthesis Inc., Los Altos, CA, 94022

As a part of NASA’s NextGen research effort, the focus area of Airspace Super-Density Operations (ASDO) performs research pertaining to highly efficient operations at the busiest airports and terminal airspaces. It is expected that multiple ASDO concepts will be interacting with one another in a complex stochastic manner. This research effort developed a high-fidelity queuing model of the terminal area suitable for the design and analysis of NextGen ASDO concepts, as well as to perform time-varying stochastic analysis of terminal area operations with regards to schedule and wind uncertainties. A unique aspect of the current approach is the discretization of terminal airspace routes into 3-nmi servers for enforcing separation requirements. The current research effort developed high-fidelity queuing models of the San Francisco International Airport (SFO) terminal airspace, based on published airspace geometry. A discrete-event simulation framework was developed to simulate the temporal evolution of flights in the terminal area. The queuing simulation framework was used in different case studies involving various phenomena in the terminal area such as compression, conflict and delay analysis, runway reconfiguration and variable inter-aircraft separation. In addition to being a useful analysis tool, the proposed simulation framework shows potential as a real time stochastic decision support tool due to its low computational cost.

Nomenclature

ASDI  Airspace Situational Data to Industry
ASDO  Airspace Super Density Operations
CARPAT Computational Appliance for Rapid Prediction of Aircraft Trajectories
DAC  Dynamic Airspace Configuration
DAFIF Digital Flight Information Files
DEQS Discrete-Event Queuing Simulation
DFW Dallas/Fort Worth International Airport
DP Departure Procedure
d-TPP digital-Terminal Procedures Publication
GPS Global Positioning System
IAP Instrument Approach Procedure
ILS Instrument Landing System
IMC Instrument Meteorological Conditions
MIT Miles In Trail
NACO National Aeronautical Charting Office
NextGen Next Generation Air Transportation System

* Research Scientist, 95 First Street, monish@optisyn.com, Senior Member AIAA.
† Research Scientist, 95 First Street, vaddi@optisyn.com, Senior Member AIAA.
‡ Research Engineer, 95 First Street, sandy@optisyn.com.
§ Vice-President and Chief Technology Officer, vcheng@optisyn.com, Associate Fellow AIAA.
I. Introduction

NASA and FAA have been involved in extensive research efforts to develop advanced concepts, technologies, and procedures for the Next Generation Air Transportation System (NextGen). Among the various areas explored under these research efforts, the research focus area ‘Airspace Super Density Operations (ASDO)’ seeks to develop efficient terminal area operations. The vision for NextGen Airspace Super Density Operations includes the following concepts:

1. Automation for optimal scheduling, sequencing, route assignment and runway assignment based on optimization algorithms.
2. 4D trajectory management for scheduling conflict free trajectories using Area Navigation (RNAV) and Required Navigation Performance (RNP) routes.
3. Conflict free continuous descent arrivals from the top of descent to the runway threshold for multiple flights to multiple airports of a metropolis.
4. Ground based and flight deck based automation systems for conflict detection, separation assurance and merging and spacing operations.
5. Very closely spaced parallel runway operations and simultaneous operations on intersecting runways under instrument meteorological conditions (IMC).
6. Integrated arrival, departure, and surface operations.
7. Optimal runway configuration management based on predicted weather and demand.

It is expected that multiple ASDO concepts will be interacting with one another in a complex stochastic manner. Therefore, the overall system performance may not be a straightforward combination of individual performance indices. It is also crucial that the overall system performance is robust to wind and operational uncertainties.

This research effort developed a fast-time stochastic analysis tool based on queuing theory that can be used to evaluate the interaction and combined performance of multiple airspace super-density operation concepts. The primary objective of this research is to create a high-fidelity queuing model of the terminal area and a framework for performing time-varying stochastic analysis of terminal area operations with regards to schedule and operational uncertainties due to factors such as wind.

The proposed simulation tool can be useful for the following purposes:

1. It can serve as an analysis tool for studying new terminal area operational concepts such as VCSPRO.
2. It facilitates integrated study of multiple NextGen concepts using high-fidelity, fast-time simulations.
3. It can serve as a fast-time stochastic simulation tool providing near real-time decision support to validate the robustness of schedules generated from optimal planners, to uncertainties such as wind.
4. It can serve as a design tool to create new arrival and departure routes in a metropolis.
5. In certain scenarios, the simulation tool can be used as a zero-conflict scheduling algorithm for simultaneously solving the scheduling and, merging & spacing problem in the terminal area.

The remainder of the paper is organized as follows. Section II presents an introduction to the queuing framework for the terminal area. The methodology for developing a queuing model of the SFO terminal area, based on published airspace geometry is described in Section III. Section IV discusses the implementation of the discrete-event queuing simulation employed to propagate the flights enforcing the separation constraint using the queuing abstraction. Section V presents various case studies performed in the terminal area using the developed link-node model and the discrete-event queuing simulation framework. Finally, summary and concluding remarks are presented in Section VI.
II. Queuing Framework for the Terminal Area

A. Spectrum of Terminal Area Queuing Models

Various terminal area models have been developed by researchers to characterize and study the flight operations in the terminal area. Figure 1 shows the wide spectrum of the existing terminal area models.

![Figure 1. Spectrum of Terminal Area Models](image)

On one end are the simple queuing models that are generally used for performing stochastic analysis. The stochastic analysis is performed using closed form queuing equations; therefore, these methods do not model the route geometry accurately and may use single queues to represent the entire route from the arrival fix to the runway. The flight arrival and service characteristics are aggregated into inter-arrival time distributions and service-time distributions. The parameters of the queuing model are generally derived from historic data and may not be relevant with a different traffic volume or after the terminal area route geometry is altered. Thus, such models find limited use in NextGen concept analysis.

