Development of Nation Wide Cost-Benefit Analysis Framework for Aviation Decision Making Using Transportation Systems Analysis Model

Yue Xu

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Civil Engineering

Dr. Antonio A. Trani, Co-Chair Dr. Hojong Baik, Co-Chair
Dr. C.Patrick Koelling
Dr. Gerardo W. Flintsch

March 28, 2008
Blacksburg, Virginia Tech

Keywords: Cost-Benefit Analysis, Transportation Appraisal, Transportation Systems Analysis Model, Lower Land Minima, Data Communication

Development of Nation Wide Cost-Benefit Analysis Framework for Aviation Decision Making Using Transportation Systems Analysis Model

Yue Xu

(ABSTRACT)

The aim of this study is to establish a nation-wide cost-benefit framework for aviation projection appraisal. This framework is built upon Transportation System Analysis Model developed at Virginia Tech Air Transportation System Model (TSAM). Both supply and demand characteristics and their inter-dependence are investigated. It attempts to solve the absence of supply constraints in aviation demand forecast in the literature. In addition, external costs in term of noise and emission are also considered. A national environmental impact analysis introduced by new generation small aircraft system is conducted.

Two case studies are discussed to illustrate the framework. The first one is based on the GPS Wide Area Augmentation System (WAAS) Lower Landing Minima capability. It represents a nation-wide cost-benefit analysis with examination of both supply and demand. System-wide benefit of accessibility improvement and infrastructure cost are scrutinized at the same time. A prioritized set of candidate airports for this technology is provided as a result.

The second study focuses on New York area. Benefits brought by DataComm technology are evaluated by multi-iteration simulations. DataComm is projected to reduce entry point intrail and final approach separation. The improvements are modeled at individual airport and New York airspace. Consumer surplus is estimated based on demand and delay relationship using TSAM.

ATTRIBUTION

Several colleagues and coworkers aided in the writing and research behind several of the chapters of this dissertation. A brief description of their background and their contributions are included here.

Prof. Antonio A. Trani - Ph.D. (Department of Civil Engineering, Virginia Tech) is the primary Advisor and Committee Chair. Prof. Trani provided enormous guidance toward all the chapters of this dissertation. Furthermore, Prof. Trani helped the acquisition and installation of the necessary software and provided technical guidance.

Prof. Hojong Baik - Ph.D. (Department of Civil Engineering, Missouri University of Science and Technology) is the advisor and Committee Co-Chair. Prof. Baik's insight on aviation operations, environmental impacts and simulation has been a valuable asset during the completion of this dissertation.

Prof. Gerardo W. Flintsch - Ph.D. (Department of Civil Engineering, Virginia Tech) is the Committee Member. Prof. Flintsch provided extensive advices on the subject of cost-benefit analysis, maintenance cost, interest rate and discount rate. In addition, he also helped the writing and organization of this dissertation.

Prof. C.Patrick Koelling – Ph. D. (Department of Industrial and Systems Engineering, Virginia Tech) is the Committee Member. Prof. Koelling discussed with the author on numerous subjects such as lower landing minima analysis, safety concerns, and cost-benefit analysis components. His comments during the completion of this dissertation were very helpful.

In addition, I'd like to acknowledge the contribution of my colleagues in the following chapters:

Chapter 4: Lower Landing Minima Analysis

Senanu Ashiabor - Ph.D. (Department of Civil Engineering, Virginia Tech) currently at Dowling Associates Corporation was a student in the author's group and contributed during his graduate studies to this chapter in terms of mode choice modeling.

Nicolas K. Hinze – MS. (Department of Civil Engineering, Virginia Tech) is a research associate at Virginia Tech. He worked in collaboration with the author on executing TSAM model and environmental impact integration into TSAM.

Chapter 5: External Cost Analysis

Howard Swingle – currently a senior research associates at Virginia Tech. His airport survey offered detailed input to the Blacksburg Airport noise case study.

Chapter 6: DataComm Analysis

Gabriele Enea - MS. (Department of Civil Engineering, Virginia Tech) currently at L-3 Communications. He worked closely with the author on the DataComm projects. His analysis on the demand scenarios provided inputs to the SIMMOD simulation.

Donghyeok Sohn - MS. (Department of Civil Engineering, Virginia Tech) currently a Ph.D. candidate at Virginia Tech. He worked closely with the author on the DataComm projects as well. The PDARS data he analyzed was used to build basic airspace structure of the SIMMOD simulation.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deep and sincere gratitide to my advisors Dr. Trani and Dr. Baik for their consistent support and guidance throughout my pursuant toward this degree. I have benefited immensely from the depths of their technical knowledge. I also want to thank the members of my committee who have helped in various ways. Dr. Koelling and Dr. Flintsch provided direction with very insightful comments during my Ph.D. progress.

I am thankful for the comments and encouragement of all the team members of the Virginia Tech Air Transportation Systems Laboratory. I also want to thank Stuart Cooke, Jeff Viken (NASA Langley Research Center), Sam Dollyhigh and Jeremy Smith (Swales Aerospace) who at various times offered constructive criticism and suggestions of the cargo and environment impact model that is implemented in TSAM.

Several of my friends and colleagues have provided a lot of support and advice during time I have worked on this dissertation. My sincere thanks goes out you all, and though I cannot mention everyone by name I especially want to thank my fellow Chinese friends Heng Wang, Yu Gao, Yang Lu, Yu Zhou, Tian Gao, Pengfei Li and Ni Shen.

Last but of course not least, this work would not have been achieved without the support and understanding of my family. They have always supported me in every way I have chosen in life.

Table of Content:

List of Figure and Tables:

Figures:

Tables:

Chapter 1 : Introduction

Background and Motivation for Research

Cost-Benefit Analysis (CBA) has been widely applied in decision making procedures of public investments. The Federal Aviation Administration (FAA) recommends CBA in all aviation investment proposals.

In the literature, civil aviation CBA analysis is usually applied at the regional level. The FAA Terminal Area Forecast (TAF) is often cited for demand projection. Consequently the credibility of such analysis greatly depends on the accuracy of TAF. However, TAF demand forecast for small to medium airports are questionable which makes the economic study at these airports difficult.

Due to the absence of a comprehensive national demand forecast model, systematic policy and technology impacts cannot be evaluated efficiently. Therefore, NASA, FAA and other federal agencies are developing models to estimate demand of new technology and policies. The Virginia Tech Air Transportation Systems Lab, under a contract with NASA, has developed a Transportation Systems Analysis Model (TSAM) to assess the impacts of the Small Aircraft Transportation System (SATS) and other aerospace technologies in the National Airspace System (NAS) (Trani, 2003). The model estimates nationwide, multi-mode intercity trips at the county level for multiple income groups from years 2005 to 2020. At the core of the model is the use of a mode choice model where passenger mode choice behaviors are based on utilities of each travel alternative.

Based on the TSAM model, policy and technology impacts can be evaluated by changes in utility function of the affected mode and passengers selecting each mode accordingly. This capability makes nationwide life-cycle cost-benefit analysis feasible by estimating system-wide costs and benefits. Compared with other national models, this method has the following advantages:

- Demand is estimated for a large number of airports (up to 3,000 including GA airports)

- Demand is sensitive to policy and technology change
- Economic impacts are evaluated systematically
- Environmental impacts are considered

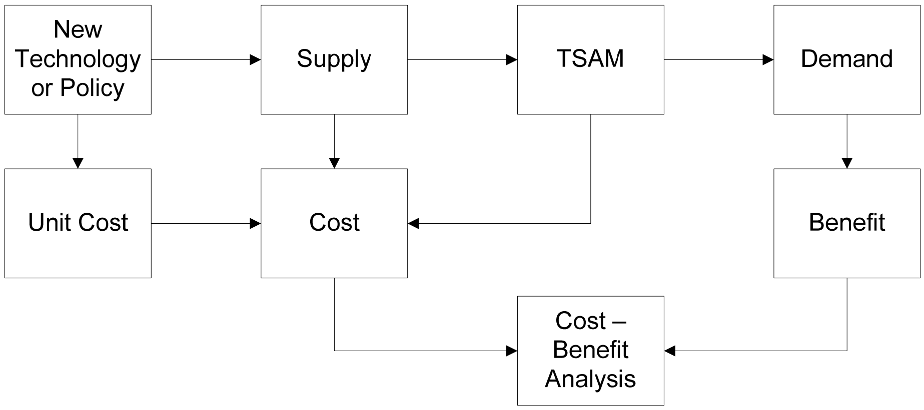

Figure 1-1: Cost-Benefit Analysis Framework Flowchart.

A national cost-benefit analysis framework is proposed. **The uniqueness of this model is the usage of utilities based, multi-mode traveler behavior model at the county level to conduct CBA analysis to assess technology and policy impacts. This model can estimate important cost benefit components including demand, travel time savings, delay and external cost.** Two case studies are discussed thereafter to illustrate the application of this framework. The first study addresses economic aspects of the Wide Area Augmentation System (WAAS) Lower Landing Minima (LLM) capacity using TSAM. The second study elaborates TSAM's application on Data Communication program (DataComm).This dissertation is organized as follows: chapter 1 provides a brief introduction of the background and motivation, chapter 2 includes a review of recent research on the topic, chapter 3 discusses proposed methodology, chapter 4, 5, and 6 focuses on detailed modeling of the LLM, environmental and DataComm analysis and chapter 7 presents conclusion and recommendations.

Chapter 2 : Literature Review

Review of Aviation Policy Decision Support Models

In the literature, simulation and analytical techniques have received extensive discussion as aviation policy decision support models.

Simulation based approaches use detailed aviation facility characteristics and entity behaviors are reproduced to mimic real operation conditions. Policy or technique changes are usually modeled by modifying entity characteristics, entity behavior or environment. The scope of simulation models ranges from macroscopic models such as the NAS Strategy Simulator (NSS) to large-scale microscopic models such as the Airport and Airspace Delay Simulation Model (SIMMOD), the Total Airspace and Airport Modeler (TAAM), and the Reorganized ATC Mathematical Simulator (RAMS). Macroscopic models simulate policy impacts at aggregated entity level with simplified assumptions and rules. Microscopic models trace the behavior of individual entity and simulate interactions among entities. Recently, a family of agent-based simulation model has received more attention (Niedringhaus, 2004), (Schaefer, 2002), (Lee, 2001), (Lewe 2003, 2005). The Agent-Based Modeling (ABM) represents the bottom-up modeling technique that builds an entity (agent) and simulates its interaction with the environment and other entities. Representative ABM models include Transportation Architecture Field (TAF) by Georgia Tech, Jet:Wise by MITRE Inc and Airspace Concepts Evaluation System (ACES) by FAA. The advantage of simulation models is the ability to imitate the behavior of each system components and their interactions. However, for complex systems, this method may need large amount of data input and computation.

Analytical models apply mathematical equations with a set of parameters. This type of models usually requires less inputs and computation. Modeling techniques such as regression analysis, time-series analysis, demand-supply curve, linear and nonlinear programming are applied. The characteristics of system components are aggregated into parameters, equations and curves. The Terminal Area Forecast by FAA, Air Carrier Investment Model by FAA and Logistics Management Institute (LMI) and Transportation System Analysis Model (TSAM) by Virginia Tech belong to this family. The advantage of analytical model is simplicity and efficiency. They usually require less detailed information and shorter computation time. But as the complexity of the system increases, analytical systems can become very difficult to solve.

The following section describes several nation-wide aviation decision support models in more detail.

Terminal Area Forecast (TAF) by FAA

The Terminal Area Forecast (TAF), produced by the Office of Aviation Policy and Plans (APO) each year, records and estimates operations covering airports in the National Plan of Integrated Airport Systems (NPIAS). It contains both historical and forecast data by facility, state, or region. It includes 266 FAA towered airports, 227 Federal contract tower airports, 31 terminal radar approach control facilities, and 2,987 non-FAA airports as of September 30, 2004 (FAA, 2006). The model can be queried online or downloaded to a computer (FAA, 2006).

The forecast is made at the individual airport assuming demand is unconstrained. This assumption implies the airport is able to absorb any demand feed without causing delays. Historical data for FAA Air Route Traffic Control Centers (ARTCC) and contract towered airports come from the FAA's Air Traffic Activity Data System (ATADS) (FAA, 2001). Forecasts at these airports are developed using relationships between airport passenger demand and/or activity measures as well as local and national factors influencing aviation activity. Other forecast techniques such as regression analysis and growth rate may be employed if the forecast deviates from the expected trend. In addition, the 35 OEP (Operational Evolution Plan) airports and 8 secondary commercial service airports in the large hub cities receive a more detailed analysis. Other factors such as local economics, growth of originating and connecting traffic, price, seating capacity and load factors are also considered. For non-towered airports, TAF data is based on estimates filed with FAA Airports District Offices using the FAA Form 5010. Operation levels at these airports are held constant unless otherwise specified by a local or regional FAA official.

TAF has been used by the FAA to meet its planning, budgeting, and staffing requirements. However, TAF has major limitations. Since TAF uses an its unconstrained demand assumption, TAF estimates at small to medium airports are unable to reflect future trends of local traffic. Moreover, new initiatives such as the Small Aircraft Transportation System (SATS), which emphasizes in utilizing small aircraft and airports for on-demand operations, cannot be properly evaluated by TAF.

Logistics Management Institute Network Model (LMINET)

The LMINET has been developed by the Logistics Management Institute (LMI) and FAA. LMINET employs queuing theory as its modeling principal. Each airport represents a server in the network. The model simulates flights by linking each airport with sequences of queuing models of the Terminal Radar Approach Control (TRACON) and Air Route Traffic Control Center (ARTCC) sectors.

At the core of the LMINET model is the airport and airspace capacity and delay model. Presently, capacities and delays models are available at 64 airports. These airports account for 85% of domestic commercial operations and more than 80% of air carrier operations (Long, 2001). A couple of hundreds airspace delay models are constructed as well. An advantage of this model is that it attempts to capture the stochastic nature of airport operations. Factors such as speed, position, wind and communication delays are expressed as random variables. Impacts of new Air Traffic Management (ATM) systems can be reflected in the changes to the distribution of random variables and their parameters.

The LMINET was used in the evaluation of the Terminal Area Productivity Program (TAP) (Lee, 1997), air traffic demand and delay estimation (Kostiuk, 2000), and in the initial projection of SATS demands at airports (Long, 2001). In the TAP analysis, program benefits were evaluated by changes in random variables. In the demand and delay estimation, ETMS data was used to derive GA schedules and TAF growth rates to feed into a FRATAR model. To the center of our interest is the SATS demand modeling where a few limitations are found.

1) The model assumes SATS is used to pick up unsatisfied commercial traffic. The assumption neglects the possibility that SATS would attract demand from other modes such as auto, GA and even current commercial airline passengers instead of unsatisfied passengers.
2) TAF is used to forecast unconstrained demand scenario. As we discussed above, TAF forecasts are questionable for small, underutilized airports where SATS is likely to be deployed.
3) Aviation System Analysis Capacity (ASAC) Air Carrier Investment Model (ACIM) is used to forecast constrained demand. This model has some limitations as well which will be discussed later.
4) Utilities of SATS (travel time and cost) are not considered and hence no traveler decision behavior is modeled.
5) TAF airports excluding military airports, TRACONs and 64 airports are used as SATS airports regardless of SATS aircraft performance. Some of these airports are not SATS compatible for a variety of reasons including insufficient runway length, width, unpaved runway surface, etc.

The model is capable of assessing delay and schedule impacts of new technology and systems if demand is provided. However, the absence of demand estimation capacity prevents LMINET from being applied to evaluate impacts of new travel modes such as SATS. There is no concrete demand model built in the LMINET.

FAA NAS Strategy Simulator (NSS)

The FAA NAS Strategy Simulator (NSS) is a macro level dynamic simulation model. It was developed by FAA, Ventana Systems and NEXTOR schools. The model is programmed using the simulation software Vensim for each years spanning from 1970 to 2025. It is designed to support the FAA decision making process by evaluating macro impacts of new policies and system changes. The model simulates three components of the NAS system: passengers and shippers, fleets of aircraft and their operators, and the system of airports and air traffic control (ATC). The three components are further broken down into modules that represent certain characteristics of the system.

The NSS estimates demand in response to tax structure and system changes. The model serves eight user groups:

1) Mainline domestic passenger service
2) Regional Commuter domestic passenger service
3) Very-light jet on-demand domestic passenger service
4) Domestic cargo-only service
5) International cargo-only service
6) International passenger service
7) General Aviation flights using jet fuel (turboprops and turbojets)
8) General Aviation flights using aviation gasoline (piston aircraft)

Each user group uses different methodology to estimate demand and supply. Demand of the first three groups is computed via demand and supply curves that depend on the Gross Domestic Product (GDP), delays, airline costs, taxes, and airline fleet growth. Domestic cargo demand responses to GDP, fuel price, and taxes. Belly cargos carried by commercial airlines are considered as well. International demand and GA flights are determined by fuel prices and taxes only. Model parameters were calibrated to match the FAA APO's forecast and resulted in a set of parameter values.

The outputs of the model include delay estimation and impact analysis, FAA operating cost, and AATF revenues.

The demand estimation of NSS depends on the accuracy of demand and supply curves. Part of the model results matches well with APO's forecast but some do not.

Large-Scale Discrete Event Simulation Models

There are three popular large-scale discrete simulation models: The Airport and Airspace Delay Simulation Model (SIMMOD) developed by the FAA, the Total Airspace and Airport Modeler (TAAM) developed by Preston Aviation Solutions Pty Ltd, a subsidiary of the Boeing Company, and the Reorganized ATC Mathematical Simulator (RAMS) developed by Eurocontrol.

These models have the same target: estimate airport and airspace utilization and performance. Each aircraft in these models is treated as an entity competing for resources as it progresses through the NAS. Movements of the aircraft are represented using simple kinematics rules and follow some sort of link-node structure. Airport and airspace resources are modeled as objects such as runways, gates, and fixes. System performances are evaluated by conflicts and delays. Typically capacity is not explicitly measured but can be reflected from delay.

The SIMMOD simulates both airport and airspace simulations. It traces aircraft movement from gate to gate and has been validated from 1985 to 1991 (Trani, 2003). Similarly, the TAAM is also an airport and airspace simulation model and uses gate to gate simulation concept as well. The RAMS, however, only simulates airspace operations. It employs good conflict detection and resolution logic.

All the three models are complex simulation models and thus demand highly detailed information and considerable learning. They are suited to evaluate impacts at terminal and en-route areas introduced by certain level of traffic or alternate flow management strategies. They do not estimate anticipated demand as a result of new technology and policy.

National Airspace System Performance Analysis Capability (NASPAC)

The National Airspace System Performance Analysis Capability (NASPAC) Model is also a discrete event simulation model. It is the first model that evaluates propagation of delays and congestion through a national or regional ATM system (Odoni, 1996). However, it is not as popular as the above-mentioned models. The FAA has no further plan to enhance or expand this capability.

On the contrary, several revisions of this model are taking place outside the FAA. European counterpart F-NASPAC is being developed by CENA in France. MITRE, the original creator of the model under a contract with the FAA, is developing simplified models such as Quickpac, Detailed Policy Assessment Tool (DPAT) and AMC (the Aggregate Modeling Capability).

The model requires input including: 1) Demand in term of schedules of aircraft itineraries; 2) Capacities of airport and other ATM resources; 3) Aircraft performance parameters. It estimates delays at point of interest in term of "technical delay" (defined as the local delay incurred at any specific point in the system) and of "effective delay" (defined as the difference between unimpeded and actual travel time).

As other large-scale simulation models, learning curve of this model is high. It does not estimate demand as a result of new technology and policy either.