On the other end of the spectrum are the complex trajectory based models that model the terminal airspace route geometry accurately and use high fidelity aircraft propagation models to perform deterministic propagation of flights through the terminal area. Such models generally require algorithms for trajectory prediction, conflict probing and generating tactical maneuvers for conflict avoidance. Tactical control mechanisms such as vectoring to maintain separation assurance have to be explicitly coded in the trajectory propagation equations. Stochastic analysis using such models performed through Monte Carlo simulations can be computationally intensive.

The queuing models developed during the course of this research fall in between the two models described above. These queuing models model the terminal area route geometry with high fidelity. A queuing abstraction is used to model the tactical control mechanisms for separation assurance as illustrated in Figure 2.

![Figure 2. Queuing Abstraction](image)
A separation constraint of 3 nmi exists in the terminal area. In the actual operation, the trailing aircraft are vectored to incur a delay so that the aircraft do not violate the 3-nmi separation constraint, as illustrated in the figure on the left. The figure on the right shows the queuing abstraction in which the terminal area routes are segmented into smaller sections called servers, with the length of each server being 3 nmi. Separation is ensured by enforcing the rule that only one aircraft can occupy a given server at a given time. This is analogous to the queuing theory principle that only one customer can be served by a single server at a time. Aircraft A1 in Server1 has to wait in queue until aircraft A2 has finished service at Server2. Thus, the wait in queue is an abstraction of the vectoring and the wait time in queue is equivalent to the delay incurred due to vectoring.

The developed queuing models are analyzed by performing a discrete event queuing simulation. The discrete event simulation does not propagate the aircraft at small time steps but jumps from one significant event to another. The significant event can be an entry of an aircraft into the network, the finishing of service of an aircraft at a server or a transition of aircraft from one server to another. Since the discrete event simulation jumps from one significant event to the next, it runs much faster than the aircraft propagation at small time steps. Hence, the Monte Carlo simulation using the discrete event simulation framework is expected to be much faster than the Monte Carlo simulation using detailed aircraft propagation models, without suffering a major loss in fidelity.

B. Technical Approach

Figure 3 shows an overview of the technical approach. The first step involves synthesizing a detailed link-node model of the terminal area from airspace data such as departure and arrival fixes, Standard Instrument Departure (SID), Standard Terminal Arrival Route (STAR), RNAV routes and procedures. The link-node model is then cast into a queuing network abstraction of the terminal area operations. This involves translating the link-node geometry and filed flight schedule into queuing network parameters such as number of servers, routing matrix or server sequence, arrival times, and service time distributions. Subsequently analysis can be performed on the queuing model to compute traffic flow efficiency metrics such as delay, throughput, and landing times. The above framework can be used to perform various analyses such as the evaluation of NextGen concepts in the terminal area, efficiency estimation for rapidly prototyped routes generated by a Dynamic Airspace Configuration (DAC) algorithm in response to weather or for the design of metroplex routes and runway configurations.

C. Link-Node Model of the Terminal Airspace

Air traffic in the terminal area has to adhere to the defined terminal area routes and procedures. This research proposes to incorporate the details of the terminal area routes and procedures into the queuing network model. Current Terminal Procedures published by the Federal Aviation Administration are available as digital-Terminal Procedures Publication (d-TPP) at the FAA National Aeronautical Charting Office (NACO) or at AirNav.com.

The terminal route geometry obtained from the above sources is converted into a link-node model. The routes are divided into 3 nautical mile segments corresponding to the separation distance, with each link representing a single server of the queuing network. If a new concept allows separation to be reduced below 3 nmi, the link-node
can be created as per the new standard. Figure 4 shows a link-node model for RNAV departure route AKUNA 2, for the Dallas/Fort Worth International Airport (DFW). Figure 5 shows a link-node model for arrivals into the San Francisco International Airport (SFO) in the West plan configuration. Thus, the queuing model formulated from this detailed link-node model, represents the terminal area geometry with high fidelity.

A queuing model analyzes situations wherein a resource (airspace) is shared among multiple customers (aircraft) and provides statistics for the delays due to unavailability of the resource (congestion). A queuing network model is traditionally defined by parameters such as 1.) External Arrival Rate Distribution, 2.) Service Rate Distribution, 3.) Number of Servers, and 4.) Routing Matrix. The current research effort takes a slightly different approach than the traditional aggregation of the aircraft arrival, departure and routing characteristics into inter-arrival time distribution, service time distribution and routing flow fractions matrix respectively. The current research retains the specific characteristics of each individual flight while conducting the queuing simulation. It treats arrival streams as finite sequences with an individual time of arrival for each flight. Every aircraft has an arrival time into the network, a service time for every server in the network based on its aircraft type and the desired altitude profile, and a sequence of servers that describes its intended routing through the network. In this way, the fidelity of the queuing analysis is increased by retaining the specific characteristics of each flight in the network. Since the arrival time for every flight is specified, the notion of a schedule is maintained, and study of the effect of input schedule uncertainties on the congestion delays can be performed. Note that the notion of a schedule is lost when the arrivals to the network are aggregated into an inter-arrival time distribution.

Flights in the terminal area are simulated using a link-node model and discrete-event simulation based on the queuing concept. The link-node model creates a network of routes from the published terminal area routes, and discretizes the routes into smaller size links called servers. The servers are used to enforce the separation constraint in the Discrete-Event Queuing Simulation (DEQS) framework. In contrast to propagating every aircraft using fixed time-steps, the discrete-event queuing simulation identifies significant events and jumps to the next significant event. Section III describes the development of a link-node model for the SFO ‘West Plan’ configuration. The DEQS framework is described in Section IV.

### III. Link-Node Queuing Network Model for SFO

The first step in the creation of a link-node model is the identification of arrival and departure routes. The route structure for the SFO ‘West Plan’ configuration is illustrated in Figure 6. Terminal airspaces are typically described by published Standard Terminal Arrival Routes (STARs), Departure Procedures (DPs) and Instrument Approach
Procedures (IAPs). Link-node models are created by splitting the pathways created by the STARs, DPs and IAPs into smaller segments while taking into account the minimum inter-aircraft separation requirements. The STARs, DPs, and IAPs consist of a sequence of waypoints described by the latitude and longitude coordinates. In this work only arrivals are considered, departures will be included in future work. Some waypoints such as "BOLDR" on the BIGSUR STAR shown in Figure 7 also have additional information such as altitude and speed restrictions. Data pertaining to the STARs is available from the FAA's Digital Terminal Procedure Publication (d-TPP) and the National Geospatial-Intelligence Agency’s Digital Flight Information Files (DAFIF).

The following STARs and approach procedures for SFO airport were obtained from References 26 and 28:
1. GOLDEN GATE5 Arrival (PYE-GOLDN, shown in colors green, magenta, black in Figure 8)
2. MODESTO Arrival (MOD-MOD3, shown in color cyan in Figure 8)
3. YOSEMITE Arrival (Yosemite-YOSEM1, shown in color blue in Figure 8)
4. BIGSUR Arrival (BSR-BSR2, shown in color red in Figure 8)
5. LOC RWY 28L (shown in light green in Figure 8)
6. RNAV (GPS) RWY 28L (shown in light green in Figure 8)

The STARs are connected to the ILS and GPS approach procedures in consultation with a subject matter expert who is a Traffic Management Controller (TMC) at the Oakland ARTCC. The connected arrival pathways to SFO are shown in Figure 8. The GOLDN5 five arrival route is split into two routes at ‘DUXYB’; one route travels over the bay and the other route travels over the peninsula.

Five individual terminal area arrival pathways are thus created. These pathways are referred to as GOLDNBAY, GOLDNPEN, MOD, YOSEM, and BSR in the remainder of this report. The subscripts ‘BAY’ and ‘PEN’ for the GOLDN route represent the two pathways resulting from the split at DUXBY. The pathways can be described by the following sequences of waypoints (marked by asterisk (*) marks in Figure 8):
1. GOLDNBAY: PYE→LOZIT→DUXYB→GGB1→GGB2→DUYET→SFO
2. GOLDNPEN: PYE→LOZIT→DUXYB→SFO→GGP1→MENLO→CEPW→DUYET→SFO
3. MOD: CEDES→OOMEN→MEHTA→RAMND→CEPW→DUYET→SFO
4. YOSEM: STEEV→FAITH→MEHTA→RAMND→CEPW→DUYET→SFO
5. BSR: SKUNK→BOLDR→MENLO→CEPW→DUYET→SFO

Whereas most of the above waypoints are standard way points in DAFIF, additional way-points GGB1, GGB2, and GGP1 are created to connect the DUXBY waypoint to DUYET along GOLDNBAY and GOLDNPEN. The queuing network in Figure 8 consists of a total of 18 waypoints distributed over 12 constant heading branches.
Individual pathways can be described by a sequence of waypoints - the number of which varies from 6 along BSR to 9 along GOLDN\textsubscript{PEN}. Further discretization of the 12 branches is necessary to model the inter-aircraft separation requirements. To achieve this, the latitude and longitude coordinates associated with each waypoint of the pathway are converted to Cartesian coordinates (with origin chosen at the SFO airport) using equations (1) and (2).

\[
x_{wp} = R_e(\lambda_{wp} - \lambda_{SFO}) = R_e\Delta\lambda
\]

\[
y_{wp} = R_e\cos\lambda_{SFO}(\tau_{wp} - \tau_{SFO}) = R_e\cos\lambda_{SFO}\Delta\tau
\]

where, \( R_e \) is the radius of the Earth, \( \lambda \) is the latitude, \( \tau \) is the longitude, \( x \) is the North position and \( y \) is the East position and the subscript "wp" refers to waypoint.

![Figure 8. Arrival Pathways to Runway 28L at SFO](image1)

![Figure 9. Numbers of Select Servers in SFO Terminal Area Link-Node Model](image2)

The Euclidean distance between the waypoints is computed and each branch of the pathway is divided into links of length 3 nmi. Note that rounding approximations enter this network formulation due to the fact that integer number of links must be used for various branches of the network. The links are referred to as servers to fit them into the queuing simulation framework discussed in the following section. The servers are given a unique identification number and the arrival pathways are described by a sequence of server numbers. The servers are shown in Figure 9 by black dots; some server numbers are also labeled for the purpose of illustration. The server sequences thus created are referred to as the queuing network in the remainder of this report. The queuing network shown in Figure 9 consists of 53 servers. The number of servers along the arrival pathways ranges from 14 servers on BSR and YOSEM, to 23 servers on GOLDN\textsubscript{PEN}. The server sequences are not necessarily continuous sequence of integers, for example the server sequence for GOLDNBAY is as follows: {1 2 3 4 5 6 7 8 9 10 35 36 37 38 51 52 53}

The link-node model presented in this section is particular to the West plan configuration at SFO. This link-node model is used for evaluating the robustness of schedule planners to wind uncertainty which is further described in a companion paper\textsuperscript{30}. Link-node model with arrival routes on the South-East plan at SFO and a larger terminal radius are used to study runway reconfiguration in Section V.C. Finer discretization of servers is used in Section V.D to implement variable inter-aircraft separation on the final approach. The next section describes the DEQS framework used to simulate traffic on the arrival pathways.

### IV. Discrete-Event Queuing Simulation Framework

The arrival pathways and the associated server sequences define the routes along which the arrival flights are expected to travel. To compute the four dimensional trajectories, flight times, conflicts, and delays experienced by different flights along the arrival pathways, it is necessary to further model the following flight details:

1. Speed profile of the arrival flights
2. Speed and direction of the prevailing winds
3. Tactical control logic for maintaining inter-aircraft separation

Using the above information aircraft equations of motion can be integrated using a numerical integration routine such as the Euler integration algorithm or the Runge-Kutta integration algorithm. This approach generally requires
algorithms for trajectory prediction, conflict probing and generating tactical maneuvers for conflict avoidance. Tactical control mechanisms such as vectoring to maintain separation assurance have to be explicitly coded in the aircraft propagation. The above approach can be computationally expensive.