Airspace Concepts Evaluation System (ACES)

The Airspace Concepts Evaluation System (ACES) is a large-scale, fast-time, agent-based simulation model. The model was developed for the NASA Virtual Airspace Modeling and Simulation Project. The purpose is to assess impacts of new operational and technological changes. It utilizes the High Level Architecture (HLA), a set of middleware, tools and process that are developed by Department of Defense, to integrate each individual simulation package. The simulated agents that represent present NAS operation components include Air Traffic Control System Command Center (ATCSCC), En route Air Route Traffic Control Center (ARTCC), Terminal Radar Approach Control (TRACON), Airport Control Tower (ATCT), aircraft, and pilot entities (Hunter, 2005).

Each agent is defined by three properties: behavior logic, environment and interactions. First of all, the agents are autonomous entities with unique characteristics and behave accordingly. Secondly, they are influenced or constrained by their environment. Finally, they interact and exchange information with other agents and adjust their behaviors according to the new information.

In the ACES, aircraft trajectory serves as links between each agent. Each agent in turn performs predefined actions on flight trajectories. They do not necessarily share the same information as none may know the true trajectory. Hence ACES employs a multi-trajectory modeling concept to simulate real world conditions. The level of accuracy of each agent’s action on flight trajectory depends on the technology capabilities of the agent. In this way, NAS new operational concept can be simulated based on each agent’s capability parameter.

The ACES was validated using FAA Aviation System Performance Metrics (ASPM) historical data. The results show the model is capable of replicating the historical conditions satisfactorily (Odoni, 1996).

The primary focus of the ACES is to simulate effects of new initiatives on the NAS including ATM, controller and pilot. It does not employ the full capability to assess effects on passengers such as demand.

AvDemand and AvAnalyst

AvDemand and AvAnalyst is a set of tools developed by the Seagull Technology Inc to model future NAS demand, delay and economic impacts (Hunter, 2005). It uses TAF forecast as demand input and applies FRATAR model to distribute demand at each O-D pair. Then the output was fed into ACES to simulate system wide operations and delays. The model attempted to use spatially-shifted schedule (utilize smaller airports around hubs) and temporally-shifted schedule (reschedule departure time to off-peak hours) to accommodate growing demand. Impacts of Extremely Short Takeoff and Landing (ESTOL) were modeled using the predicted O-D pair operations and schedule. Finally an economic layer was suggested to conduct benefit analysis. Methodology and result of this layer is not available in detail in the literature.

The Seagull model performs as a schedule prediction tool to distribute TAF forecast to each O-D pair and evaluate delay reduction strategy. It does not build real demand generation.

Agent-Based Simulation Models (ABM/S)

Agent-Based Simulation Models (ABM/S) represents a family of bottom-up simulation approaches to model complex systems. The building block of ABM/S system is the adaptive and autonomous agent that interacts with the environments and other agents to achieve certain objectives (Wooldridge, 1995), (Holland, 1995).

ABM/S is especially suited to model complex system such as NAS. Georgia Institute of Technology is developing a multiple agent-based simulation prototype named Transportation Architecture Field (TAF). It constitutes of four basic entity groups: resources, stakeholders, drivers, and disruptors. The resource entities represent physical components such as vehicle and infrastructure. The stakeholders reside in both private and public sectors including consumers, service providers, insurers, regulatory agencies, infrastructure providers, etc. The diver entities influence the stakeholders implicitly in terms of economic, societal, and psychological situations. On the other hand, the disruptor entities affect the performance of the resource network explicitly by reducing the efficiency or connectivity of the network. Each entity embraces a set of attributes, functions, interfaces, and sentience with a flexible boundary.

To formulate TAF, an abstractive geographic unit 'locale' is introduced as a basic building block where agents reside. Each locale consists of four entity groups that are treated as global modules. The locale is further interconnected and interacts with other locales. As the simulation progress, the collective behavior of the network can be fed back to the global modules, which in turn changes the locales. This retrospective procedure builds the basic agent-based simulation model of NAS.

The model is calibrated against 1995 American Travel Survey (ATS) and the result is satisfactory (Lewe, 2005). Two case studies are conducted. The first study evaluates the Personal Air Vehicle (PAV) and the Small Aircraft Transportation System (SATS) concepts proposed by NASA. The second case study concentrates on the roadability and vertical takeoff capability of PAV system. It is not clearly stated in the literature whether the outputs have been verified.

The MITRE Corporation's Center for Advanced Aviation System Development (CAASD) is developing an agent-based model named Jet:Wise (Niedringhaus, 2004). The inputs of Jet:Wise include airline information (number of seats, schedule, frequency and fare), airport capacity, and passenger trip purposes. After hundreds or thousands iterations, the model produces outputs such as airline's fleet, schedule and fares. Network performance parameters such as passenger delay, load factors, and revenue passenger miles can be obtained from the output as well. The core of Jet:Wise is the agent-based learning module where agents learn from experience and adjust their behaviors to make the optimum decisions. The learning procedure is executed by applying seven continuous and discontinuous learning tools. Continuous tools are used to approximate a real-valued parameter such as fare whereas discontinuous tools apply when discrete actions such as buying N aircraft are needed.

The model only concentrates on metropolitan areas instead of the whole NAS. It has been verified for leisure passengers but not for business passengers yet.

The agent-based models possess unique advantages in modeling complex systems such as NAS. The impact introduced by new policies can be modeled by modifying agent characteristics, objectives, rules to interact with others or environment parameters. The shortcoming of agent-based models is that it needs highly simplified agents, well-defined rules and relatively large computation time.

Review of Aviation Cost/Benefit Models

Implementation of new policy or technology is only likely to be realized if substantial benefits can be expected. Meanwhile, there will also be costs that inevitably accrue during implementation and after. This requires careful comparison of anticipated benefits against projected costs. In addition, costs consist of non-recurring investment during development and recurring maintenance during operation. Therefore a life-cycle cost benefit model which evaluates costs and benefits quantitatively is essential to evaluate such developments.

Crucial as life-cycle cost benefit analysis is, there has been only minimal general-purpose capability currently available (Odoni, 1997). Two economics models, developed by NASA and FAA respectively, are identified in the field.

Air Carrier Investment Model (ACIM)

In order to identify and evaluate promising technologies, NASA is building an Aviation System Analysis Capacity (ASAC) in conjunction with the LMI. It differs from previous NASA modeling efforts in that 'the economic behavior of buyers and sellers in the air transportation and aviation industries is central to its conception' (Wingrove, 1997). The model employs high level economics parameters to predict future demand and airline costs. The outputs include the Revenue Passenger Miles (RPM), operating margin for air carriers, and size of the total U.S. scheduled passenger air carrier fleet in numbers of aircraft. The model was programmed using an Excel spreadsheet. A schematic view of the model is presented below.

The model identified 85 key US airports serviced by 13 major air carriers. The airport-specific demand of each carrier was estimated using regression analysis. Explanatory variables included the air carrier's own fare, competitor's fare, per capita income, population, and unemployment rate. Total demand for a specific carrier was then constructed by summing the airport-specific demand.

The second major component of the model, air carrier operating cost analysis, was based primarily on the Department of Transportation (DOT) Form 41 data. The cost function is based on output quantities, factor prices, aircraft attributes, and network traits. Then a family of translog cost equations was derived for each of the carrier's factors of production including labor, energy, materials and capital. The translog cost equation can be explained as 'a second-order approximation of the cost function dual to a generic production function' (Wingrove, 1998). Policy and technology impacts then can be reflected from changes in factors of production. Aircraft attributes are modeled from various characteristics of the aircraft fleet. Air carrier's network is brought into the model by using two parameters: average stage length and passenger load factor.

General steps to evaluate scenarios are as follows:

1) Forecast RPM based on economic characteristics and demand projection.
2) Estimate airline revenue based on RPM and future fare assumptions
3) Predict airline operating cost based on RPM, changes in input prices, and changes in aircraft and network characteristics
4) Predict the aircraft inventory and airline employment based on airline operating costs, a capital share equation, fare assumption and aircraft size.
5) Validate results with other source of data

This model has reached certain level of maturity and has been applied to cargo operations (Johnson, 1999). It is essentially an econometrics model that is capable of predicting demand, supply and operating cost. However, it has the following limitations:

1) It only represents a portion of NAS network and users
2) The user needs to input future economic supply and demand characteristics such as fare and the model projects the airline and aircraft industry economic situation accordingly.

3) It replicates historical data and utilizes simple regression function. Future projection primarily depends on a continuance of these economics conditions (MIT, 1996).
4) The model is designed to produce specific outputs from specific inputs. It lacks the flexibility to evaluate other input variables.

Furthermore, it caters for airline and aircraft manufacturing industry but not administrative agencies such as NASA and FAA. The model primarily evaluates benefits of new policy and technology that targets at improving efficiency and productivity of new aircraft.

National Airspace Resource Investment Model (NARIM)

The National Airspace Resource Investment Model (NARIM) is being developed jointly by the Investment Analysis and Operations Research Directorate (ASD-400) which supports the FAA's Office of System Architecture and Investment Analysis (ASD) and NASA Interagency Integrated Product Team (IPT) for Air Traffic Management (ATM). The model is developed to 'provide an analysis framework that enables the assessment of the operational, investment and architectural implications of new operational concepts from the perspectives of the integrated aviation community" (Bradford, 1996).

The NARIM incorporates three components (FAA, 2004):

1) Operational Modeling: models movement of the aircraft
2) Architecture/Technical Modeling: evaluates impacts of procedural / system changes on NAS infrastructure.
3) Investment Analysis Modeling: provides methodology to trade between alternatives using cost benefit analysis

The prototype of NARIM has been completed. The Enhanced Traffic Management Systems (ETMS) was integrated into the model recently. FAA is currently expanding model's capabilities to evaluate NASA and FAA interagency IPT for ATM.

This model was exercised to evaluate the following scenarios:

1) Controller conflict resolution baseline study
2) Nation-wide conflict resolution benefits study

3) Cockpit Display of Traffic Information (CDTI) requirements analysis and benefits study

This model is still in the stage of development and hence its effectiveness and validity cannot be sufficiently evaluated. The model intends to be an integration of existing models and new models. FAA tries to take maximum advantage of existing models and make proper modifications if necessary.

Limitations of National Aviation Decision Making Models

Majority of the discussed national aviation DSM models are simulation models. These models are capable of simulating influences of future traffic scenarios but do not estimate demand directly. Two models, TAF and NSS, employ simple regression method to predict demand. Other models either require user demand input (TAAM, RAMS, SIMMOD, and NASPAC) or use TAF (LMINET and AvDemand). No model was found to have the capability to estimate national demand as a result of travel time and cost change. As we will discuss in the following section, the most important user benefits of a transportation project is travel time savings. None of the above mentioned models predict demand based on mode utility, hence makes it difficult to conduct proper cost-benefit analysis. Table 2-1 summarizes the capability and limitation of these models.

Table 2-1: Capability and Limitations of National Decision Support Models.

Model	Capability	Limitation
Terminal Area Forecast (TAF)	Records and estimates operations covering airports in the National Plan of Integrated Airport Systems (NPIAS).	Limited estimation accuracy at smaller airports.
LMINET	Employs queuing theory to estimate policy impacts using integrated airport capacity and delay model.	No demand generation scenario. Uses TAF as demand projection.
NAS Strategy Simulator	Macroscopic model that evaluate impacts of new	High level model that does not deal with detailed travel

	policies and system changes.	behavior.
TAAM, RAMS, and SIMMOD	Estimate airport and airspace utilization and performance.	Do not estimate demand directly. Requires very detailed knowledge on the airport or airspace. High learning curve.
NASPAC	Evaluates propagation of delays and congestion through a national or regional ATM system.	Complex model requiring large learning curve. No demand estimation capability.
ACES	Assess impacts of new operational and technological changes using agent-based simulation.	Does not employ the full capability to assess effects on passengers such as demand.
AvDemand / AvAnalyst	Models future NAS demand, delay and economic impacts.	Does not build real demand generation. TAF is used.
ACIM	Employs high level economics parameters to predict future demand and airline costs.	Represents portion of the NAS system. Lack of certain flexibility.
NARIM	Assesses the operational, investment and architectural implications of new operational concepts.	In the stage of development. Effectiveness and validity cannot be sufficiently evaluated.

Review of Aviation Cost and Benefit Estimation

User Benefit Estimation

Generally, there are three types of benefit due to the expansion of construction of transportation facilities:

Transportation System User Benefit: Quantitatively, transportation system user benefits are defined as 'the savings in vehicle operating costs, travel time value, accident costs, and fares that the users of improved highway facilities or transit systems will enjoy' (AASHTO, 1977). Qualitatively, users are benefited from improvement of accessibility, comfort and convenience, etc. In general, the qualitative terms are hard to model and convert to monetary terms explicitly but will be considered in the justification of the project.

Social Benefits: Social benefits describe the benefit not only enjoyed by the actual user of the facility but also beneficial to other non-users. The social benefits include reduced air pollution and noise level, and the economics growth of the adjacent region.

Transportation Agency Benefits: For transportation agencies, new facilities will improve riderships and revenue will increase if there are any charges associated with the new facility.

The following sections will concentrate on user benefits.

Transportation System User Benefit

Traditionally, the user benefits are modeled as follows (Williams, 1976):

$$User\ Benefits = \left(U_0 - U_1\right)\frac{\left(V_0 + V_1\right)}{2} \quad (2\text{-}1)$$

Where:

U_0 = the user cost per unit of traffic *before* the facility development

U_1 = the user cost per unit of traffic *after* the facility development

V_0 = the level of traffic *before* the facility development

V_1 = the level of traffic *after* the facility development

It is worth noting that reductions of users' perceived costs will often induce additional traffic or patronage in the absence of improvements. This cost includes both travel time and cost. So the level of traffic after the facility developments or improvements includes not only the existing traffic but the induced and diverted traffic. The induced or diverted traffic come from diversion from other travel modes or other routes of the same mode, longer trips and existing trips shifting from off-peak to peak conditions. The proper measures of benefits should consider induced and diverted traffic in addition to existing traffic.

Equation (1) is consistent with the theory of Consumer's Surplus. The consumer's surplus is defined as the benefit which a consumer enjoys, in excess of the costs which he or she perceives. The basic idea can be shown in Figure 2-1:

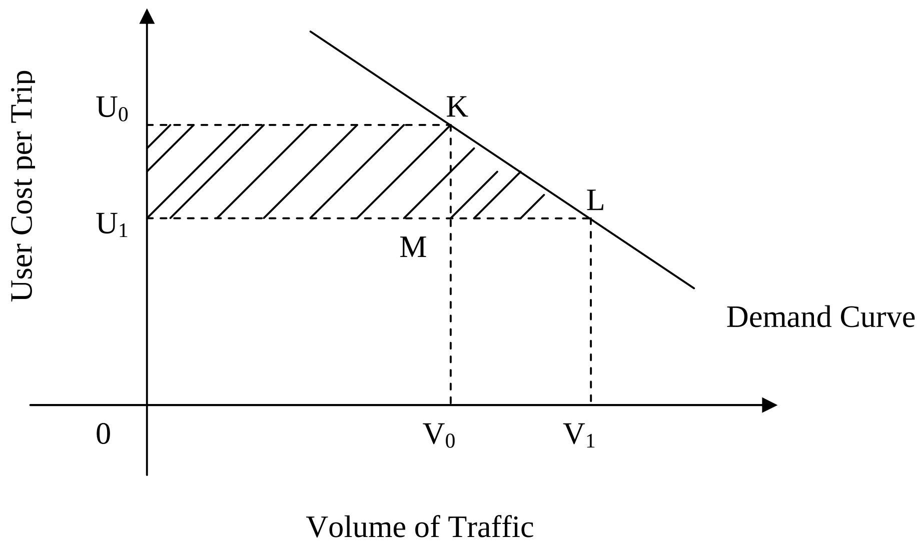

Figure 2-1: Consumer's Surplus.

The consumer's surplus at a given price is the total area above the price enclosed by the demand curve up to the point of its intersection with the price axis. In user benefit evaluations the benefit becomes the area enclosed by the user cost axis, the demand curve and two user cost lines (U_0KLU_1) in Figure 2-1, which is consistent with Equation 1. The area U_0KMU_1 represents the user benefit for existing users and KLM for induced or diverted users.

The amount of induced traffic is measured by the demand curve KL in Figure 2-1. The slope of demand curve (KL) is often referred to as elasticity of demand with respect to service cost. It represents the percentage change of demand in response to the percentage change in cost. The elasticity is always a negative value because the trip becomes more attractive when the cost decreases.

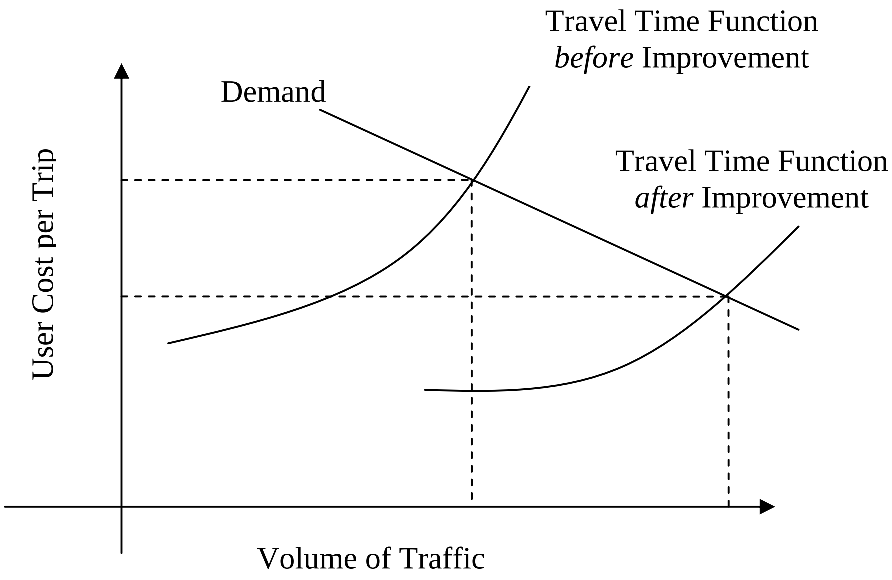

Figure 2-2: Estimating Induced Travel.

Figure 2-2 illustrates a method to estimate induced travel. Induced trips can be estimated under the appropriate estimation of elasticity and travel time function after improvements.

In addition to travel time and travel cost, infrastructure investments may anticipate improvements in safety, comfort level, macroscopic economics, dispatch reliability, and schedule predictability. These benefits are usually referred as hard-to-quantify benefits in the literature. Since the purpose of this dissertation is to develop a cost benefit analysis framework, these benefits will not be discussed in detailed. Once the framework is built, additional benefits can be integrated.

User Cost Estimation

User costs are primarily evaluated considering three aspects: 1) transport costs (automobile operation costs, transit fare, etc); 2) travel time and 3) other social costs. For the majority of infrastructure improvement projects, travel time savings constitute the most significant perceived benefits.

Travel time is ‘the time required to traverse a route between any two points of interest’ (TTI, 1998). Travel time savings refer to the value of time saved by reducing the total length of travel time. Travel time savings are the most widely accepted user benefits introduced by transportation facility constructions and expansions (Mackie, 2001). They are used as the primary justification of such development needs. Travel time can be broken down into several categories: actual travel time, waiting time, vehicle accessing time and time spent for parking. They constitute more than half of the all project benefits (Hobeika, 2003) and are quantified using the concept of the ‘Value of Time (VOT)’. The value of time differs depending on several factors: trip purpose, income level, vehicle type, in-vehicle versus out-of-vehicle and magnitude of savings. The following section will describe the methods used in literature to estimate the VOT under each category:

- Trip purpose: As the trip time is consumed, the value of time associated with the activity should be modeled with respect to the value of that activity. Business travelers tend to be value their time higher than non-business travelers.

- Income level: It is well documented that VOT is related to income, but not proportionally (Hobeika, 2003) (Wardman, 2001). Efforts in Europe to model the relationship between the value of time and income include a multiplicative model, a segmented model and a Meta-Analysis model.

- Trip distance: Value of time varies when the trip distance is different. In particular, intercity trips time is usually valued higher than local trips (DOT 1997) in that intercity travelers need to pay for services such as hotel rooms, meals, and entertainments.

- Vehicle Type: Travel time value of commercial vehicles is a combination of freight value (especially for perishables), drivers’ salary, vehicle costs and overhead costs. Personal vehicle travel time value considers drivers’ salary and vehicle operating costs. Other modes have unique factors. Table 2-2 lists travel time cost factors used in ground transportation analyses (Dar al-Handasah, 1997).

Table 2-2: Travel Time Cost by Vehicle Type.

Vehicle Class	Value of Time ($ / hr.)
Passenger Car	3.68
Taxi	2.80
Minibus / Van	2.31
Bus	2.31
Medium Truck	3.24
Heavy Truck	3.24

This research reveals trucks employ lower VOT than passenger cars. However, in other research findings, trucks are generally considered to have a higher VOT than passenger cars because the truck deliveries are more time-sensitive. For example, a study in the Netherlands (Gunn, H, 1997) used $20/hr VOT values for trucks.

- In-vehicle and Out-of-vehicle: Waiting and walking are usually assigned a higher VOT than in-vehicle time. Some studies assigned 1.3 to 2 times those of in-vehicle time values.

- Magnitude of savings: The value of time differs by the magnitude of time saved.

In addition to time savings, improvement of infrastructures, expansion and installation of facilities can contribute to the reduction of accidents too. Accidents statistics should be gathered or estimated before and after the improvements. The cost per accident, by degrees of severity, should be estimated based on vehicle mix, driver behavior and climate. The National Safety Council issues periodic report on the costs of vehicular accidents. Representative accident costs per reported accident can be found in AASHTO, 1977, table 11.

As for air transportation, passenger travel time tends to be valued higher than ground transportation because it represents a higher-cost service used by higher income groups. Studies suggest air travelers' value of time is proportional to their wages (Gronau, 1970), (Brown, 1971), (De Vany 1974), (Gellman Research Associates, 1998), (Miller, 1996). The percentage ranges from 61 to 85 percent for business purpose travelers and 112 to 170 percent for non-business travelers. FAA adopted a value of 150 percent of the wage for personal air travel based on a Gellman report (Gellman Research

Associates, undated). However, the Department of Transport (DOT) believes that 130 percent is considerably higher than other estimates and recommends 70 percent (DOT, 1997). Later FAA accepts the DOT figures and uses it as recommended value of time in FAA investment and regulatory decisions (FAA 2003). Table 2-3 lists the recommended value of time.

Table 2-3: Recommended Hourly Values of Travel Time Savings (2000 U.S. dollars / person).

Category	Recommendations	Sensitivity Range	
		Low	High
Air Carrier			
Personal	$23.30	$20.00	$30.00
Business	$40.10	$32.10	$48.10
All Purpose	$28.60	$23.80	$35.60
General Aviation			
Personal	$31.50	NA	NA
Business	$45.00	NA	NA
All Purpose	$37.20	NA	NA

Compared with the FAA figures a survey conducted by the Air Transport Association (ATA) (ATA, 1993) yields estimates of $27.80 for personal travel and $34.50 in 1993 dollars. Another survey conducted by the Aircraft Owners and Pilots Association (AOPA) where general aviation travelers constitute a large portion of the attendants indicates a VOT of $37.50.

In future analysis, the FAA recommended value will be used as they are applied in the standard FAA investment appraisals.

External Costs

In economics external cost refers to a negative side-effect of a project, transaction, or production. In a voluntary economic transaction between person X and Y, an external cost may be imposed involuntarily on individual Z, violating their freedom of choice (Wikipedia, 2006).

During a transportation project, one of the most common external costs is the environmental impact including noise and emission. In terms of SATS, improved travel time could attract more travelers to make more trips by small aircraft. On the other hand, increasing SATS operations would introduce heavier pollution including emission and noise. These negative social effects need to be analyzed carefully during a project appraisal. The following section presents a review of available aviation environment analysis packages.

- **Noise Analysis Models**

Since the late 1950s, air transportation environmental impact has generated controversy from many communities around airports. This concern led to the passage of legislation by Congress and thus regulations by the aviation administration. Therefore, aviation noise and emission, two major environment negatives, have been the subject of studies and regulations ever since.

In the early years, noise at airports was surveyed by continuously monitoring sound exposure levels at places of interests. An important survey revealed that when the sound levels exceed 65 decibels, people report a noticeable increase in annoyance (Schultz, 1978). This survey assigned additional weight to sounds at night and this measurement became the standard aviation noise measurement: Day/Night Average Sound Level (DNL).

The increased number of noise studies around airports prompted the development of the Integrated Noise Model (INM) by the Federal Aviation Administration (FAA) in 1978. This model is the standard tool accepted by the federal government to conduct Federal Aviation Regulation (FAR) Part 150 noise compatibility planning and FAA Order 1050 environmental analysis (FAA, 1999). It is an average value model to quantify annual noise influences using the concept of an ‘average annual day’. The average annual day comprises various typical long-term average conditions. The DNL is one of the 16 noise metrics supported by INM. The aircraft profile and noise calculation algorithms are based on three documents (FAA, 2002): the Society of Automotive Engineers (SAE) Aerospace Information Report (AIR) SAE-AIR-1845 (SAE, 1986), SAE-AIR-1751 (SAE, 1986) and SAE Aerospace Recommended Practice (ARP) SAE-ARP-866A (SAE, 1975). Noise impacts are reported by contour areas, population affected and noise level at points of interests.

In addition to the single-airport analysis tool INM, Metron Aviation developed the Noise Integrated Routing System (NIRS) under contract to the FAA to address large-scale aviation noise modeling over multi-state regions (Metron Aviation, 2005). The first version of NIRS was introduced in 1998 and new capacities are integrated continuously. The NIRS model allows users to specify tracks in three dimensions or follow standard tracks. Traffic elements that cause principle noise impacts can be identified. It also provides comparisons of noise impacts across alternative airspace routing designs. Noise analysis results of NIRS are presented by noise comparison maps and tables.

The INM models fixed-wing aircraft operation noise impacts and helicopter and rotorcraft are modeled by the Heliport Noise Model (HNM) and the Rotorcraft Noise Model (RNM). The HNM is based on the INM but it is able to model more complicated helicopter flight activities. The RNM, developed by NASA, is capable of developing approach and departure noise abatement procedures to promote civilian use of rotorcraft (ATAC, 2000).

As a military counterpart of the INM, Noisemap has been used to model sound exposure in the vicinity of military air bases. Another military aircraft noise analysis model called the Military Operating Area and Range Noise Model calculates noise from subsonic military aircraft over Military Training Routes (MTRs), Military Operating Areas (MOAs) and Special Use Airspaces (such as ranges).

One common feature of all models is the evaluation of noise exposure due to multiple aircraft activities. Single aircraft flyover noise levels can be evaluated using Menu 10 or an updated version Sound Exposure Level Calculator (SELCal). There are papers on enhancements to INM and INM error analysis in the literature. Such discussion is beyond the scope of this dissertation and thus will not be elaborated.

- **Emission Analysis Models**

Emission represents another primary environmental concern introduced by air transportation besides noise. Research on aviation emission effect is driven by increasing air transportation activities and environment protection awareness since the 1960s.

The most widely applied terminal area emission analysis model in U.S. is the Emission and Dispersion Modeling System (EDMS) developed by FAA. In 1998, FAA revised its policy on air quality modeling procedures to identify EDMS as the required model to perform air quality analyses for aviation sources instead of a preferred model. It utilizes Aircraft Engine Emission Databank provided by ICAO to estimate emissions during five phases, namely idle, takeoff, climb out, approach and touch-and-go operations. The model integrates several Environment Protection Agency's (EPA) model to estimate aircraft, Ground Support Equipment (GSE) and Auxiliary Power Unit (APU) emissions. Pollutants such as Total Hydrocarbons (THC), Non-Methane Hydrocarbons (NMHC), Volatile Organic Compounds (VOC), Oxides of Nitrogen (NOx), and PM2.5 are calculated. EPA's state-of-art dispersion model AERMOD, along with its supporting modules, is integrated for dispersion analysis. As detailed airport information is required for dispersion analysis, air quality evaluation will concentrate on emission only. The EDMS does not estimate emission beyond mixing height. Model's default mixing height is 3,000 ft and user can modify this value according to local condition.

For altitude beyond 3,000 ft, a method developed by Boeing Company called Boeing Method (BM2) has been widely applied (Baughcum, 1996). Different from the EDMS, BM2 emission estimation is based on power levels instead of set mode points. It is essentially a curve fitting method that plot emission indices and fuel flow on a logarithmic basis and make a series of linear fits between pair of mode points.

Similarly, EUROCONTROL developed a Toolset for Emission Analysis (TEA) (Eurocontrol Experiment Center, 2003) including an emission module Advanced Emission Model (AEMIII), a contrail formation prediction tool CONTRAIL, and a meteorological database. The AEMIII applies the same emission estimation philosophy in the vicinity of the airport as the EDMS and extends the analysis to en-route. En-route emission analysis is based on aircraft fuel burn calculated using the Base of Aircraft Data (BADA). Emission rate and fuel flow from the ICAO databank is adapted to the atmospheric condition using the BM2. The model output agrees well with the historical data (Carlier, 2004)

Climate change caused by emission will not be considered in this dissertation.

- **Aviation External Costs**

Aviation external costs including noise and emission have been a major concern when airport infrastructure project is proposed.

Historically, measurement of the economic value of quietude is based on fluctuation of real estate values before and after aviation noise influence is introduced (Nelson J.P., 1980, 2003), which reflects public's willingness to pay for quietude. A regression analysis conducted at 33 airports in Canada and U.S. concluded a noise discount of 0.50% - 0.60% per dB (Nelson J.P., 2003) in real estate value. A summary table of previous research results can also be found in this paper (Nelson J.P., 2003) as well. Survey studies suggested a non-linear nature of noise impact on property prices in that noise level in excess of 75 dB may cause health problem and thus unusable for residence (Feitelson et al, 1996).

Aviation emission degrades local or global air quality and thus may cause adverse health problems. A Europe report (Wit, R.C.N., 2003) summaries the following result from literature:

Table 2-4: Overview of Middle Estimates of Emission Damage Costs (Unit: Euro / kg, in 1999 Euros).

	Average	Urban	Rural
NOx	9	12	7
PM_{10} / $PM_{2.5}$	150	300	70
HC	4	6	3
SO_2	6	10	4

Internalization of external costs has attracted extensive discussion in the literature. However, no consensus has been reached. A wide range of value is suggested due to diverse methodology, assumption and economic conditions. Traditionally the external cost is not included in the cost-benefit analysis explicitly because of the controversy over the methodology.

Environmental impacts of SATS in terms of noise and emission are modeled using standard FAA toolbox. Please refer to the chapter of methodology for more detailed discussion. Sensitivity studies will be conducted upon the values suggested in the literature to derive environmental costs. However, due to the wide range of the values and diverse methodology to derive them, those costs will serve as a reference instead of an integrated cost component.

Benefit-Cost Analysis (BCA) Guidelines

FAA BCA Analysis Guidance

In 1994, FAA established the requirement for BCA to demonstrate the merit of any capacity-related program seeking federal Airport Improvement Program discretionary funds (FAA BCA Guidance, 1999). In 1999, regulation was passed to demand all capacity projects with a budget more than 5 million to have a total discounted benefit that exceeds total discounted cost. Meanwhile, FAA published an airport benefit-cost analysis guidance that provides comprehensive descriptions on standard procedure, cost and benefit measurements and risk analysis. Table 2-5 shows suggested cost and benefit components and their units.

Table 2-5: Cost and Benefit Components and Unit.

Benefit Type	**Measurement Unit**
Reduced Delay	
• Reduced aircraft delay	• Reduced aircraft delay hours by airborne, taxi, or gate status for each aircraft class (air carrier, commuter, GA, military)
• Reduced passenger delay	• Reduced passenger delay hours by airside, ATB, and landside status • Reduced passenger vehicle delay hours in landside access
• Reduced cargo delay	• Reduced units of express cargo arrived at/departing from airport after time required to make guaranteed delivery time • Reduced air freight ton delay hours by airside, ATB, and landside status • Reduced truck delay hours in landside access

Improved Schedule Predictability	
• Aircraft operator ability to make more efficient use of equipment and personnel due to more predictable schedules	• Reduced numbers of aircraft and crew required to accommodate posted schedules
• Passenger confidence to take later flight with expectation of arriving at destination on time • Passenger confidence to arrive at ATB closer to flight time with expectation of making flight • Passenger confidence to leave residence or business later for airport with expectation of arrival at ATB in time for check in	• Reduced hours of passenger travel time scheduled to accommodate potential delay by airside, ATB, and landside components (less the amount of reduced delay associated with the project)
More Efficient Traffic Flows	
• Reduced aircraft vectoring and taxing	• Reduced aircraft and passenger hours due to more efficient layout of runways, taxiways, hold pads, and aprons
• Shortened pedestrian traffic distances	• Reduced passenger time required to walk or travel within ATB (not attributable to reduced ATB congestion)
Use of Larger, Faster and/or More Efficient Aircraft	
• Reduced aircraft operation costs and shorter passenger travel times due to service by larger, faster, and/or more efficient aircraft	• Lower cost/fare per revenue passenger mile • Lower cost/charge per revenue cargo ton mile • Reduced passenger hours associated with new direct flights • Reduced passenger hours associated with new jet flights • Reduced cargo ton hours associated with new direct flights
Safety, Security, and Design Standard Benefits Associated with Capacity Projects	
• New capacity project complies with FAA safety, security, and design standards	• No benefits applicable. All new capacity projects must be built to FAA safety, security, and design standards to qualify for AIP funds
• New capacity project enables compliance of pre-existing infrastructure within FAA safety, security, and design standards	• Value of most cost-effective alternative means to bring pre-existing infrastructure into compliance with FAA safety, security and design standards (if new project were not built)
• Increased safety associated with precision approaches	• Number of precision approaches flown with new landing system (will be calculated by FAA)

Environmental Benefits	
• New capacity project complies with Federal environmental requirements	• No benefits applicable. All new projects must be built to Federal environmental requirements
• New capacity project brings pre-existing infrastructure into compliance with Federal environment requirements	• Value of most cost-effective alternative means to accommodate Federal environment requirements (if new project were not built)
Airport Operating and Maintenance Benefits	
• Lower operating and maintenance costs	• Reduced employees, power, fuel, and maintenance materials per passenger

In addition, FAA sponsored publishing guidance on economic value for investment and regulatory decisions (GRA Inc., et al, 2004). The guidance provides suggested values of:

1) Value of Time
2) Value of Life
3) Aircraft capacity and utilization factors
4) Aircraft operating costs
5) Unit replacement and restoration costs of damaged aircraft
6) Economic values related to aircraft performance factors
7) Labor costs
8) Aviation accident investigation costs

The two reports provide detailed guidance on BCA analysis procedure and values, which reflect FAA's emphasis on extensive cost benefit estimation.

Eurocontrol CBA Analysis Guidance

Eurocontrol, different from FAA, devotes more efforts to collaborative air traffic management among European countries. Therefore, Eurocontrol is developing the European Air Traffic Management System (EATMS) to support this mission. A series of reports and guidelines have been published to assist benefit and cost assessment for strategic planning and initial stage of EATMS program (Eurocontrol 2000, 2003). The reports suggest taking the following cost and benefit components into consideration (Eurocontrol 2003):

Benefit category:

1) Additional revenue to Stakeholders
2) Investment expense savings
3) Flight Efficiency Improvements
4) Improvements in productivity of ATSP
5) Reliability savings
6) Delay cost savings or avoidance
7) Environmental benefits
8) Safety benefits
9) More predictability of operations
10) Increase in military mission effectiveness
11) Airports operating expense savings
12) Contingency benefits
13) Upgradeability
14) International commitments
15) Harmonization, co-ordination, convergence and standardization

The 2003 report breaks down cost to each stakeholder. For the sake of simplicity, these costs are not listed here.

Chapter 3 : Methodology

Introduction

Virginia Tech Air Transportation System Lab (ATSL) is currently developing the Transportation System Analysis Model (TSAM) to evaluate aviation policy and technology impacts. TSAM was initially designed to quantify the impacts of the Small Aircraft Transportation System (SATS). SATS comprises a series of technologies developed by NASA Langley Research Center and the National Consortium for Aerospace Mobility (NCAM) to provide point-to-point air transportation service between thousands of underutilized airports. Around 3,416 public airports in the United States are identified as candidate SATS airports. Candidate SATS airports are public landing facilities with a minimum runway length of 915 meters (3,000 ft.). If priced correctly, the SATS Program could improve the mobility of a segment of the population using these services. NASA developed four SATS technical capacities as part of the SATS Program. These capabilities are: a) improvements to Lower Landing Minima (LLM) at airports without precision approaches, b) High Volume Operations (HVO) at non-towered airports, Single Pilot Performance and Safety operations (SPP), and d) seamless integration of SATS vehicles in the en-route airspace system (ERI). The LLM capability is the subject of further investigation in this dissertation.

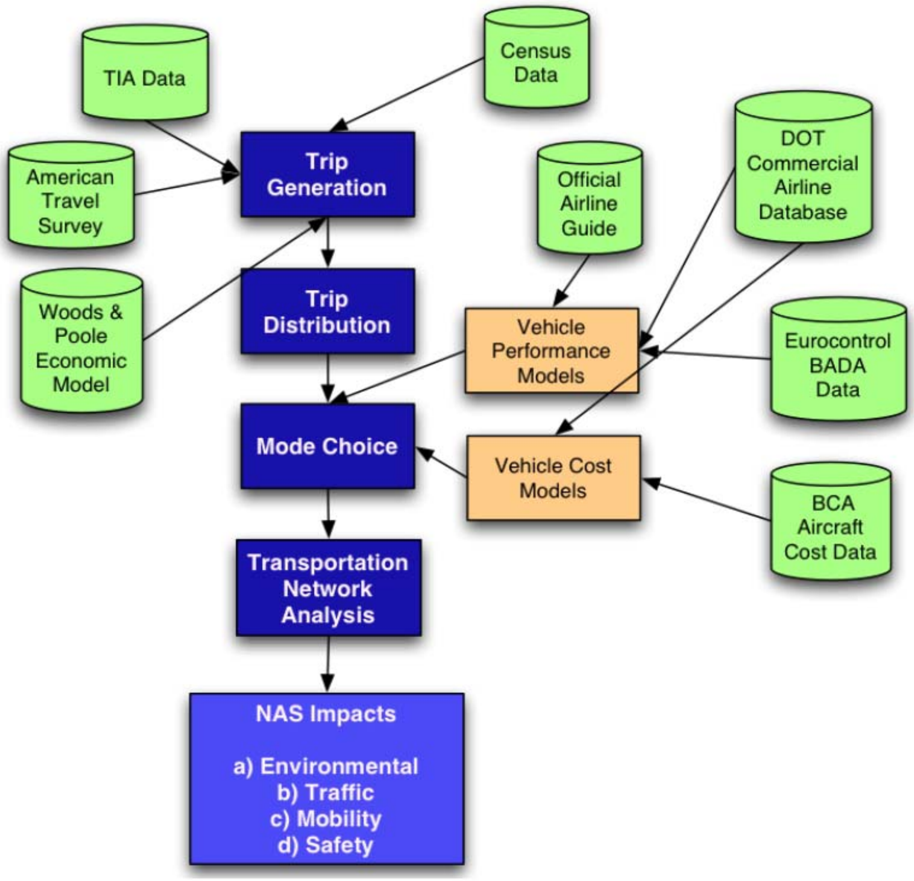

Figure 3-1: TSAM Model Components and Data Sources.