The current work adopts a different approach that can be described as follows:

1. **Temporal discretization**: A discrete-event simulation framework is used in the current work. This obviates the necessity to integrate equations of motion with a constant time step. The simulation identifies the next significant event and jumps directly to the time associated with that event.

2. **Spatial discretization**: The positions of the flights are described using the discretized server locations. Therefore, each flight depending on its route can assume finite position coordinates or equivalently server numbers. At every event of the discrete-event simulation, at most one flight progresses to its next server as per its server sequence.

3. **Queueing**: When the progression of a flight to its subsequent server is impeded by another flight, the first flight is added to the queue of its subsequent server. The server concept facilitates enforcing the separation requirements in DEQS by restricting the capacity of the servers to 1.

This approach only calculates the delay and does not specify how the delay should be implemented. As a consequence there is some loss of fidelity, as this approach does not test whether the delay can be enforced through a feasible trajectory modification. However, limits on the maximum amount of delay (e.g., 5% of flight time) can be enforced and the fidelity of the simulation can be maintained. The primary advantage gained by accounting for delay as wait, is the significant increase in execution speed, which makes stochastic analysis using the Monte Carlo approach feasible. The following sections further elaborate the details of this framework.

### 1. Nomenclature & Definitions

The following definitions and nomenclature are useful for further description.

1. \( F = \{ f_1, f_2, \ldots, f_N \} \) Set of arrival flights over a chosen time horizon of interest
2. \( S = \{ s_1, s_2, \ldots, s_N \} \) Set of terminal area servers
3. \( S_i = \{ s_{i1}, s_{i2}, \ldots, s_{iN_i} \} \) Server sequence for the \( i \)th flight
4. \( N_i \) = Number of servers along the path of the \( i \)th flight
5. \( TOA_{i,s} \) = Scheduled time of arrival for the \( i \)th flight at its entry server \( s_{i1} \)
6. \( TOA_i \) = Actual time of arrival for the \( i \)th flight at its entry server \( s_{i1} \)
7. \( TOD_{i,j,s} \) = Scheduled time of departure for the \( i \)th flight from its \( j \)th server \( s_{ij} \)
8. \( TOD_i \) = Actual time of departure for the \( i \)th flight from its \( j \)th server \( s_{ij} \)
9. \( V_i = \{ V_{i1}, V_{i2}, \ldots, V_{iN_i} \} \) = Airspeed profile for of \( i \)th flight over its server sequence \( S_i \)
10. \( V_{ij} \) = Airspeed of \( i \)th flight on its \( j \)th server \( s_{ij} \)
11. \( \chi_{ij} \) = Heading angle of \( i \)th flight on its \( j \)th server \( s_{ij} \)
12. \( W_{ij} \) = Wind speed experienced by \( i \)th flight on its \( j \)th server \( s_{ij} \)
13. \( \chi_{ij,w} \) = Wind heading angle experienced by \( i \)th flight on its \( j \)th server \( s_{ij} \)
14. \( \Delta s_{ij} \) = Path length of \( j \)th server \( s_{ij} \) for the \( i \)th flight
15. \( \Delta T_{ij} \) = Service time at the \( j \)th server \( s_{ij} \) for the \( i \)th flight
16. \( D_{ij} \) = Delay experienced by the \( i \)th flight at its \( j \)th server \( s_{ij} \)

### 2. Input Arrival Streams

Each arrival to SFO is modeled as an input to the queuing network characterized by the following:

1. A sequence of servers along the pathway from entry server to the landing server at SFO represented by \( S_i \)
2. Scheduled time of arrival to the entry server represented by \( TOA_{i,s} \)
3. Sequence of airspeed values at each server along its server sequence \( V_i \). The current framework does not make any assumptions on the airspeed profiles of the individual flights.

### 3. Time of Arrival and Time of Departure Computation

The service time for a flight in a particular server depends on the ground speed adopted by that flight over that particular server. Given airspeed profile specified by speeds over each server along the pathway, the service time for the \( i \)th flight at its \( j \)th server is computed using equation (3):

\[
\Delta T_{ij} = \frac{\text{server length}}{\text{ground speed}} = \frac{\Delta s_{ij}}{V_{ij} + W_{ij} \cos (\chi_{ij} - \chi_{ij,w})} 
\]  

(3)

Time of departure from a server and time of arrival to a server are computed using the following equations:

\[
TOD_{ij} = TOA_{ij} + \Delta T_{ij} 
\]  

(4)
4. Events & Simulation Logic

The following events are of interest in the discrete-event simulation:

1. Internal transition
2. Internal conflict
3. External arrival
4. External conflict

The following procedure describes the simulation logic to detect the above mentioned events:

1. The DEQS tracks the state of the terminal area traffic using the following information for each flight:
   - Flight id
   - Current server id
   - ID of the subsequent server.
   - TOAs to the subsequent server

2. The state of the system is updated at discrete-times and time instances are recorded as event times.

3. Events are detected based on the TOA that is closest to the current time. Once the “next-event time” is identified:
   - Flight corresponding to the above TOA is identified as the “next-event flight”.
   - Subsequent server for the “next-event flight” is identified as “target server”.