Using the classical four-step transportation planning procedure shown Figure 3-1, Virginia Tech developed the Transportation Systems Analysis Model (TSAM) to estimate demand for SATS operations. At the core of the model is the use of a mode choice model where the number of SATS airports and the reliability of the airports play an important role in the outcome of mode choice behavior. The four-step planning model is a sequential demand forecasting model composed of **trip generation**, **trip distribution, mode choice** and **trip assignment** (See Figure 3-2).

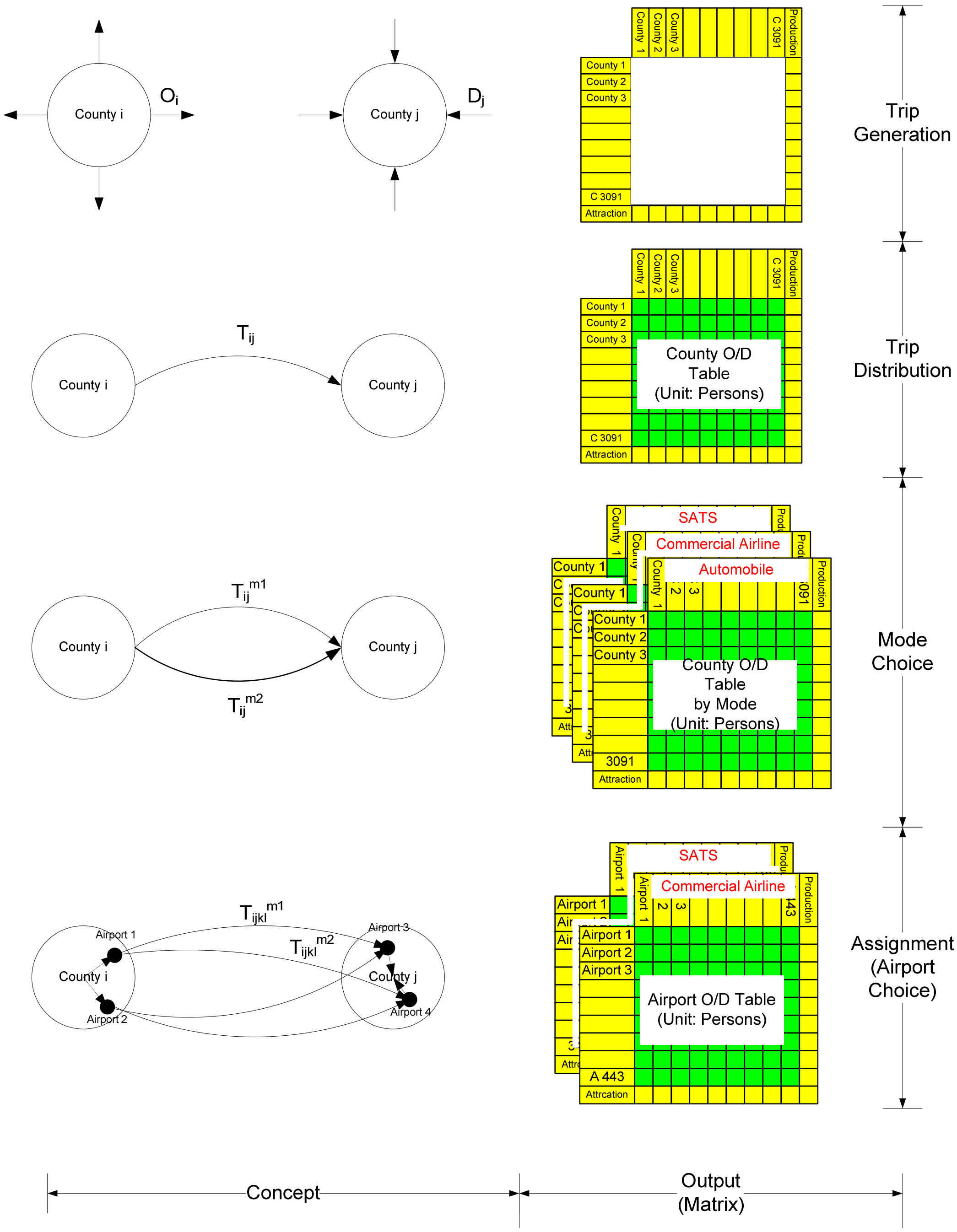

Figure 3-2: Multi-step Illustration of Trip Demand Analysis.

Trip generation is used to predict the number of trips produced and attracted at each zone of interest. In TSAM, the number of trips produced and attracted is estimated at county level. There are 3,091 counties in the U.S. according to Woods & Poole (Woods & Poole 2005). The trip rate tables are aggregated from the American Travel Survey (ATS) and applied at each county. As shown in Figure 3-2, the output of this model is the number of trips attracted and produced by each county.

Trip distribution is used to predict the Origin-Destination (OD) flows. Gravity model is applied in this step to distribute trips obtained from the trip generation. Number of trips is segregated by trip purpose and income groups because travelers with different characteristics behave differently when making travel decisions. Travelers are grouped by two trip purposes, namely business and non-business and five income groups. Therefore the output of this procedure is a trip interchange table by county, trip purpose and income level (2 * 5 * 3091 * 3091). Figure 3-2 block2 shows the content of one of the output matrix.

Mode choice predicts the percentage of person-trips selecting each mode. It is based on the concept that each person weights the attractiveness of each available travel mode when making their mode choice. The attractiveness of each mode is called utility which includes such parameters as travel time and cost. People with different trip purpose or from different income group may assign a different weight to travel time and cost than others. Therefore the mode choice applies different coefficient to travel time and cost based on trip purpose and income group. The output of this step is the number of traveler taking each mode by trip purpose and income group.

Trip assignment converts the origin-destination flows for each mode on specific routes through the respective networks. This step is partially realized by the airport choice model. Number of trips between each county pair is converted to trips between airport pair. The assignment represents an unconstrained scenario without capacity constrain. An airport capacity and delay model is expected to solve this problem.

Lower Landing Minima

SATS is proposed to utilize small to medium airports to save travel times. However, majority of these airports do not currently have precision approach. Thus the dispatch reliability heavily depends on the weather condition. Poor reliability confines the utility of the SATS mode and in turn affects the number of SATS travelers.

The FAA plans to deploy Wide Area Augmentation System to improve reliability of smaller airports. WAAS is expected to provide near precision approach to achieve lower landing minima at minimally equipped airports. Theoretically, WAAS is capable to provide 76.2m (250 ft.) Decision Height (DH). Three types WAAS aided approach procedures have been published, namely Lateral Navigation (LNAV), Lateral /Vertical Navigation (LANV/VNAV) and LPV. LNAV is a form of non-precision approach with an optimal minimum of 400 ft. This approach provided lateral guidance to pilots flying WAAS certified approaches. LNAV/VNAV provides precision approach down to 106.7m (350 ft) by offering both lateral and vertical guidance. LPV combines LNAV/VNAV vertical accuracy to utilize full WAAS satellite signal protection limits. LPV approaches can achieve near instrument landing system precision down to 76.2m (250 ft.). The first commercial flight equipped with a Technical Standard Order (TSO) certified GPS/WAAS receiver was conducted in Juneau, Alaska on March 31st, 2003. The best-known receivers today provide localizer-equivalent precision. According to FAA there are 2057 LNAV, 719 VNAV and 39 LPV certified procedures (FAA, GPS/WAAS website).

WAAS approach certification procedures involve geo-spatial analysis of terrain and navigation facility information to determine Obstacle Clearance Surfaces (OCS). The approach analysis also includes the determination of Decision Height (DH) minima. These procedures are developed by specialists considering potential obstacles in the approach path on a case-by-case basis for each runway. For the SATS program, a system-wide study is required to derive the optimal airport set that maximizes the potential SATS demand subject to airport cost-benefit constraints. There are few active nationwide LLM research programs. The GPS Approach Minima Estimator (GAME), developed by the

MITRE Corporation, is a computer-based tool to estimate GPS landing minima considering terrain and controlling objects (Crane, 2001). Height Above Touchdown (HAT) and Runway Visual Range (RVR) are derived without consideration of airport infrastructure effects. However, the benefit of introducing WAAS approaches cannot be assessed without aviation demand and without consideration of obstacle removal costs. The approach proposed in this paper addresses some of the shortcomings of GAME as we consider the potential SATS demand using a logit model.

Public airports with paved runway more than 3,000 feet are considered SATS compatible. Currently the Instrument Landing System (ILS) continues to be the primary form of precision approach procedure in the U.S. In 2004, there were 685 airports in the U.S. with full ILS capability (Trani, 2003). This implies majority of the SATS candidate airports do not have the precision approach capability. The lower landing minima technology is expected to deliver minima down to 100 ft. and ½ mile visibility to those airports. The lower landing minima technology improves the dispatch reliability in that it allows precision approaches in near all-weather conditions. However, not all the SATS candidate airports need to be updated with LLM capability when the user benefits (travel time savings, ticket savings, etc) do not justify the user costs (infrastructure investment). An optimum set of LLM-enabled airports needs to be identified for updates based on cost-benefit analysis.

User benefits and costs of LLM are the two major components in this analysis. Figure 3-3 shows a flowchart of user benefit and cost analysis procedure.

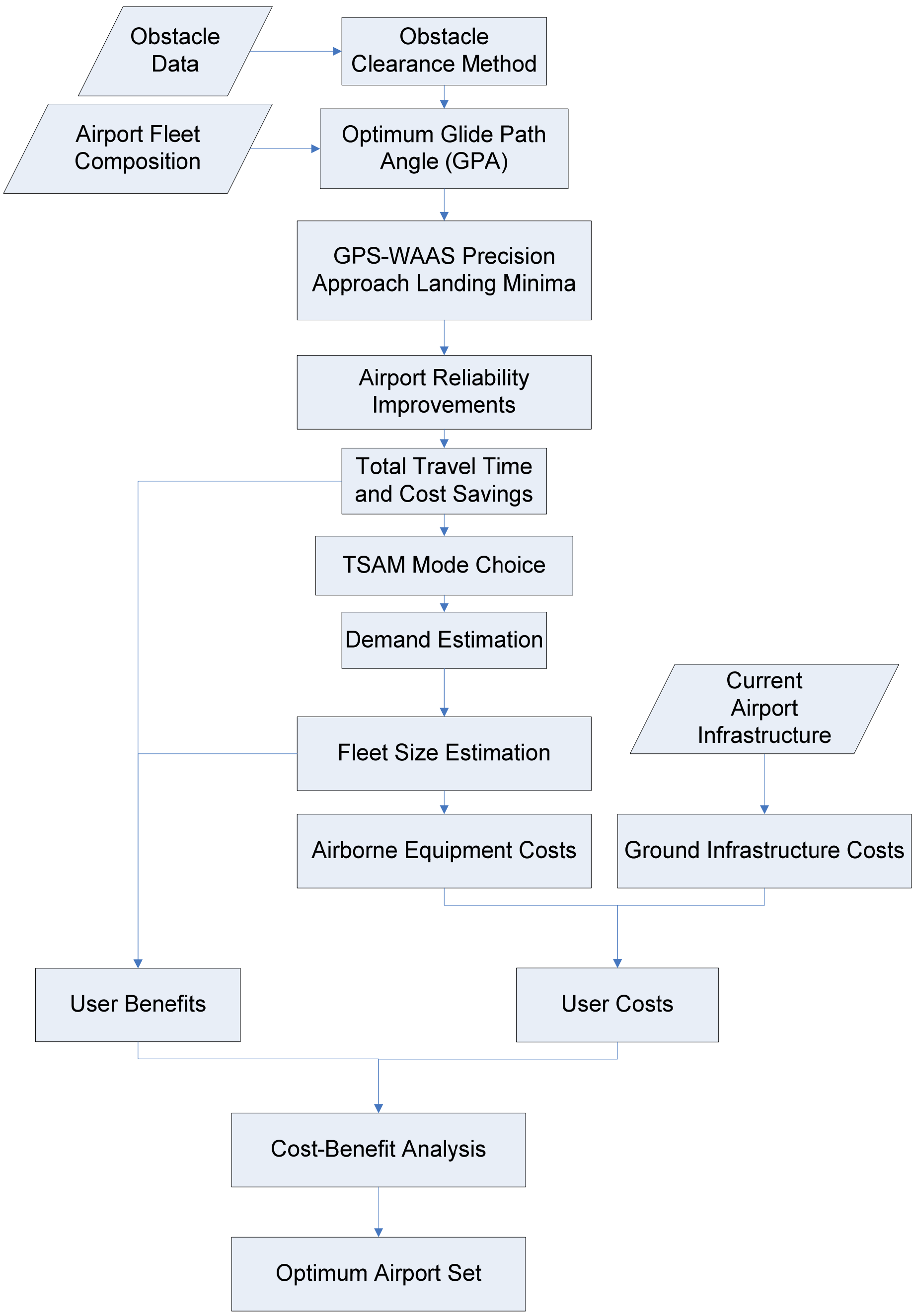

Figure 3-3: LLM Cost-Benefit Analysis Flow Chart.

- User Benefits

Generally, user benefits from improved infrastructure constitute two aspects: travel time and travel cost. One of the advantages of SATS is to utilize neighborhood airports to operate from origin to destination without detouring to hubs. The improved dispatch reliability will open more airports to public and thus save access and egress times. Increased utility of SATS mode would offer travel time savings and thus attract more passengers. Travel time savings and induced demand constitute major system benefits brought by LLM.

The Wide Area Augmentation System (WAAS) is projected to be the primary tool to achieve lower landing minima. However, safety concerns may prevent some airports from using WAAS if obstacles around runways impose a danger to pilots. Analysis was executed to extract the minimum required Glide Path Angle (GPA) based on critical obstacles. Different approach groups of aircraft have different maximum GPA constraints (FAA, 2003) and a GPA of less than 5 degrees is preferred by instrument approach pilots (Lancaster, 2001).

Different airport set scenarios were executed using Virginia Tech TSAM to estimate demand impacts. Improvements of dispatch reliability will not only save travel times for current users, but also induce additional user from other modes due to attractiveness of the utility.

Apart from travel time savings, researches have indicated a total 11% of ticket price decrease due to the increase of load factors.

By exercising TSAM, travel time savings and demand under different scenarios can be obtained. Benefits in monetary terms can be estimated using the VOT concept.

- Infrastructure Costs

Infrastructure investments to support GPS-WAAS approach consist of two components: Ground infrastructure investments and airborne equipment updates. Ground infrastructure requirements include lighting system, obstacle removal, approach fix installation, markings and specific Runway Protection Zone, Object Free Zone and Runway Safety Area dimension. Some of the requirements and associated costs will be discussed in detail.

Figure 3-4 lists infrastructure requirements at different landing minima. Different technology may incur different infrastructure costs.

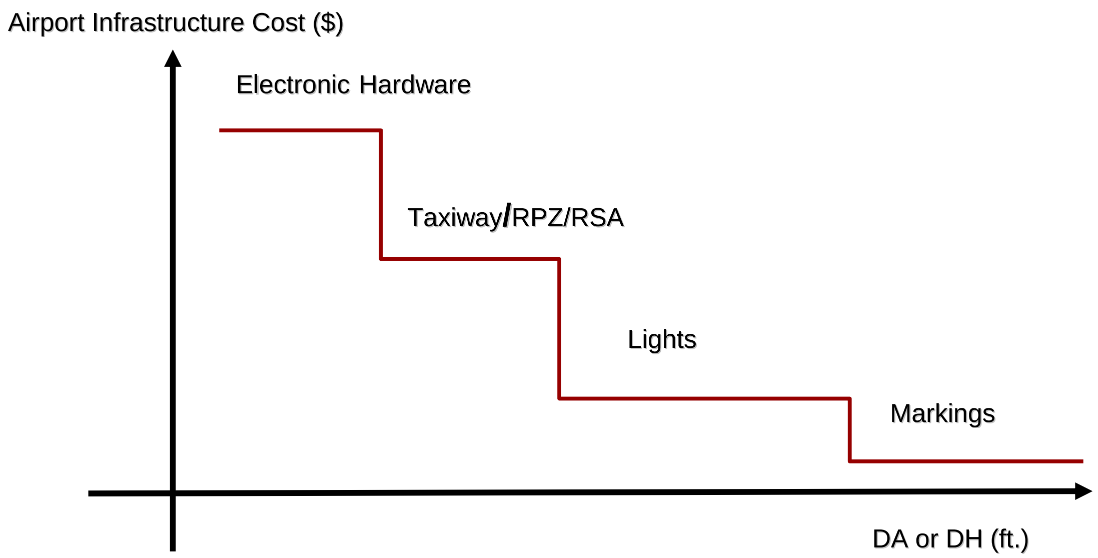

Figure 3-4: Airport Infrastructure Requirements.

Markings

Airport design regulations require proper markings for associated approaches. GPS precision approach requires the most complete runway markings (Table 3-1).

Table 3-1: Marking Requirements (FAA Order 8260.48, 1995).

Marking element	Visual runway	Non-precision runway GPS Non-precision	Precision runway GPS Precision
Designation	X	X	X
Centerline	X	X	X
Threshold marking	X[1]	X	X
Aiming point	X[2]	X[2]	X
Touchdown zone			X
Side stripes	X[3]	X[3]	X

1 Only required on runways used, or intended to be used, by international commercial transport.
2 On runways 4,000 feet (1200 m) or longer used by jet aircraft.
3 Used when the full pavement width may not be available as a runway.

Approach Lighting Systems

Standard precision landings require a combination of low GPA and standard approach lighting systems. Table 3-2 shows the GPA and lighting requirements to achieve a certain minimum. Several lighting system costs can be found in literature (Delta Airport Consultants, 1999), (Minnesota DOT, 2006).

Table 3-2: Standard Precision Landing Minimums (FAA, 1999).

Glide Path Angle (with Approach Light Configuration)		Minimum HAT	Aircraft Category			
			A	B	C	D & E
			Minimum Visibility			
3.00º – 3.10 º	★	200	¾ 4000			
	#	200	½ 2400			
	$	200	1800			
3.11º – 3.30 º	★	200	¾ 4000		NA	
	★	250	¾ 4000		1 5000	NA
	#	200	½ 2400		NA	
	#	250	½ 2400		¾ 4000	NA
	$	200	1800		NA	
	$	250	1800		½ 2400	NA
3.31º – 3.60 º	★	200	¾ 4000		NA	
	★	270	¾ 4000		1 5000	NA
	#	200	½ 2400		NA	
	#	270	½ 2400		¾ 4000	NA
	$	200	2000		NA	
	$	270	2000		½ 2800	NA
3.61º – 3.80 º	★	200	¾ 4000		NA	
	#	200	½ 2400		NA	
3.81º – 4.20 º	★	200	¾ 4000	NA		
	★	250	¾ 4000	1 5000	NA	
	#	200	½ 2400	NA		
	#	250	½ 2400	¾ 4000	NA	
4.21º – 5.00 º	★	250	¾ 4000	NA		
	#	250	½ 2400	NA		
5.01º – 5.70 º	★	300	1 5000	NA		
	#	300	¾ 4000	NA		
5.71º – 6.40 º Airspeed NTE 80 Knots	★	350	1¼	NA		
	#	350	1 5000	NA		

★ = No Lights # = MALSR, SSALR, ALSF
$ = # Plus TDZ/CL Lights NA = Not authorized

Runway Protection Zone (RPZ) and Object Free Zone (OFZ) Dimension

Dimensions of RPZ and RSA increase as the landing minima decreases. Additional land acquisition costs need to be estimated if the extensions of RPZ and OFZ go beyond the control of airport agencies.

Table 3-3: RPZ/OFZ Requirements for Different Landing Minima.

Visibility Minimums	Changes in Airport Design Standards
Visual to Not lower than 1 – Mile (1,600 m)	No change in airport design standards
Not lower than 1-Mile (1,600 m) to Not Lower than ¾ - Mile (1,200 m)	Increase in RPZ dimensions Increase in threshold siting standards
Not lower than ¾ - Mile (1,200 m) to Not lower than CAT I	For aircraft approach category A & B runways: Increase in runway separation standards Increase in RPZ dimensions Increase in OFZ dimensions Increase in runway design standards Increase in threshold siting standards
	For aircraft approach category C & D runways: Increase in runway separation standards Increase in RPZ dimensions Increase in OFZ dimensions Increase in runway design standards Increase in threshold siting standards
Not lower than CAT I to Lower than CAT I	Increase in OFZ dimensions for runways serving large airplanes Increase in threshold siting standards

The requirements above demand a comprehensive database of current airport infrastructure inventory. Appropriate data source containing such information need to be located to estimate ground investment costs.

As for the airborne equipment requirements, many types of commercial GPS/WAAS receivers are available in the market. But only one of them is certified to provide localizer-equivalent precision (GPS/WAAS Gamma 3). Apollo CNX 80 from Garmin. Garmin also offer a WAAS upgrade to Garmin 400/500 series GPS receivers at a cost less than $1,500 (Ruley 2004). These onboard equipments are necessary for each aircraft utilizing GPS/WAAS navigation.

The total cost should be estimated using a life cycle cost approach. Salvage value and maintenance cost are to be properly estimated.

Once the user benefits and costs are estimated, the optimum set of airports can be identified (Figure 3-5).

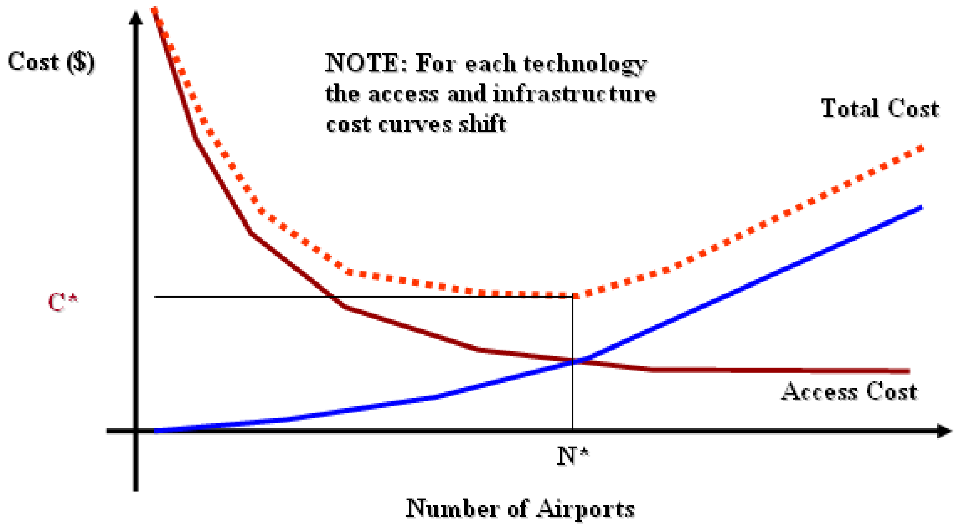

Figure 3-5: Cost – Benefit Analysis.

Several trade-off studies need to be conducted to quantify user benefits and costs. Necessary user benefits and costs estimations are summarized in the table below.

Table 3-4: User Benefits and Costs Estimations for Trade-off Studies.

	User Benefits	User Costs
Number of airports and cost of technology	Travel time and cost savings, and induced demand. Benefits various because difference technology delivers different landing minima possibility.	Costs estimation for different technology, both ground and on-board
Number of airports and demand	Travel time and cost savings and induced demand.	Cost estimation for different number of airports, both ground and on-board.
Dispatch reliability and demand	Travel time and cost savings and induced demand. Weather data needed to analyze dispatch reliability improvement when lower landing minima is achieved.	Cost estimation for the airport and on-board.

Environmental Impact Assessment

From the beginning of the program, SATS proponents identified environment impacts as critical to the acceptance to the concept. The cost-benefit analysis model is expected to assess environment costs in addition to infrastructure costs. This dissertation presents the preliminary evaluation 1) noise impacts performed at five typical SATS enabled airports and 2) emission impacts for the full spectrum of 2,286 SATS enabled airports. The noise and emission impacts of SATS operations are assessed using the standard Federal Aviation Administration (FAA) Integrated Noise Model (INM) version 6.1c and Emission and Dispersion Modeling System (EDMS) respectively.

Three representative SATS aircraft are modeled: 1) a new generation Very Light Jet (VLJ) aircraft, 2) an advanced technology Single-Engine (SE), piston-powered aircraft, and 3) an advanced technology Multi-Engine (ME), piston-powered aircraft. The advanced Single-Engine and Multi-Engine aircraft are substituted by aircraft with analogous features. The VLJ aircraft is modeled as a new vehicle with advanced low-thrust, medium by-pass ratio turbofan engines.

The SATS population considered in the analysis includes proposed very light jets weighing less than 3,181 kg (7,000 lb). The new generation of VLJ aircraft is best represented by the Eclipse 500, currently in flight testing. In our analysis, single engine and multi-engine turboprop are modeled using similar aircraft (i.e., substitution method) because the power plants and flight characteristics of new generation SE piston-powered aircraft are similar to modern aircraft in the same category today. The noise profile of the very light jets is created in the INM and EDMS as they represent new generation of small jets powered by substantially lower thrust engines than their current corporate jet counterparts.

Thirteen current corporate jet engine emission rate tables are extracted to establish basic statistical model between four emission indices (HC, CO, NOx, SOx). A liner relationship can be established between the NOx emission rate and the fuel flow. In the absence of published VLJ CO and HC emission rates, two assumptions are evaluated: same rate as current light jet corporate jets and a reduced (80%) rate. Cessna Citation Sovereign is used as a prototype to model VLJ. SOx emission rate is assumed to be consistent with current light corporate jets.

Our analysis starts with several general assumptions. The ratios of Visual Flight Rules (VFR) over Instrument Flight Rules (IFR) and Day Night operation are extracted from the General Aviation and Air Taxi Activity (GAATA) report. Flight paths are constructed using U.S. Terminal Procedure Charts to model IFR arrivals and departures.

VFR arrival and departure tracks are modeled using standard airport flight patterns. Three types of aircraft compose the SATS fleet: Single-Engine, Multi-Engine aircraft and Very Light Jet. The base scenario (year 2005 without SATS) considers 37.5% of approach, 37.5% of departures and 25% of touch-and-go operations except for jets. Jets only operate 50% approach and 50% departures. All SATS operations are assumed to be itinerant operations with equal number of arrivals and departures. A three percent annual growth in the GA operations has been assumed as well.

Future scenarios with SATS operations (year 2014) are executed based on the SATS demand estimated by the Virginia Tech TSAM Model. Woods & Poole county socio-economic prediction is applied to estimate impacted population. Population around the airports is assumed to grow at the same pace as the county. A typical SATS aircraft, the Very Light Jet (VLJ) is developed as a new aircraft and engine model in both INM 6.1c and EDMS.

SATS noise impacts are evaluated at five general aviation airports: Manassas, Virginia (HEF), Blacksburg, Virginia (BCB), Danville, Virginia (DAN), Teterboro, New Jersey (TEB) and Goodland Municipal Airport, Kansas (GLD). The case study conducted at BCB represents a comprehensive noise study with detailed survey of fleet mix, flight track and runway utilization. These airports were selected among 2,286 airports modeled in TSAM because they represent a good cross-section of airport operations, aircraft mix, runway configurations and proximity to population centers. Baseline scenarios without SATS activities are executed based on average daily traffic operations and using the based aircraft fleet mix reported by Airnav. The GAATA categorizes general aviation aircraft into five major groups and this categorization is applied in our analysis: Single-Engine Piston-Powered, Twin-Engine Piston-Powered, Single-Engine Turbo-Propeller, Twin-Engine Turbo-Propeller and Jet. VFR and IFR flight paths are constructed using the U.S. Terminal Procedural charts and pilot anecdotal information. Runway utilization estimation is gathered from various sources. Local operation information is used wherever available.

SATS emission impacts are analyzed at 2,286 SATS enabled airports. The estimate uses FAA landing facility database to derive airport baseline operations. The same aircraft categorization is followed and one aircraft in EDMS database is chosen to represent the group. Aircraft emission indices in the EDMS are obtained from ICAO Aircraft Engine Emission Databank. Aircraft emission is the product of annual operations and emission rate. Emissions of Group Support Vehicles and Auxiliary Power Units are estimated by the integrated MOBILE 6.

Sensitivity analysis is conducted with different fleet composition and glide path angles for the year 2014.

DataComm Analysis

Impacts of DataComm technology are evaluated using discrete event simulation tool SIMMOD. Three large hub airports at New York metropolitan area are chosen for the study as they represent one of the most congested airspaces in the US. Flight observations from the Performance Data Analysis and Reporting System (PDARS) are analyzed to build airspace structure. PDARS is also used to derive aircraft performance and separation matries. Impacts of DataComm are assessed at individual airport level and then for the entire New York airspace. Demand profile applies FAA Air Traffic Organization (ATO) traffic observations and predictions. SIMMOD is a large-scale discrete event simulation model developed by the FAA. It simulates aircraft movement from gate to gate to assess airport and airspace efficiencies.

New York metropolitan area is chosen for the study. This area is composed of three airports: John F. Kennedy Airport (JFK), La Guardia Airport (LGA) and Newark Liberty Airport (EWR). They represent one of the busiest and most congested terminal systems in the U.S. According to the most recent statistics, JFK, EWR and LGA ranked 7th, 10th and 20th in terms of passenger and all cargo enplanements in 2006 (FAA, 2006). In 2007, JFK, EWR and LGA ranked 5th, 4th, and 7th in terms of total delays (BTS, 2007). Innovative technologies such as DataComm are expected to increase efficiency and reduce controller workload at these busy airports.

Two studies are conducted to analyze the impact of DataComm. Demand and delay is simulated using FAA current and future demand scenarios. Then capacity improvement is evaluated using computer generated demand profiles. The consumer surplus is estimated in the last step using simulation and TSAM model.

Chapter 4 : Lower Landing Minima Analysis

Preliminary Assessment of Lower Landing Minima Capabilities in the Small Aircraft Transportation System Program

Yue Xu, Antonio A. Trani, and Hojong Baik

Published in Transportation Research Record 1915

[Abstract] A preliminary assessment is presented of the required lower landing minima (LLM) capabilities needed to support the Small Aircraft Transportation System (SATS) Program. The goal of this analysis is to understand the number of potentially challenged SATS airports and to identify methods to remove obstacles by using technology solutions. Four obstacle removal methods are considered to assess the challenges faced by the SATS Program in providing LLM capabilities to 3,416 U.S. airports.

Two views of runway obstacle analysis are presented: a critical object analysis and a detailed multi-object analysis that includes terrain information. A comparison is made between decision altitudes (DAs) derived by approach lighting infrastructure and glide path angle thresholds and DA values considering other airport characteristics such as terrain. A detailed case study is presented to compare the single critical object analysis with the more detailed multi-object analysis, which was performed for Blacksburg Airport, in Virginia.

Key words: GPS Wide Area Augmentation System (WAAS), Lower Landing Minima (LLM)

The Small Aircraft Transportation System (SATS) is composed of a series of technologies being developed by the National Aeronautics and Space Administration (NASA) at Langley Research Center and the National Consortium for Aerospace Mobility (NCAM) to provide point-to-point air transportation service between thousands of underutilized airports. The analysis presented here centers around 3,416 public airports in the United States identified as candidate SATS airports, which are public landing facilities with a minimum runway length of 915 m (3,000 ft). If priced correctly, the SATS Program could improve the mobility of that segment of the population who use these services. NASA is developing four SATS technical capacities as part of the SATS Program: (*a*) improvements to lower landing minima (LLM) at airports without precision instrument approaches (PIR), (*b*) high volume operations at no towered airports, (*c*) single-pilot performance and safety operations, and (*d*) seamless integration of SATS vehicles in the en route airspace system. LLM capability is the subject of further investigation in this paper.

The Air Transportation Systems Laboratory at Virginia Polytechnic Institute and State University (Virginia Tech) has developed an air transportation system decision support model to assess the impacts of SATS and other aerospace technologies on the National Airspace System (NAS) (*1*). At the core of the model is the use of a mode choice model in which the number of SATS airports and the reliability of the airports play an important role in the outcome of mode choice behavior. Higher airport reliability can be achieved by utilizing SATS LLM capabilities, thus providing higher accessibility to airports that do not have PIR. The purpose of this study is to understand, from a systems engineering viewpoint, the technical constraints of providing LLM capabilities in support of the SATS Program to 3,416 public airports currently without PIR. The analysis reflects the status of all 3,416 NAS airports and considers the implementation of the Global Positioning System (GPS) and the Wide Area Augmentation System (WAAS).

Literature Review

WAAS is expected to provide a near-precision approach to achievement of LLM capabilities at minimally equipped airports. Theoretically, WAAS is able to provide a decision height (DH) of 76.2 m (250 ft). Three types of WAAS-aided approach procedures have been published, namely, lateral navigation (LNAV), lateral and vertical navigation (LNAV/VNAV), and lateral precision with vertical guidance (LPV). LNAV is a form of no precision approach with an optimal minimum of 121.9 m (400 ft). This approach provides lateral guidance to pilots flying WAAS-certified approaches. LNAV/VNAV provides a precision approach down to 106.7 m (350 ft) by offering both lateral and vertical guidance. LPV combines LNAV/VNAV lateral and vertical accuracy to utilize full WAAS satellite signal protection limits. LPV approaches can achieve near instrument landing system precision down to 76.2 m (250 ft). The first commercial flight equipped with a Technical Standard Order–certified GPS/WAAS receiver was conducted in Juneau, Alaska, on March 31, 2003. The best-known receivers today provide localizer equivalent precision. According to the FAA, there are 2,057 LNAV-, 719 VNAV-, and 39 LPV-certified procedures (*2*).

WAAS approach certification procedures involve geospatial analysis of terrain and navigation facility information to determine the obstacle clearance surface (OCS). The approach analysis also includes the determination of DH minima. These procedures are developed by specialists considering potential obstacles in the approach path on a case-by-case basis for each runway. For the SATS Program, a system wide study is required to derive the optimal airport set that maximizes the potential SATS demand subject to airport cost–benefit constraints. There are few active nationwide LLM research programs. The GPS Approach Minima Estimator (GAME), developed by the MITRE Corporation, is a computer-based tool to estimate GPS landing minima considering terrain and controlling objects (*3*). Height above touchdown (HAT) and runway visual range (RVR) are derived without consideration of airport infrastructure effects. However, the benefit of introducing WAAS approaches cannot be assessed without knowing aviation demand and without consideration of obstacle removal costs. The approach proposed in this paper addresses some of the shortcomings of GAME by using a logit model to consider potential SATS demand.

SATS Candidate Airports

According to the FAA, there are more than 20,000 landing facilities in the United States (*4*). This analysis of LLM capabilities for the SATS Program considers all public-use airports with a paved runway longer than 915 m (3,000 ft). The selection of 915 m of runway takes into account the performance of the set of aircraft expected to make up SATS. The analysis considers proposed very light jets, weighing less than 4,546 kg (10,000 lb), such as the Eclipse 500, Adam 700, Safire Jet, and Cessna Mustang. The SATS aircraft population involved includes proposed and existing turboprop aircraft (Raytheon B300, Pilatus PC-12, Ibis Ae 270, and Cessna Caravan) and the new generation of high-performance single-engine piston-powered aircraft (e.g., Cirrus SR-22 and Lancair 400). A 915-m runway is an absolute minimum for operating a very light jet with a modest payload at sea level conditions. Turboprop and single-engine piston aircraft can operate from a 915-m runway with good payloads. The total number of airports meeting the runway length and ownership criteria is 3,416. This number excludes the top 31 hub airports, to avoid further congestion at these airports. The airport locations are shown in Figure 4-1. Spatial analysis shows that 96% of the U.S. population lives within 30 statute miles from the selected SATS airport set (*1*). In contrast, only 34% of the U.S. population lives within 30 mi of the large airport hubs (31 airports in 2004).

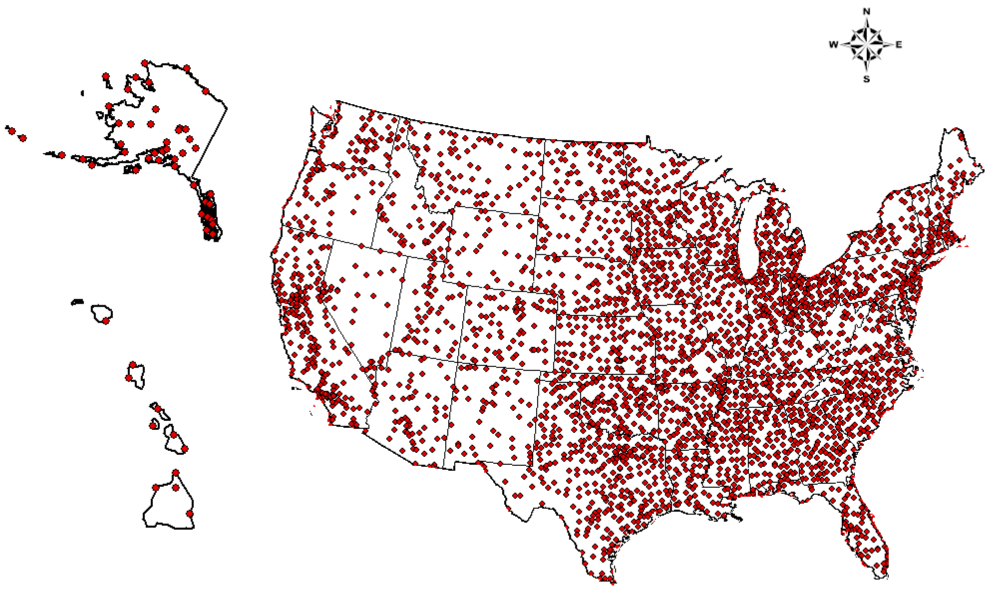

Figure 4-1: SATS Candidate Airports in the United States.

FAA WAAS Approach Procedure

The instrument landing system continues to be the primary form of precision approach procedure in the United States. In 2004 there were 685 U.S. airports with full instrument landing system capability (*4*). This fact implies that many thousands of airports in the SATS set do not have precision approach capabilities. One goal of the SATS Program is to develop airborne technologies (e.g., displays and aircraft controls) aided by WAAS that would allow LLM approaches down to 30.5 m (100 ft) and 1⁄2-mi visibility. The use of GPS and WAAS approaches is the first step in assessing the benefit of LLM technologies in the SATS Program. Technically, SATS aircraft equipped with GPS- and WAAS-enabled technologies can execute precision approaches to airports at which conventional precision approaches are unavailable. However, the approach procedures using WAAS and GPS are restricted by specific FAA criteria related to the size of the WAAS qualification surface (WQS) and OCS (Figure 4-2). The dimensions and clearance criteria of those surfaces determine important approach characteristics such as glide path angle (GPA) and the feasibility of the approach procedure. The FAA criteria to determine the size of WQS and OCS are well documented in FAA technical notes (*5*–*8*).

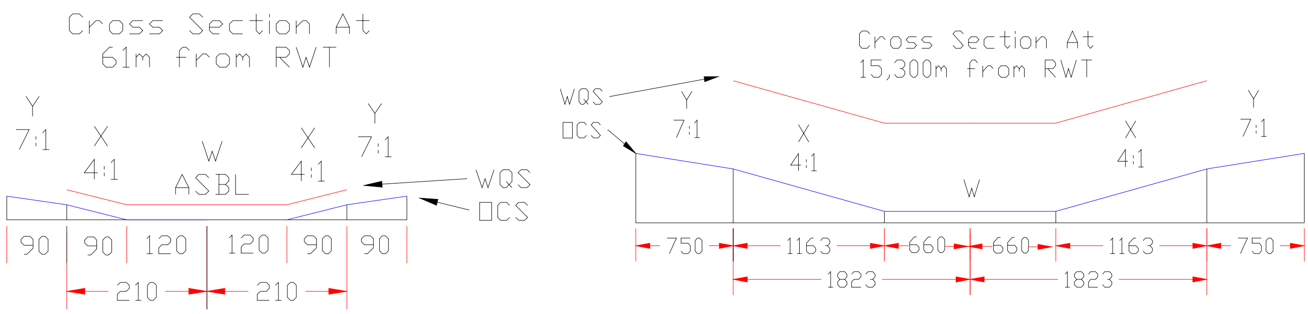

Figure 4-2: WQS/OCS Final Approach Segment Dimensions and Clearance Criteria (in Meters) (source: FAA).