4. If “target server” is not occupied:
   - Event is recorded as “internal transition” if “next-event flight” is inside the network
   - Current server of “next-event flight” is updated to “target server”
   - Simulation time is updated to TOA of “next-event flight” to “target server”
   - If “target server” is terminal server the “next-event flight” is removed from simulation state
   - If “next-event flight” is a new arrival the “next-event flight” is added to the simulation state

5. If “target server” is occupied:
   - Event is recorded as “internal conflict” if “next-event flight” is inside the network
   - Current server of “next-event flight” is updated to “target server”
   - The “next-event flight” is added to the queue of the “target server”
   - Delay for “next-event flight” is computed as the difference in TOD of “next-event flight” from current server to the TOD of the “conflict flight” from “target server”.
   - The computed delay is added to the TOA of “next-event flight” to “target server”
   - The computed delay is also added to TODs of all subsequent servers of the “next-event flight”

5. Simulation Outputs

The following performance metrics are computed as outputs by the DEQS framework:

1. $D_i = \sum_{j \in S_i} D_{ij}$ = Flight-wise delay or total delay experienced by the $i^{th}$ flight
2. $D_{j,s} = \sum_{j \in F} D_{ij}$ = Server-wise delay or total delay experienced by all flights at the $j^{th}$ server
3. $D_{Total} = \sum_{l \in F} D_l$ = Total delay or sum total of delays for all flights
4. $C_{ij} = \begin{cases} 1 & \text{if the } j^{th} \text{ flight experiences a conflict at the } i^{th} \text{ server} \\ 0 & \text{otherwise} \end{cases}$
5. $C_i = \sum_{j \in S_i} C_{ij} = \text{Integer variable representing flight-wise conflicts or the total number of conflicts experienced by the } i^{th} \text{ flight.} \quad (C_i < N_i)$
6. $C_{Total} = \sum_{l \in F} C_l = \text{Total number of conflicts experienced by all flights}$
7. $C_{j,s} = \sum_{l \in F} C_{ij} = \text{Integer variable representing server-wise conflicts or the total number of conflicts experienced by all flights at the } j^{th} \text{ server.}$

V. Terminal Area Case Studies

This section describes the results of various case-studies performed using the developed queuing model for SFO and the discrete event queuing simulation.
A. Compression Case Study

In this section the link-node model and the DEQS framework described earlier are employed to verify a phenomenon known as “compression” in air-traffic parlance. Arrival flights typically enter the terminal area at a high speed of about 300 knots but slow down to about 130 knots at the time of landing. Therefore, flights further downstream on the pathways and hence closer to the airport travel at lesser speeds than the flights behind them on the same pathway. The difference in speeds causes a decrease in separation resulting in possible separation related conflicts downstream.

A single arrival stream consisting of only those flights arriving from the South via the BSR arrival pathway is considered in this simulation. Hypothetical streams of 22 flights, each stream characterized by a constant inter-aircraft Miles-In-Trail (MIT) separation at the entry server are created. The flights are propagated through the BSR server sequence using the DEQS framework. A linear speed variation from 300 knots to 130 knots is used for all the flights. Wind is not included in the simulation. The speed profile and the constant size 3-nmi servers result in service times that are monotonically increasing from the entry server to the last server. Shown in Figure 10 are the service times of the 14 servers along the BSR pathway. It should be noted that the service time of the last server (server #14) is more than twice the service time of the entry server (server #1). The difference in service times indicates that aircraft travel twice as slow in the terminal servers as compared to the entry servers.

Total conflicts for all 22 flights (as per the definition in Section IV.5) resulting due to the loss of separation at servers between the entry and landing servers are shown in Figure 11. The independent variable in these plots is the MIT spacing of arrivals at the entry server. Flights entering the terminal area at 3-nmi separations are seen to experience as many as 270 conflicts. The corresponding total delay is 9471 seconds as seen in Figure 12. Therefore, it can be concluded that 3-nmi separation at the entry server is not sufficient to ensure 3-nmi separation at all subsequent servers. It is a common practice among air-traffic controllers to use higher value of separation away from the airport.

Figure 10. Service Time for Different Servers along BIGSUR STAR

Figure 11. Number of Conflicts as a Function of MIT Spacing at the Entry Server

Shown in Figure 12 are the total delays for all the flights over all the 14 servers as a function of the MIT separation at the entry server. It can be inferred from the figure that separation related conflicts and the associated delay can be eliminated if the MIT separation at the entry server is more than 6.5 nmi. It should be noted that an inter aircraft MIT spacing of 6.5 nmi corresponds to an inter-aircraft time spacing that is about equal to the service time of the 14th server which is the bottleneck server in the sequence.

Figure 13 shows the total time required to process and land all the 22 flights in the input stream; this time is a measure of the throughput. The total time to process the 22 flights remains the same when the inter-aircraft separation is less than 6.5 nmi as expected. However, any further increase in separation results in down-time for the slowest server and degrades the throughput of the system.
Thus the queuing framework correctly accounts for ‘compression’ as it calculates that a larger separation of 6.5 nmi must be employed at the entry to the terminal area, although the separation constraint specified in the queuing framework is only 3 nmi. The queuing framework does not enforce delay at the same location, as should be enforced in actual flight; however it calculates the exact delay that must be imposed in actual flight.

**B. Simulation of Arrivals at SFO**

In this section, the DEQS framework is used to evaluate an actual 4 hour traffic scenario involving flights on all five arrival pathways in the SFO terminal area. Realistic data is necessary to conduct simulations of terminal area traffic using the DEQS framework. OSI has access to Airspace Situational Data to Industry (ASDI) data for all flights arriving at and departing from US airports during the year 2007. The data is stored in files referred to as TRX files. The dates of March 15th and 16th were selected for current case-studies and all flights departing to SFO were isolated from the TRX files. Each flight is assigned to one of the five terminal area pathways identified in Section III, based on the flight-plan information in the TRX files. The flights are then propagated using the trajectory propagator (CARPAT) till they reach the entry point of pre-determined pathway. The time of arrival at the entry way-point is noted. The entry way-points for the SFO terminal area are chosen such that the crossing altitude for these points is less than 10000 ft which falls under class B airspace. The upper limit on speed for class B airspace is 250 KIAS. The typical landing speeds for jet aircraft are in the range of 130-155 knots. For the purpose of the current case-studies, linear speed variation is assumed from the entry server to the landing server for all flights.
The 4.5 hr simulation takes about 4 seconds to execute in MATLAB. A total of 249 conflicts and 61 minutes of delay were observed. Spatial distribution of conflicts is shown in Figure 14 and Figure 15. High densities of conflicts are noted on the final approach and the merge points near the final approach.