Critical Obstacle Locations and Types

Critical runway obstacle information for this study was obtained from the FAA airport database (*4*). Runways with available obstacle information were extracted for the SATS candidate airport database. The position and type of critical obstacles are identified in Figure 4-3 and Figure 4-4, respectively. The analysis shows that 92% of the critical runway obstacles are found inside the WQS–OCS final approach segment. For this reason, the rest of analysis presented here focuses on the final approach segment only. Figure 4-4 shows that around 60% of the critical obstacles are easily removable objects such as trees, woods, and bushes. However, the possibility of direct removal depends on whether the removable obstacles are inside the airport perimeter. The fact that they are removable but still constitute obstacles (according to the FAA) suggests that they are either unreachable or too costly to remove.

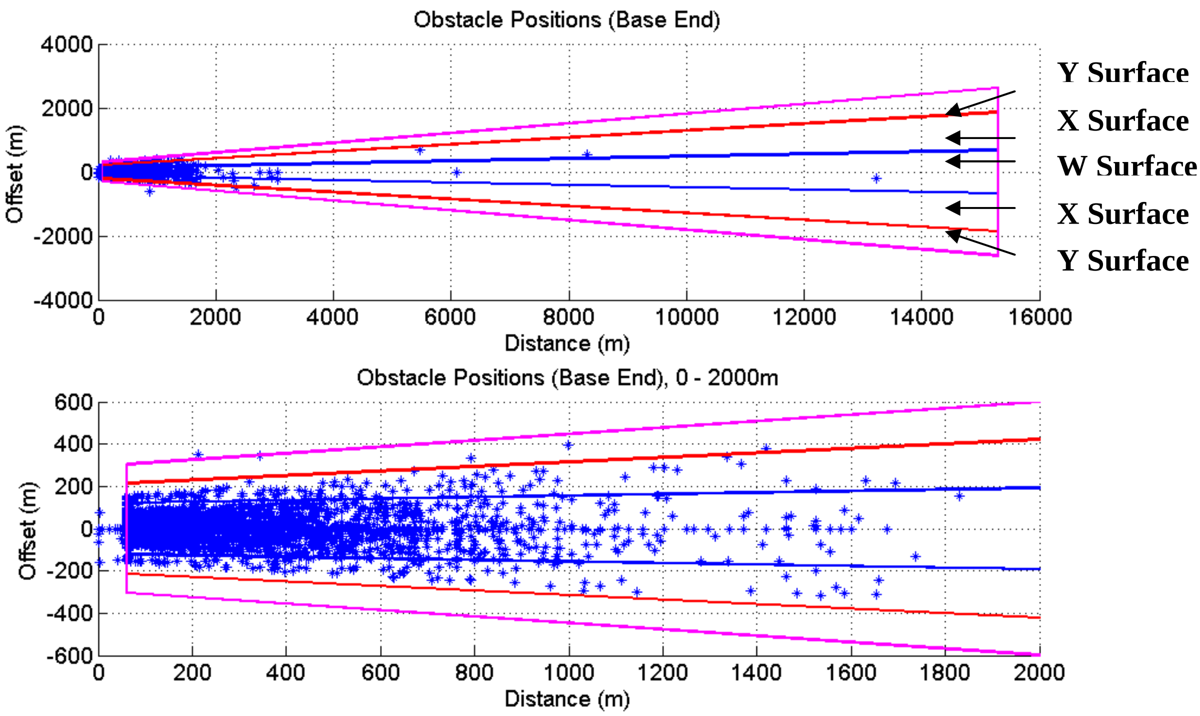

Figure 4-3: Position of Critical Obstacles (3,416 airports).

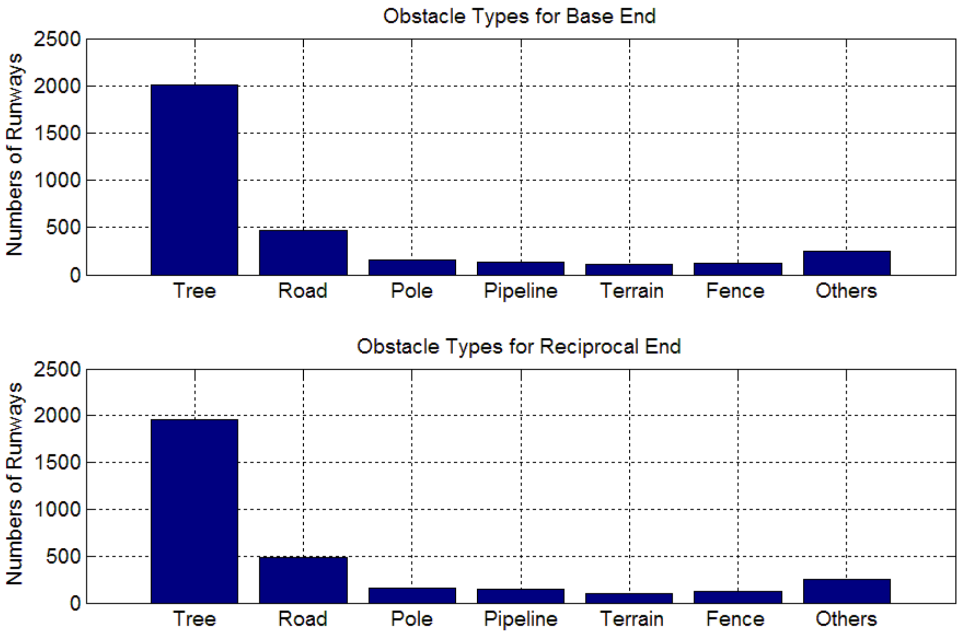

Figure 4-4: Types of Critical Obstacles at 3,416 Runways.

Obstacle Clearance Methods

Other than direct removal, four obstacle clearance methods are proposed to achieve LLM capability and yet satisfy FAA approach design criteria (Figure 4-5): (*a*) raising the approach GPA, (*b*) displacing the runway threshold, (*c*) reducing the WQS, and (*d*) offsetting the final approach segment. Excluding reduction of the WQS, all the other methods can be applied to airports that meet the minimum WQS-OCS criteria. The obstacle clearance method of reduction of the WQS is justified by advances in aircraft controls and displays. Better flight deck navigation displays and controls provide improved pilot situation and lower workload. This improvement could reduce total flight technical errors (FTEs) during execution of WAAS-aided approaches. This subject will be investigated further in the SATS Program and has been demonstrated by the use of three-dimensional predictive displays.

Problematic airports can be identified at which GPS-WAAS approach criteria cannot be met after each method of obstacle clearance has been applied. In the following sections, the results are described of applying the four obstacle clearance methods to the set of 3,416 airports.

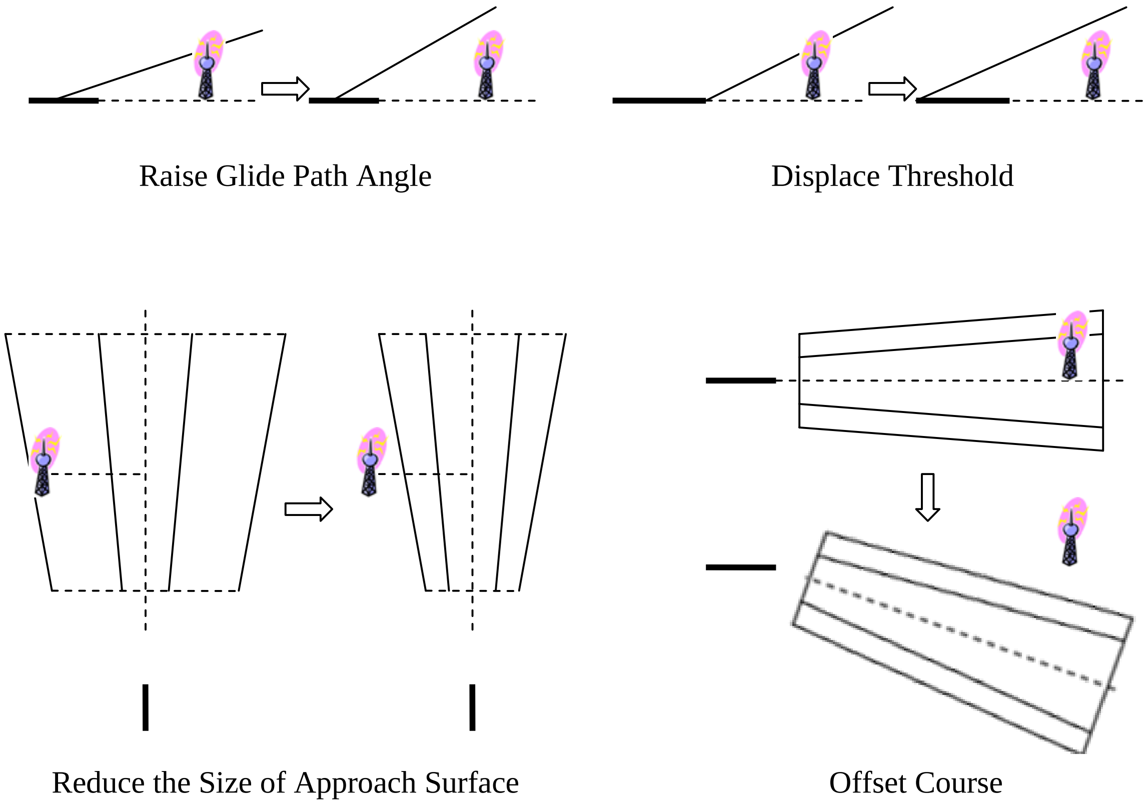

Figure 4-5: Obstacle Clearance Methods.

Raising GPA

The most intuitive way to clear obstacles is to raise the GPA. Recent general aviation pilot simulation research reveals that instrument-rated pilots have difficulty flying stabilized approaches in actual instrument meteorological conditions beyond 5 degrees with current flight deck instrumentation (*9*). SATS aircraft are expected to be equipped with improved navigation displays, but it is uncertain whether these displays could improve the GPA limit in instrument meteorological conditions. The transition from instrument navigation to the acquisition of the runway at the decision height seems to be a limiting factor driving the GPA limit. Anecdotal experience in the same pilot study suggests that under visual meteorological conditions pilots could fly steeper approaches.

The SATS aircraft considered in this study are very light jets, turboprop- and piston-powered aircraft with approach speeds up to 121 knots. For air traffic control purposes these aircraft are classified according to Terminal Area Procedures (TERP) Groups A and B. According to existing FAA rules, the maximum allowable GPA is 6.4 degrees for 81 knots or less, 5.7 degrees for 81 to 90 knots in TERP-A aircraft, and 4.2 degrees for TERP-B aircraft (*5*). Figure 4-6 shows the gain by raising the GPA to clear obstacles at 3,416 airports for both (*a*) base and (*b*) reciprocal approach ends. Base-end results represent runways labeled from 00 to 18, and reciprocal runways include identifiers 19 to 36. By using recent general aviation pilot simulation study results, the 5-degree GPA is chosen as the critical safety threshold in this analysis (*9*). The results in Figure 4-6 indicate that 46% of base-end and reciprocal-end runways studied meet the WQS criteria if the GPA is 3 degrees. However, if the GPA is increased to 5 degrees, the percentage of runways in which the critical obstacle is cleared would increase to 78%. The observed trends for base-end and reciprocal-end runways are very similar. In the rest of this paper, only the results for base-end runways are discussed. However, reciprocal-end runways were also studied.

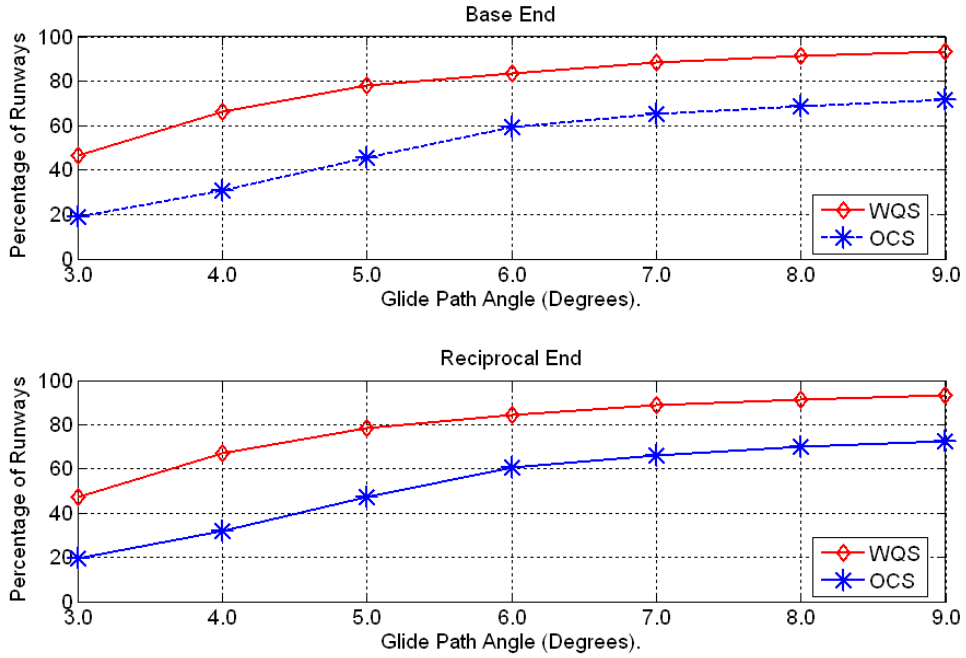

Figure 4-6: Benefits of Raising Glide Path Angle.

Displacing Runway Threshold

Another technique to remove runway obstacles is to displace the runway threshold. A displaced runway threshold of up to 305 m (1,000 ft) is now introduced as an obstacle clearance method. This threshold is coupled with GPAs of up to 5 degrees to test the gains in obstacle clearance. To satisfy runway operability and safety, the remaining runway length is set to be greater than 915 m (3,000 ft), which complies with runway length minima. Displacing the runway threshold offers the most cost-effective way to clear critical obstacles. A displaced threshold has little or no infrastructure investment needs. The percentage of runway ends that benefited from threshold displacement is shown in Figure 4-7. With a 3-degree GPA as an example, 46% of the runways meet the WQS criteria. Displacement of thresholds up to 305 m resulted in 82% of the runways meeting the WQS criteria with at least 915 m of remaining runway. Figure 4-7 also shows that 152-m (500-ft) runway threshold displacement makes 75% of the runways studied compliant with the WQS criteria.

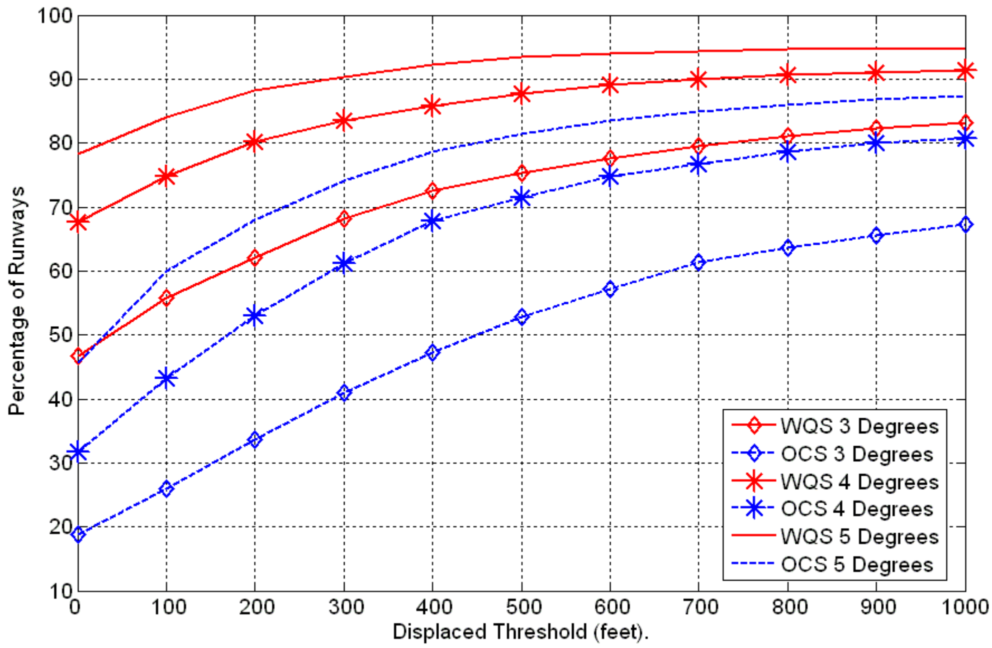

Figure 4-7: Benefits of Displacing the Runway Threshold.

Reducing WQS

By conducting human factors research and applying new display technologies, NASA and NCAM are exploring the possibility of reducing FTEs, which are one of the primary drivers of WQS and OCS size. Large FTEs imply poor navigation-keeping ability in the final approach procedure and thus result in large vertical and lateral allowances in the design of the approach to keep the aircraft away from obstacles. If future pilot studies demonstrated that FTEs could be reduced with improved aircraft controls and displays, the FAA might decide to accept smaller WQS and OCS. This change would only be possible if flight tests confirmed that the level of safety of the operations was not degraded. The foregoing possibility provided the incentive to study obstacle clearance by reducing the size of the WQS from its original dimension to an extreme case, a line with the same slope as the WQS–OCS surface. A reduction factor ranging from 1 to 0 is defined to indicate the normalized size of the WQS. A reduction factor of 1 implies the full size of the WQS. A value of zero implies that the WQS is a line.

The benefits of reducing the WQS are shown in Figure 4-8. With the same 3-degree GPA as the reference point with 46% of runway ends meeting the desired WQS criteria and with a reduction factor of 1 (making FTE zero), 80% of the runway ends would be in WQS compliance. The estimates of potential reductions in FTE here would suggest a reduction factor of no more than 0.5 in real applications. Recent flight simulation results using a Highway in the Sky display supported this conclusion. This finding would make the gains in obstacle clearance relatively small (4%). The results of this analysis indicate that navigation technologies can be made more precise and yet produce modest results in obstacle clearance.

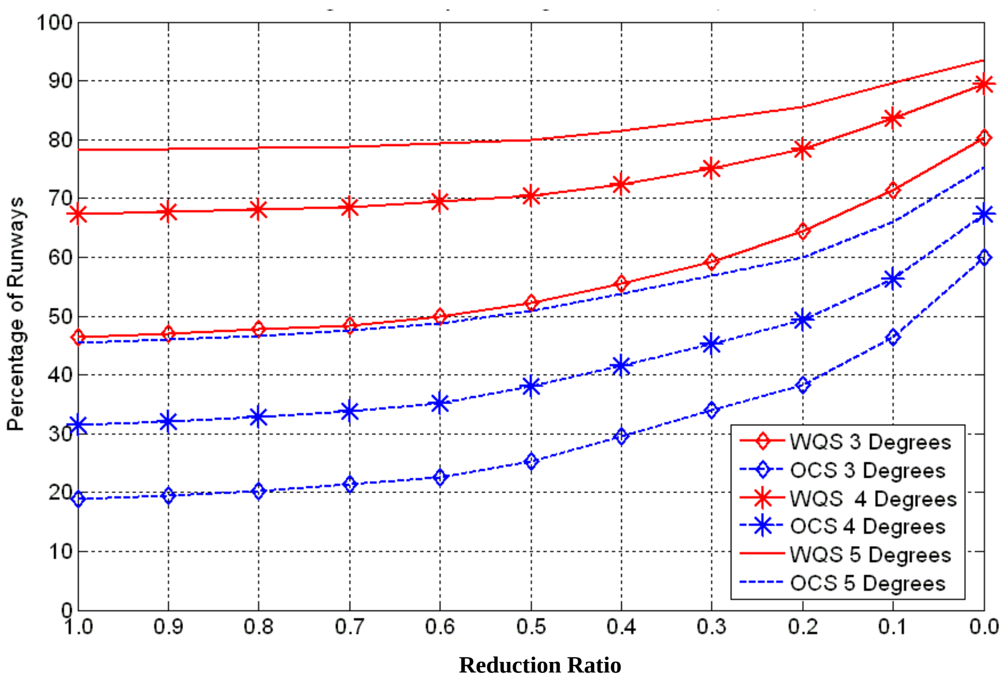

Figure 4-8: Benefits of Reducing the Size of the WQS Surfaces.

Offsetting Final Approach Segment

Current FAA WAAS design approach criteria allow final approach segment offsets of up to 3 degrees provided that other criteria are met (*5*). Offsets up to 5 degrees are studied to illustrate the possible effect of new display technologies and aircraft controls in extending this limit. In this analysis a typical decision altitude (DA) of 91 m

(300 ft) and a distance from the DA to the intersection of the runway centerline approach course of 351 m (1,150 ft) were used to comply with FAA rules.

Figure 4-9 shows the benefit of the offset method. Starting with a 3-degree GPA, if the maximum permissible offset of 3 degrees is applied to 3,416 runway base ends, the proportion of runways complying with WQS criteria increases from 46% (at zero offset) to 65%. If the offset angle is allowed increase to 5 degrees, the proportion of runways meeting WQS criteria would be 70%, a modest improvement but perhaps a worthwhile goal for the SATS LLM technology program.

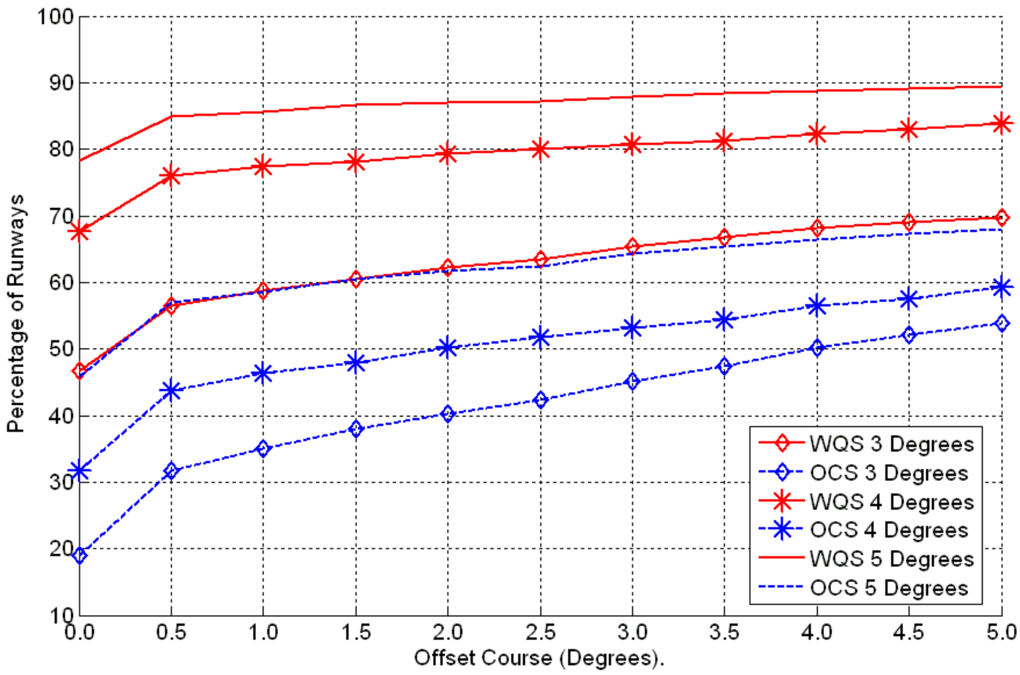

Figure 4-9: Benefits of Offsetting the Final Approach Course.

Results of Standard Precision Landing Minima

The FAA specifies precision approach landing minima contained in FAA Order 8260.48, Area Navigation Approach Construction Criteria (Table 1). On the basis of GPA analysis results obtained in the previous section and the available approach lighting system information obtained from the FAA airport database, the minimum HAT can be obtained for all runway ends studied (see Table 1). HAT defines the DH (in feet) or the minimum descent altitude above the runway touchdown zone elevation (TDZE). HAT defines a critical point in the approach at which a pilot initiates a missed-approach procedure if the runway is not in sight.

Table 4-1: FAA Order 8260.48, Table 2-2B, Standard Precision Landing Minimums (source: FAA).

<table>
<tr><th colspan="2" rowspan="3">Glidepath Angle (with Approach Light Configuration)</th><th rowspan="3">Minimum HAT</th><th colspan="4">Aircraft Category</th></tr>
<tr><th>A</th><th>B</th><th>C</th><th>D & E</th></tr>
<tr><th colspan="4">Minimum Visibility</th></tr>
<tr><td rowspan="3">3.00º – 3.10 º</td><td>★</td><td>200</td><td colspan="4">¾ 4000</td></tr>
<tr><td>#</td><td>200</td><td colspan="4">½ 2400</td></tr>
<tr><td>$</td><td>200</td><td colspan="4">1800</td></tr>
<tr><td rowspan="6">3.11º – 3.30 º</td><td>★</td><td>200</td><td colspan="2">¾ 4000</td><td colspan="2">NA</td></tr>
<tr><td>★</td><td>250</td><td colspan="2">¾ 4000</td><td>1 5000</td><td>NA</td></tr>
<tr><td>#</td><td>200</td><td colspan="2">½ 2400</td><td colspan="2">NA</td></tr>
<tr><td>#</td><td>250</td><td colspan="2">½ 2400</td><td>¾ 4000</td><td>NA</td></tr>
<tr><td>$</td><td>200</td><td colspan="2">1800</td><td colspan="2">NA</td></tr>
<tr><td>$</td><td>250</td><td colspan="2">1800</td><td>½ 2400</td><td>NA</td></tr>
<tr><td rowspan="6">3.31º – 3.60 º</td><td>★</td><td>200</td><td colspan="2">¾ 4000</td><td colspan="2">NA</td></tr>
<tr><td>★</td><td>270</td><td colspan="2">¾ 4000</td><td>1 5000</td><td>NA</td></tr>
<tr><td>#</td><td>200</td><td colspan="2">½ 2400</td><td colspan="2">NA</td></tr>
<tr><td>#</td><td>270</td><td colspan="2">½ 2400</td><td>¾ 4000</td><td>NA</td></tr>
<tr><td>$</td><td>200</td><td colspan="2">2000</td><td colspan="2">NA</td></tr>
<tr><td>$</td><td>270</td><td colspan="2">2000</td><td>½ 2800</td><td>NA</td></tr>
<tr><td rowspan="2">3.61º – 3.80 º</td><td>★</td><td>200</td><td colspan="2">¾ 4000</td><td colspan="2">NA</td></tr>
<tr><td>#</td><td>200</td><td colspan="2">½ 2400</td><td colspan="2">NA</td></tr>
<tr><td rowspan="4">3.81º – 4.20 º</td><td>★</td><td>200</td><td>¾ 4000</td><td colspan="3">NA</td></tr>
<tr><td>★</td><td>250</td><td>¾ 4000</td><td>1 5000</td><td colspan="2">NA</td></tr>
<tr><td>#</td><td>200</td><td>½ 2400</td><td colspan="3">NA</td></tr>
<tr><td>#</td><td>250</td><td>½ 2400</td><td>¾ 4000</td><td colspan="2">NA</td></tr>
<tr><td rowspan="2">4.21º – 5.00 º</td><td>★</td><td>250</td><td>¾ 4000</td><td colspan="3">NA</td></tr>
<tr><td>#</td><td>250</td><td>½ 2400</td><td colspan="3">NA</td></tr>
<tr><td rowspan="2">5.01º – 5.70 º</td><td>★</td><td>300</td><td>1 5000</td><td colspan="3">NA</td></tr>
<tr><td>#</td><td>300</td><td>¾ 4000</td><td colspan="3">NA</td></tr>
<tr><td rowspan="2">5.71º – 6.40 º
Airspeed NTE
80 Knots</td><td>★</td><td>350</td><td>1¼</td><td colspan="3">NA</td></tr>
<tr><td>#</td><td>350</td><td>1 5000</td><td colspan="3">NA</td></tr>
</table>

★ = No Lights # = MALSR, SSALR, ALSF
$ = # Plus TDZ/CL Lights NA = Not authorized

Two sets of HAT minima are presented in Figure 4-10 and Figure 4-11 as a result of different dimensions and obstacle clearance criteria. The first set of results applies to the WQS and the second to the OCS. DA is the sum of HAT and TDZE. Presented in Figure 4-10 and 3-11 is a first-order analysis of the HAT values for every runway because the GPA for the critical obstacle and the approach lighting system at the runway end were the only variables considered in determining HAT. Under real-world conditions, other requirements such as the availability of taxiway, the size of the runway object-free zone and the runway safety area, and runway markings are to be considered whenever the information is available. An airport infrastructure inventory is necessary to understand the current status of the DA and the potential to reach LLM at the 3,416 airports. Unfortunately, an airport database with all the necessary information is not readily available. To understand other factors that drive the current values of HAT, an informal airport instrument approach survey was conducted.

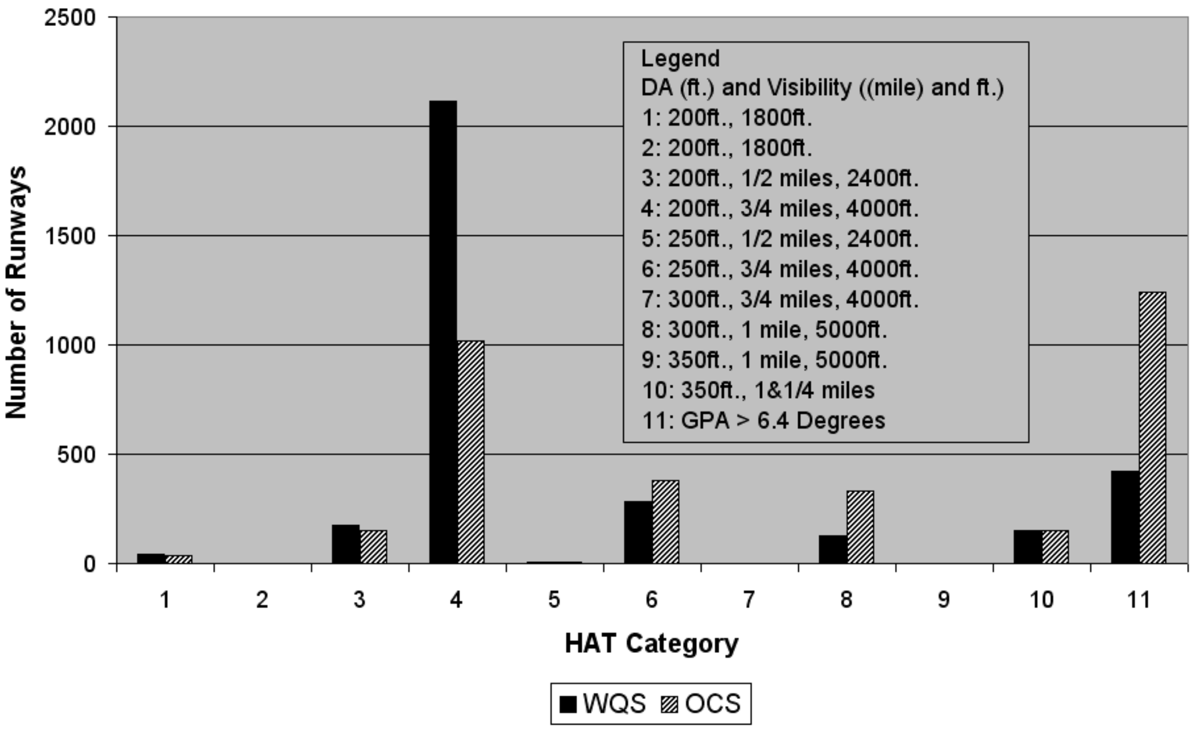

Figure 4-10: TERP-A Height Above Threshold (HAT) for Runway Base End Runway.

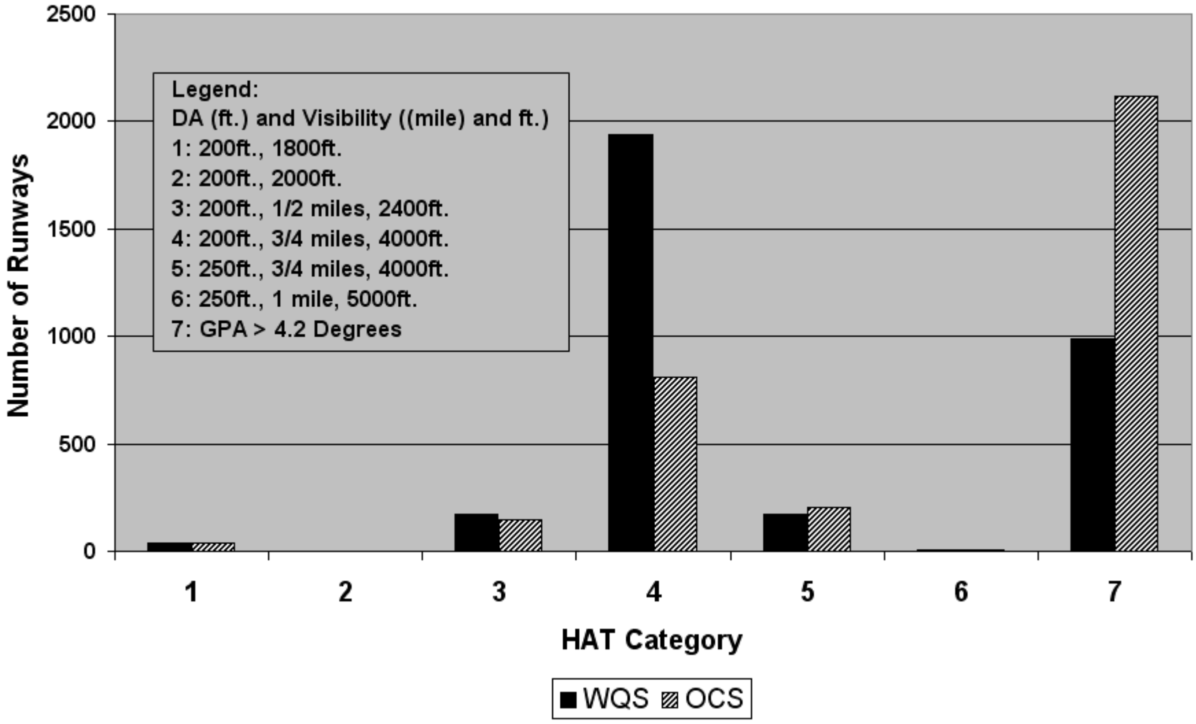

Figure 4-11: TERP-B Height Above Threshold (HAT) for Runway Base Ends.

SATS Airport Instrument Approach Survey

An informal survey was conducted of the top 500 airports from the selected SATS airport set that produces the largest demand according to the Virginia Tech air transportation decision support model (*1*). Approach characteristics surveyed were TERP group, type of approach, DA, RVR, real DA, and RVR considering TDZE, taxiway, and GPA. Among the 500 top SATS demand-producing airports, 85 have visual approach procedures only. The cumulative distribution functions of several TERP-A approach parameters are shown in Figure 4-12, and the TERP-B parameters are similarly distributed. A minimum HAT is assigned to each surveyed runway as its current best minimum (Figure 4-13). The survey acts as a starting point for a more comprehensive airport infrastructure inventory in future research. At the same time it provides useful data on actual approach parameters and how they compare with the minima contained in Table 4-1.

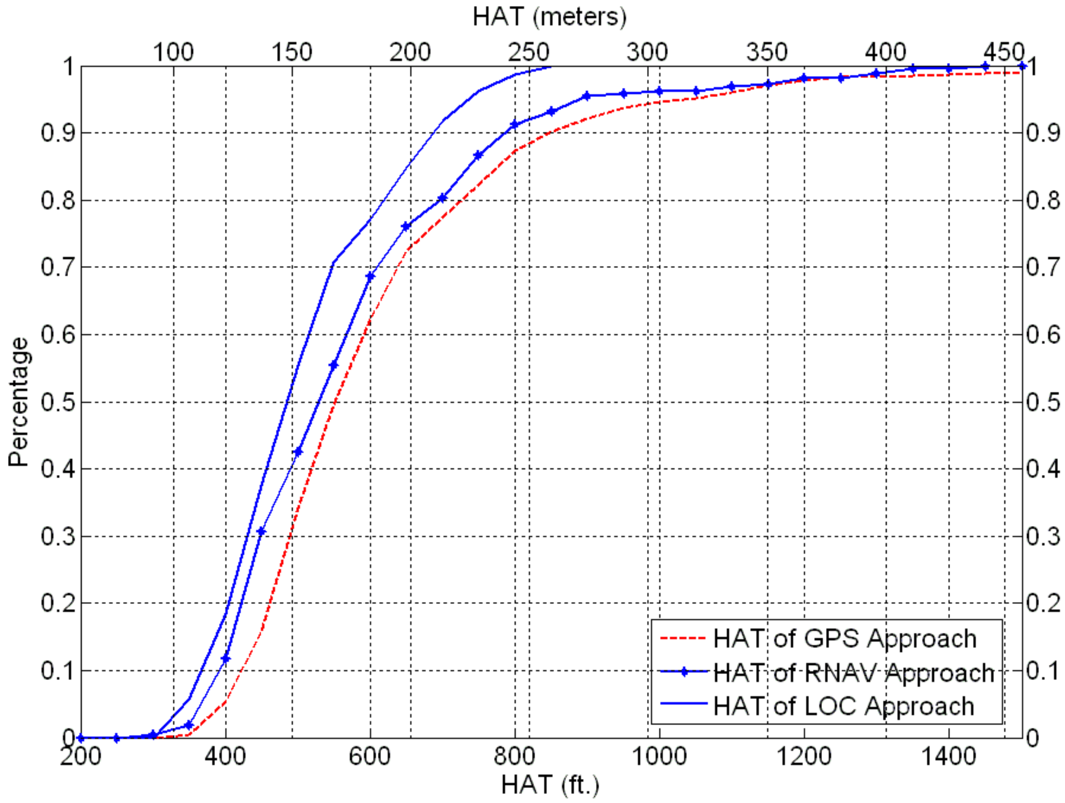

Figure 4-12: Cumulative Distribution Function of TERP-A Instrument Approach HAT Values.