The top five server-wise conflicts are shown in Figure 16. Server numbers are indicated on the right hand side of the horizontal bars. Figure 17 shows the top 5 servers that experienced the most delay. It should be noted that the top servers conflict-wise and delay-wise are either on the final approach which is the slowest link or at nearby merge points such as 47. The distribution of the server-wise conflicts is shown in Figure 18. A closer look at the data associated with the leftmost vertical bar reveals that 20 out of the 53 servers experienced zero conflicts. Most of these servers are close to the entry servers.

The top 5 flight-wise delays and the distribution of the flight-wise delays for all the 100 flights are shown in Figure 19 and Figure 20 respectively. From these figures it can be inferred that the maximum flight-wise delay is about 300 seconds and 60% of the flights experienced less than 50 seconds of delay.

The results in this section show that the DEQS framework can rapidly evaluate the terminal area delays due to congestion, given the aircraft entry schedules. Note that the DEQS framework enforces separation constraint throughout the entire terminal area, and not only at the metering points. It also enforces separation constraint at merge points in the terminal area route network. These capabilities can be useful in designing scheduling algorithms that include merging and spacing considerations.

Figure 16. Top 5 Server Conflicts

Figure 17. Top 5 Server Delays

Figure 18. Distribution of Server-Wise Conflicts
C. Runway Reconfiguration Case Study

Figure 21 shows two runway and airspace configurations that are usually adopted within the San Francisco Bay Area terminal airspace. These are referred to as the “West Plan” and the “South-East Plan” respectively. The “West Plan” uses the runways 28L and 28R for arrivals to SFO and runways 1L and 1R for departures from SFO. The “South-East Plan” uses the runways 19L and 19R for arrivals and the runways 10L and 10R for departures at SFO. The active airspace configuration at any given time is determined based on systemic traffic flow interests, local terrain, proximate airport traffic, prevailing wind conditions, weather & environmental constraints and availability of airport assets (gates, ramps, taxiways, approach guidance systems). This case study demonstrates the utility of the developed DEQS in analysis and decision support during the transition period when the runway configuration is switched from one to another.

Runway reconfiguration in the context of the current work involves assigning a runway to all arrival flights based on a given reconfiguration time. Some of these arrival flights may not even have departed from their destination airport, some are en route and some may already be in the terminal area. The assignment problem is particular tricky for flights that are very close to the terminal area or already in the terminal area. Once the airport changes from configuration A to configuration B, it is imperative that no flights land on configuration A. Similarly, it is also expected that flights not land on configuration B before the configuration is active. The two aberrations can occur because of (i) congestion in terminal area and (ii) path length difference between alternative arrival routes.
Congestion in the terminal area can cause a flight to land later than its nominal landing time. Therefore, an assignment that is purely based on nominal landing time can result in a flight arriving after the configuration has changed. On the other hand flights can arrive earlier than their scheduled time because of path-length differences. For example arrivals to SFO from the South have two route options: (i) BIG-SUR in West plan leading to landings on 28L and 28R runways, and (ii) HADLY in South-East plan leading to landings in 19L and 19R runways. Flights traveling these two routes are faced with different path lengths which results in different flight times and hence different landing times. For example, the HADLY arrival route has 43 servers as opposed to 26 on the BIG-SUR arrival route as shown in Figure 22. Therefore, southern arrivals to SFO can experience a flight time difference of 15 minutes along the two routes as is evident from Table 1. In fact, all routes for arrivals on the South-East plan landing on runway 19 are longer than their route counterparts landing on 28. Therefore, flights are expected to land sooner on runway 28 as opposed to runway 19, unless they adjust their speeds to compensate for the time difference. The numbers of Table 1 are computed based on linear speed variation starting with 300 knots at the entry server to 130 knots at the landing server.

![Figure 22. Path Length Variations for Arrival Routes in West Plan and South-East Plan](image)

<table>
<thead>
<tr>
<th>Table 1. Flight Time Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Servers to Runway 28</strong></td>
</tr>
<tr>
<td>Northern Arrivals</td>
</tr>
<tr>
<td>Southern Arrivals</td>
</tr>
<tr>
<td>Eastern Arrivals</td>
</tr>
</tbody>
</table>

The logic used for assigning runways to arrival flights can play a significant role in avoiding anomalous arrival patterns. Two different assignment schemes are chosen for evaluation in the current work. The first scheme is based solely on the nominal landing time and is described as follows:
Scheme 1:
- Given reconfiguration time (time at which it is desired to switch the airport from configuration A to configuration B)
- Compute nominal landing times in configuration A
- If (nominal landing time < reconfiguration time)
  - Assign flight to configuration A
- Else
  - Assign flight to configuration B

The second configuration scheme is based on impeded landing times computed from DEQS. The impeded landing times computed from the DEQS account for the delays in the terminal area and therefore result in more accurate prediction of landing times. The assignment scheme can be described as follows:

Scheme 2:
- Given reconfiguration time (time at which it is desired to switch the airport from configuration A to configuration B)
- Compute nominal landing times for both configuration A and configuration B
- Conduct DEQS using configuration A
- Compute impeded landing times in configuration A
- If (impeded landing time in configuration A < reconfiguration time)
  - Assign flight to configuration A
- Else
  - Assign flight to configuration B
  - If (nominal_landing_time in configuration B < reconfiguration time)
    - Delay flight TOA by difference in nominal landing times = (flight time in configuration A – flight time in configuration B)

Results obtained from runway reconfiguration simulations using the two assignment schemes will be presented in this section. The simulations are based on the operation of the SFO airport. Four simulations involving the following were studied:
1. Configuration change from West plan to South-East plan using assignment Scheme 1
2. Configuration change from West plan to South-East plan using assignment Scheme 2
3. Configuration change from South-East plan to West plan using assignment Scheme 1
4. Configuration change from South-East plan to West plan using assignment Scheme 2

Figure 23 shows the landing times resulting from a simulation of the reconfiguration scenario using assignment scheme 1. The arrival traffic for this simulation is the same traffic that was used the terminal area simulation in Section V.B. The time for reconfiguration has been chosen as 8097 seconds, with the time of arrival of the first flight at the entry server being the datum. The blue dots indicate landings before change of configuration and red dots indicate landings after change of configuration. The reconfiguration time is indicated by the vertical green line. It is desired that all landings on 28 are blue dots and all landings on 19 are red dots. However, as can be inferred from Figure 23 two flights (red dots) arrive at runway 28, 2 minutes after the airport has changed configuration. Figure 24 shows the landing times for the same reconfiguration using assignment scheme 2. It should be noted that the late arrivals have been eliminated. Impeded landing times computed using the queuing simulation are used in assignment scheme 2.

Figure 25 shows the results of the reconfiguration from South-East plan to West plan using assignment scheme 1. In this scenario 3 flights (blue dots) arrive on runway 28 as early as 12 minutes before the change of
configuration. Also, two flights arrive on runway 19 as late as 2 minutes after the change of configuration. Figure 26 shows the landing times for the same reconfiguration using assignment scheme 2 where both late and early arrivals are eliminated.

Figure 23. Landing Times in Reconfiguration from West Plan to South-East Plan Resulting from Assignment Scheme 1

Figure 24. Landing Times in Reconfiguration from West Plan to South-East Plan Resulting from Assignment Scheme 2
Figure 25. Landing Times in Reconfiguration from South-East Plan to West Plan Resulting from Assignment Scheme 1

Figure 26. Landing Times in Reconfiguration from South-East Plan to West Plan Resulting from Assignment Scheme 2

D. Variable In-Trail Inter-Aircraft Separation
The approach presented in this paper, divides the terminal area routes into fixed length servers according to the separation constraint i.e. 3nmi. The question naturally arises whether this approach can handle variable inter-aircraft separation between aircraft, which this section addresses.

Separation requirements on the final approach are different from the 3-nmi separation requirement in the terminal area. The separation requirements on the final approach are function of the types of the leading and trailing aircraft on the final approach. They are also function of the meteorological conditions such as Visual Meteorological Conditions (VMC), Marginal Visual Meteorological Conditions (MVMC), and Instrument Meteorological Conditions (IMC). Implementation of variable inter-aircraft separation in the queuing framework will be discussed in the following sections.

1. Inter-Aircraft Separation on Final Approach

Table 2 shows the inter-aircraft separation requirements on the final approach as a function of the leading and lagging aircraft type under VMC. It is clear from this table that the actual separation requirement can take any value between 1.7 nmi to 3.9 nmi. The average separation would depend on the fleet mix and schedule.

<table>
<thead>
<tr>
<th>VMC</th>
<th>Leading</th>
<th>Heavy</th>
<th>B757</th>
<th>Large</th>
<th>Small</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trailing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heavy</td>
<td>2.60</td>
<td>2.60</td>
<td>1.70</td>
<td>1.70</td>
<td></td>
</tr>
<tr>
<td>B757</td>
<td>3.25</td>
<td>2.60</td>
<td>1.70</td>
<td>1.70</td>
<td></td>
</tr>
<tr>
<td>Large</td>
<td>3.25</td>
<td>2.60</td>
<td>1.70</td>
<td>1.70</td>
<td></td>
</tr>
<tr>
<td>Small</td>
<td>3.90</td>
<td>3.25</td>
<td>2.60</td>
<td>1.70</td>
<td></td>
</tr>
</tbody>
</table>

The 3-nmi servers used in the terminal area are clearly not suitable for the final approach. Servers of finer granularity are required on the final approach. For the purpose of Phase I demonstration, 0.5-nmi servers are used on the final approach. Approximate separation requirements are created by rounding off to the nearest 0.5-nmi separation value and also by preserving the qualitative trend of separation requirement as shown in Table 3. The idea is to create servers with length equal to the highest common factor (0.5 nmi) of the separation required between various combinations of aircraft pairs. To enforce separation of 2.5 nmi, the single server occupancy constraint in the queuing model can be enforced over the next 5 servers. In this manner, the separation requirement in terms of number of servers is shown in Table 4.

<table>
<thead>
<tr>
<th>VMC</th>
<th>Leading</th>
<th>Heavy</th>
<th>B757</th>
<th>Large</th>
<th>Small</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trailing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heavy</td>
<td>2.5</td>
<td>2.5</td>
<td>1.5</td>
<td>1.5</td>
<td></td>
</tr>
<tr>
<td>B757</td>
<td>3.0</td>
<td>2.5</td>
<td>1.5</td>
<td>1.5</td>
<td></td>
</tr>
<tr>
<td>Large</td>
<td>3.5</td>
<td>3.0</td>
<td>2.0</td>
<td>1.5</td>
<td></td>
</tr>
<tr>
<td>Small</td>
<td>4.0</td>
<td>3.5</td>
<td>2.5</td>
<td>1.5</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Inter-Aircraft Separation (nmi) by Aircraft Performance Class: Quantized Values to be Integral Multiples of 0.5 nmi

<table>
<thead>
<tr>
<th>VMC</th>
<th>Leading</th>
<th>Heavy</th>
<th>B757</th>
<th>Large</th>
<th>Small</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trailing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heavy</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>B757</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Large</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Small</td>
<td>8</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Inter-Aircraft Separation by Aircraft Performance Class: Number of Servers
2. Queuing Simulation Results

The following changes have been made to the queuing simulation to accommodate the variable inter-aircraft separation:

1. The final approach of the queuing network is discretized into 0.5-nmi servers.
2. The total number of servers increases from 53 to 63. Only the last two 3-nmi servers that are downstream to the last merge point on the final approach are discretized into 0.5-nmi servers.
3. The server sequences for arrival flights are changed to include the extra number of servers in their arrival routes.
4. An extra field is introduced into the simulation to track the aircraft class.
5. Conflicts in the terminal area are defined as separation violation of 3 nmi which requires 1 server separation. However, on the final approach conflicts are defined as per Table 4. Therefore, a small aircraft is registered in a state of internal conflict if there is a heavy aircraft within 8 servers ahead.
6. The delay computation remains the same.
7. The 89 flight arrival sequence used for the terminal area simulation (Section V.B) results is used.
8. Nominal wind model data is used in this simulation.
9. A random sequence of aircraft class is created with equal probability for all classes. This represents a homogeneous fleet mix of all aircraft classes.

Total delay using 3-nmi servers for the same traffic was found to be 32.75 minutes. The number decreases to 29.05 minutes using the variable server separation. Therefore, the variable server separation results in a total delay reduction of 3.7 minutes. Figure 27 shows the difference in landing times obtained from the simulation using 3-nmi servers and the simulation using both 3-nmi and 0.5-nmi servers. It can be seen that all flights except one arrives earlier in the latter simulation. The one flight that arrives late is a small aircraft trailing a heavy aircraft which requires 4-nmi separation.

![Figure 27. Difference in Landing Times Obtained using 3-nmi Servers and 0.5-nmi Servers on the Final Approach](image)

VI. Summary & Concluding Remarks

A. Summary

The work under this Phase I SBIR effort deals with modeling of terminal area operations using a queuing framework. It can serve in the following forms: (i) as a design tool for terminal area routes and terminal operations, (ii) as an analysis tool for NextGen terminal area concepts, and (iii) as a real-time decision support tool. The approach is based on a network constructed from published terminal area routes such as STARs, DPs and IAPs. The routes are discretized into smaller servers to enforce separation requirements. Arrival flight routes from metering fix to the landing runway are characterized in terms of finite number of server sequences referred to as arrival pathways. Each flight is assigned a server sequence that is dependent on the direction in which the flight enters the terminal area. Flights are further characterized by their scheduled time of arrival at the entry server and a desired airspeed.
profile. A discrete-event queuing simulation framework is developed for the propagation of flights over different servers while satisfying the mandated inter-aircraft separation requirements using the queuing model abstraction. The discrete-event simulation detects conflicts and computes the minimum delay required at each server to maintain separation. The delay represents an abstraction of controller action such as path-stretching or speed reduction.

The DEQS framework is tested in multiple case-studies formulated over the SFO terminal airspace. As a simple initial test, the queuing simulation is used to verify a known terminal area phenomenon known as compression while using a single pathway. Another case study involving all terminal routes studied conflicts and delays for 100 arrival flights over a duration of 4.5 hours. The queuing simulation is also used to study different scenarios of runway reconfiguration at the SFO airport. Two different schemes of runway assignment for arrivals are evaluated using the queuing simulation. The runway reconfiguration simulation is shown to capture phenomena involving flights arriving on the runway before the configuration is active or arriving late after the configuration has been made inactive. An assignment scheme that was based on the queuing simulation was shown to avoid these problems. The queuing simulation was also used to study the effect of variable inter-aircraft separations that are functions of aircraft performance class.

B. Unique Contributions
The following are some of the unique contributions of the current work:
1. Creation of 3-nmi servers to enforce mandated inter-aircraft separation in terminal area.
2. Creation of a discrete-event queuing simulation framework that is suitable for fast-time stochastic evaluation of terminal area operations.

C. Conclusions
The queuing simulation developed under this work is a computationally efficient apparatus for predicting arrival times, conflicts, and delays in terminal area. The simulation currently executes in less than 4 seconds on MATLAB. Therefore, it has very good potential for serving as a real-time decision support tool that is based on stochastic evaluation of terminal area traffic. The queuing simulation can be used to study and design a variety of NextGen terminal area concepts such as optimal planners, very closely spaced parallel runway operations, continuous descent arrivals, metroplex route design and airspace design.

Acknowledgments

This research was supported under NASA Ames Research Center Contract NNX10CC16P. The authors would like to thank the technical monitor, Mr. John E. Robinson, and the following researchers at the NASA Ames Research Center: Ms. Savita Verma, Mr. Daniel Mulfinger, Ms. Jane Thipphavong, Mr. Douglas R. Isaacson, Mr. Harry Swenson, Dr. Todd Callantine and Dr. Seongim Choi, for their review and constructive feedback which enhanced the quality of this work. The authors sincerely thank Mr. Michael C. McCarron, Director of Community Affairs, San Francisco International Airport (SFO) and Mr. Bert Ganong, Manager of the Aircraft Noise Abatement Office, SFO, for providing the Radar track data for arrivals and departures into SFO recorded by the FAA Automated Radar Terminal System (ARTS). The Radar track data provided realistic data and observed flight behaviors in the SFO terminal area and greatly improved the fidelity of the developed queuing models and the simulations performed during the course of this research. The authors are also grateful to Mr. Fred G. Bollman, Traffic Management Coordinator, En Route and Oceanic Operations Western Area Office, Oakland ARTCC, for providing the authors with the insight into SFO terminal area operations from an air-traffic controller’s perspective.

References


AirNav Website http://www.airnav.com/airport/KDFW

National Geospatial-Intelligence Agency’s (NGA) Digital Aeronautical Flight Information Files (DAFIF) https://www1.nga.mil/ProductsServices/Aeronautical/DigitalAeroFlightInfoFiles/Pages/default.aspx

SFO WestPlan and South-East Plan Configurations: http://www.flysfo.com/web/page/about/news/pressres/