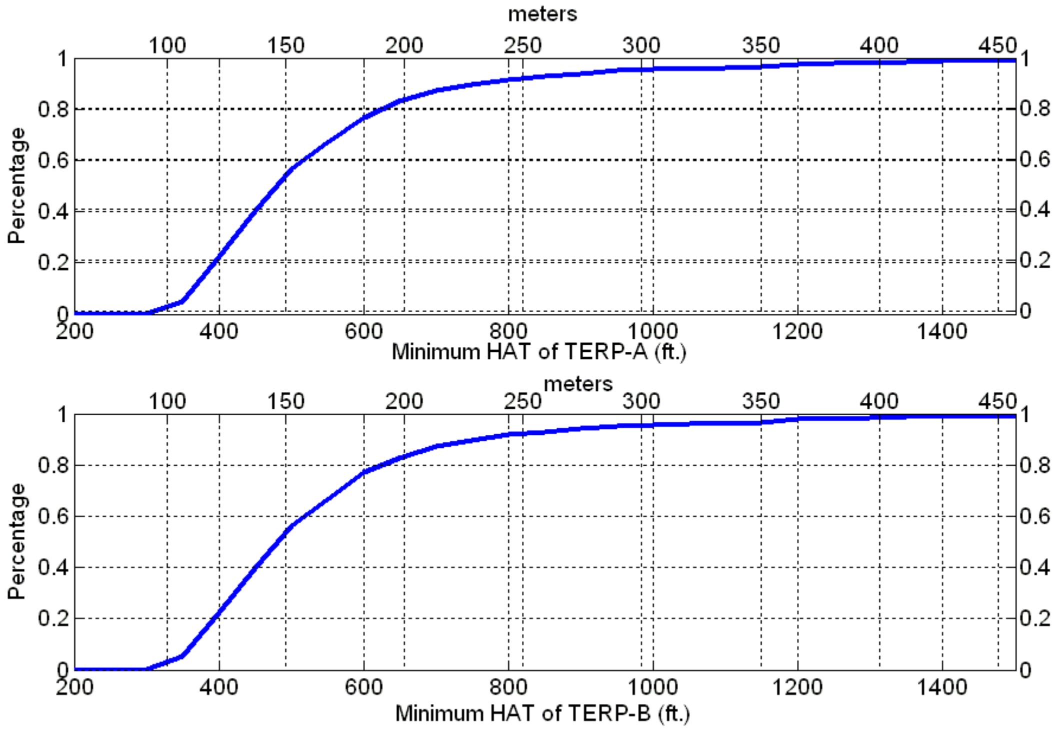

Figure 4-13: Cumulative Distribution Function of the Minimum HAT for Each Runway Surveyed.

Application of Obstacle Removal Methods

The effectiveness of the obstacle removal methods described here might depend on the views of various stakeholders. For example, an airport manager might have different views on how to remove obstacles than a pilot flying the approach or a technical analyst designing the aircraft controls to achieve safe operations for a challenged airport. On the basis of analysis of 3,416 runways in the United States, a family of sequential methods is proposed that rank in order of importance pragmatic approaches to the removal of obstacles. The same sequences can serve as guiding principles to rank LLM research in the SATS Program. The three ranking sequences to achieve obstacle removal are shown below in order from the best to the worst:

• Theoretical benefit: direct removal, raising the GPA, displacing the threshold, reducing the size of the approach surface, and offsetting the approach course;

• Approach safety: direct removal, displacing the threshold, offsetting the approach course, raising the GPA, and reducing the size of the approach surface; and

• Clearance cost: displacing the threshold, direct removal, offsetting the approach course, raising the GPA, and reducing the size of the approach surface.

Caution should be taken in the application of these clearance method priorities. Use of the theoretical benefit approach means that only the gain of applying each method is considered, and costs and safety are neglected. The approach safety procedure analysis is airport site-specific. For example, runway length may raise the concern of threshold displacement safety, and the terrain around the airport may prohibit offsetting the approach course. The clearance cost approach should include the removal cost of the obstacles and the associated ground and airborne equipment costs.

The obstacle removal methods explained were applied to thousands of airports nationwide by using only the critical obstacle available for each runway end. The results from this analysis provide an initial assessment of the challenges in providing all-weather landing capabilities for the 3,416 airports. Once this assessment has been performed, a more detailed design procedure is to study each individual airport considering more detailed obstacle information (i.e., multiple objects) and the terrain information around each airport. A case study of Blacksburg Airport is highlighted in the following section to illustrate the more detailed study of a GPS-WAAS precision approach procedure.

Case Study: Blacksburg Airport (Virginia Tech)

Blacksburg, Virginia, is located in mountainous terrain that presents a challenge to design precision approaches. The airport has a partial instrument landing system approach with localizer information. To design an approach procedure, a U.S. Geological Survey 1:250K digital elevation model containing detailed topographic representation was used. Four existing approach fixes (PULASKI, ZOOMS, HAWTO, and SUNNY) were chosen for Runway End 12, and four pseudo-fixes were chosen for Runway End 30 to satisfy the FAA approach design criteria. Obstacle data were gathered from the National Geodetic Survey Universal Data Delivery Format for Runway End 12 (*10*). A resultant 4.2-degree GPA is necessary for this approach considering all the obstacles instead of the 3-degree approach obtained using the critical obstacle alone. The approach surface together with the terrain information for Runway 12 is shown in Figure 4-14. The analysis shows that a GPA of 4.2 degrees would meet the WQS clearance requirement for the three approach segments. The lesson learned from this detailed analysis is that a national-level analysis conducted with all four obstacle clearance methods provides the best-case scenario of the challenges to improve LLM capability at the airports studied.

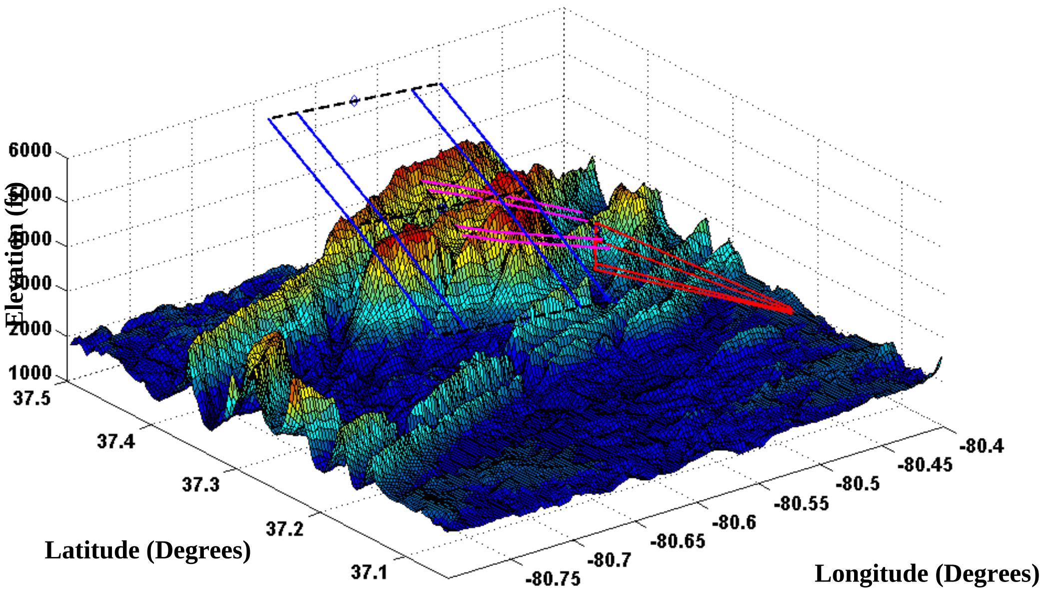

Figure 4-14: WAAS WQS Surface with Terrain Information for Runway 12 at BCB Airport.

Conclusions and Future Research

Four methods were presented to mitigate the effects of existing obstacles at airport runways using the critical object for every runway end. Other than direct removal of runway obstacles, the most promising methods to maximize WAAS approach compliance are raising the GPA and displacing the runway threshold. The SATS Program should continue investigating the safety implications of these methods.

A national-level study was done using the critical obstacle of every runway to identify challenges in providing LLM capabilities for the SATS Program. A detailed study of a challenged airport provided insight about the variations between the critical object analysis and more detailed consideration of terrain and obstacle databases for an airport. The obstacle removal methods presented here offer a first step to utilizing GPS and WAAS.

A further study of LLM costs and benefits is needed to find an optimum number of WAAS-capable airports to be part of the SATS Program that justifies infrastructure investment. SATS demand at candidate airports, weather patterns, travel time savings, ground and airborne WAAS equipment cost, and airport infrastructure inventory are the basic elements of a future cost–benefit model to identify the optimum set of airports needed for the SATS Program. The cost–benefit analysis requires knowledge of the demand function expected at the airport (to derive user benefits) and the infrastructure costs of every proposed LLM technology (both onboard and on the ground).

The SATS LLM operational technologies under development would provide safety benefits to the aviation community even if the SATS Program is not able to convince many passengers to switch to the air mode. The safety implications of LLM technologies should be studied using piloted and computer simulation studies. These technologies could well be one of the best legacies of the program.

References

1. Trani, A. A., H. Baik, H. Swingle, S. Ashiabor, N. Hinze, A. Seshadri, and K. Murthy. *SATS Transportation Systems Analysis Model.* Virginia Polytechnic Institute and State University, Blacksburg, Dec. 2003.

2. *GPS/WAAS Approaches.* FAA, U.S. Department of Transportation. http://gps.faa.gov/CapHill/approachGPS-WAAS-text.htm. Accessed May 2004.

3. Crane, H. L., and S. V. Massimini. *A Tool for Estimating Airport Access Benefits from GPS-Based Instrument Approaches.* AIAA 2001- 4236. American Institute of Aeronautics and Astronautics and MITRE Corporation, 2001.

4. *Landing Facility Database.* FAA, U.S. Department of Transportation, March 2004.

5. *Area Navigation (RNAV) Approach Construction Criteria.* Federal Aviation Administration Order 8260.48. FAA, U.S. Department of Transportation, April 8, 1999.

6. GPS Product Team. *Wide Area Augmentation System.* 2003 FRP User Conference, FAA, U.S. Department of Transportation, May 19, 2003.

7. *Civil Utilization of Global Positioning System.* Federal Aviation Administration Order 8260.38A. FAA, U.S. Department of Transportation, April 5, 1995.

8. National Aeronautical Charting Office. *Terminal Procedures Publications.* FAA, U.S. Department of Transportation, July 10, 2003.

9. Lancaster, J. A., J. J. Saleem, B. M. Kleiner, G. S. Robinson, and J. G. Casali. Preliminary Study on the Effects of Approach Angle and Lower Landing Minimum Level on Pilot Performance in a Low-Fidelity Static Aircraft Simulator. Presented at 12th Biennial International Symposium on Aviation Psychology, Dayton, Ohio, 2002.

10. *Universal Data Delivery Format.* National Geodetic Survey (NGS) Aeronautical Survey Program, January 28, 2004. www.ngs.noaa.gov/AERO/aero.html. Accessed March 2004.

General Demand Estimation Based on System Characteristics

Technologies such as Lower Landing Minima can increase the accessibility of airports without precision approach capability. The following general demand impact scenarios have been evaluated to assess impacts of LLM.

Different airport set scenarios were executed using Transportation Systems Analysis Model (TSAM) to estimate demand impacts. The airport set evaluated includes a full SATS-compatible airport set, three airport sets that allow less than 5, 4, and 3 degrees Glide Path Angle (GPA) WAAS approach respectively, and an Instrument Landing System (ILS) equipped airport set. WAAS approach improves dispatch reliability to enable airport to operate under near all weather. Figure 4-15 illustrates the travel time savings when dispatch reliability improves.

Figure 4-15: Travel Time Savings (Year 2010).

Improvements of dispatch reliability will not only save travel times for current users, but also induce additional user from other modes due to attractiveness of the utility function. Figure 4-16 shows the induced demand due to reliability improvements.

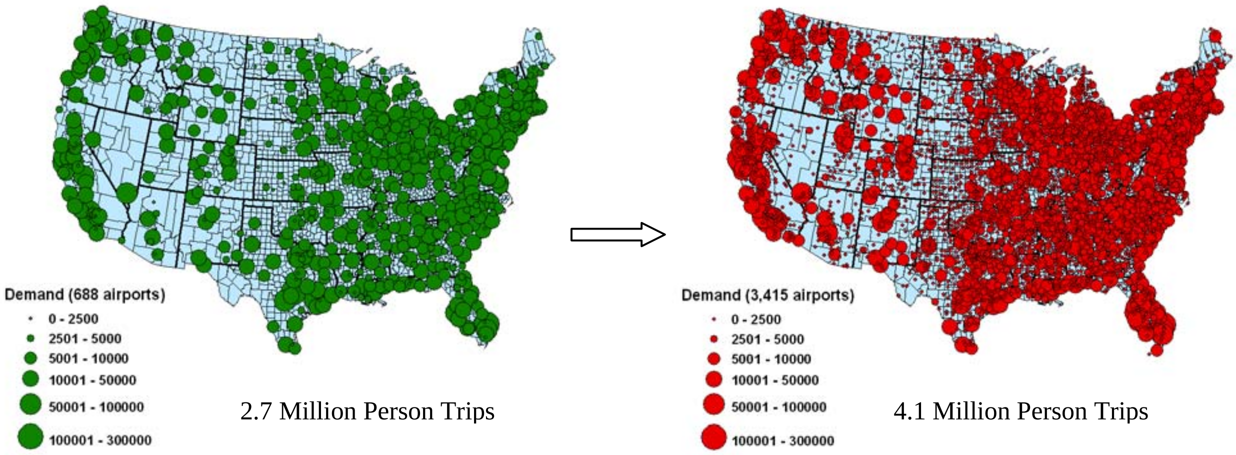

Figure 4-16: Demand Estimation (676 airports vs. 2,286 airports).

Figure 4-17 shows the demand estimation with different GPA configuration.

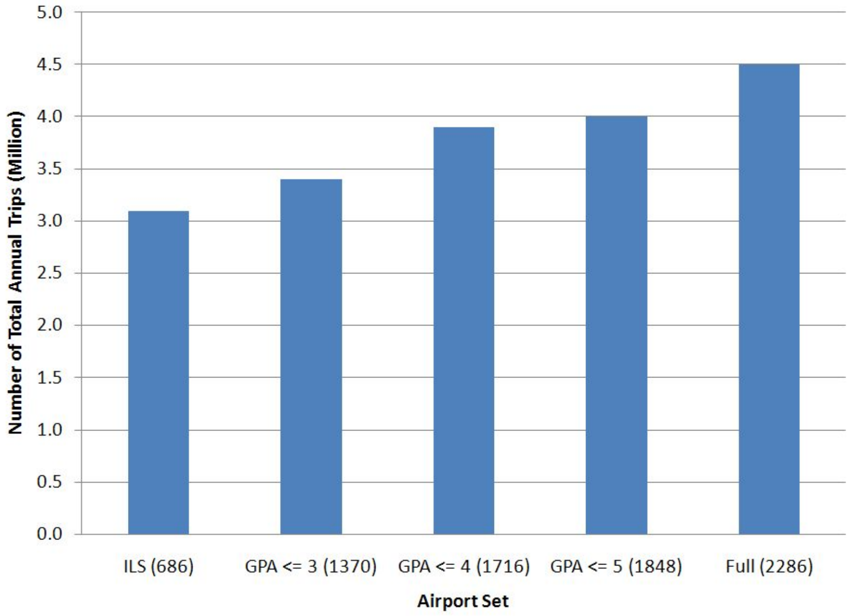

Figure 4-17: Demand Estimation (with Different GPA Sets).

In addition to travel time savings, researches have indicated a total 11% of ticket price decrease due to the increase of load factors.

These findings can be applied in further cost benefit analysis employing value of time concept and ticket price reduction. The ticket price reduction effect will not be elaborate in the dissertation.

Cost and Benefit Estimates

The ground costs for LLM capacity considered in this dissertation include taxiway construction, approach lighting system installation, approach fixes installation, obstruction removal to maintain proper Runway Protection Zone (RPZ), Object Free Zone (OFZ) and Runway Safety Area (RSA) dimensions, runway markings, and environmental costs. The airborne costs are primarily from avionics upgrade costs to equip GPS WAAS enabled receivers and displays. Benefits are driven by travel time savings introduced by LLM capability. Table 4-2 lists the infrastructure investments required by FAA. Various cost estimates are obtained from the literature. The following section will discuss each cost and benefit components.

Table 4-2: Approach Procedure with Vertical Guidance Requirements (FAA, 2006).

Visibility Minimums	< ¾- Mile	< 1 - Mile	1 – Mile	>1 Mile
Height Above Touchdown	250	300	350	400
TERPS Paragraph 251	34:1 clear	20:1 clear	20:1 clear or penetrations lighted for night minimums (see AC 70/7460-1)	
Precision Object Free Zone	Required	Recommended		
Airport Layout Plan	Must be on approved ALP			
Runway Marking	Non-precision (precision recommended)		Non-precision	
Runway Edge Lights	HIRL/MIRL		MIRL/LIRL	
Parallel Taxiway	Required		Required	

Approach Lights	Required – ODALS/MALS, SSALS	Recommended
Runway Design Standard	APV OFZ Required	

Taxiway construction

Taxiway construction represents one of the major infrastructure investments. Table 4-3 lists the findings on the cost of taxiway constructions. It shows bigger airports usually require more investment on taxiway (Tulsa and Spokane International) whereas investments on smaller airports range from 1 million to 2 million.

Table 4-3: Taxiway Investment Examples.

Airport Name	Total Investment	Dimension (ft)	Unit Cost
Tulsa International[1]	$8,000,000	4,000	$2,000
Belfast Regional[2]	$1,800,000	4,000	$450
Redding Municipal[3]	$1,300,700	4,200	$310
Spokane International[4]	$7,000,000	9,000	$778
Wickenburg Municipal[5]	$944,000	6,000	$157
Savannah Airport[6]	$971,655	N/A	N/A
Dare County Regional[7]	$245,083	700	$350
	$505,674	1,600	$316
	$416,455	1,500	$278

The estimates above are obtained online. In the literature, a comprehensive cost-benefit analysis of taxiway construction (Daniel, J.I., 2002) suggests an investment of 6 million for taxiway F at New Castle Airport (ILG). This is consistent with the investment figures at bigger commercial airports. For LLM analysis, it is assumed that the unit cost is $300 per foot in 2008.

[1] Ginger Shepherd, 2006.
[2] City of Belfast, 2006
[3] Redding Municipal Airport Bidding Tabulation, 2007
[4] Emily Brandler, 2006
[5] Wickenburg Municipal Airport Runway Expension Plan, 2006
[6] Lori Lynah, 2007
[7] Dare County Regional Airport Master Plan Update, 2005

In addition to initial construction, other costs including pavement rehabilitation and overlay should be considered as well. The Dare County Regional Airport analysis estimates are presented in Table 4-4:

Table 4-4: Taxiway Strengthen, Overlay and Rehabilitation Costs.

Project	Dimension	Cost	Unit Cost ($ / ft^2)
Strengthen / Overlay Taxiway 'A'[8]	5,600' x 35'; 60,000 lbs	$1,700,400	$8.68
Strengthen / Overlay Taxiway 'C'[8]	300' x 50'; 60,000 lbs	$130,260	$8.68
Strengthen / Overlay Taxiway 'D'[8]	300' x 50'; 60,000 lbs	$60,840	$4.06
Strengthen / Overlay Taxiway 'E'[8]	300' x 50'; 60,000 lbs	$60,840	$4.06
Strengthen / Overlay Taxiway 'G'[8]	300' x 50'; 60, 000 lbs	$60,840	$4.06
Strengthen / Overlay Taxiway 'H'[8]	300' x 100'; 60,000 lbs	$174,200	$5.81
Rehabilitate / Overlay Taxiway 'B'[8]	1,500' x 50'; 20,000 lbs	$379,925	$5.07
Rehabilitate / Overlay Taxiway 'C-N'[8]	700' x 40'; 20,000 lbs	$143,325	$5.12
Overlay Taxiway 'B'[9]	3200' * 40'	$700,000	$5.47

Table 4-4 shows the average strengthening, overlay and rehabilitation cost ranges from $4 - $9 per square foot. Thus this cost is assumed to be $7 per square foot in this study.

The instrument approach survey of the top 500 GA airports indicates about 23% of the surveyed airports do not have a taxiway. Therefore we assume 25% of the GA airports considered do not have a taxiway. Figure 4-18 (FAA Landing Facility Database, 2005) shows the runway length distribution. The mean of the runway set is 5,280 ft and median 5,000 ft. So an average of 5,000 ft long taxiway is assumed.

[8] Dare County Regional Airport Master Plan Update, 2005
[9] The Broward County Aviation Department, 2006.

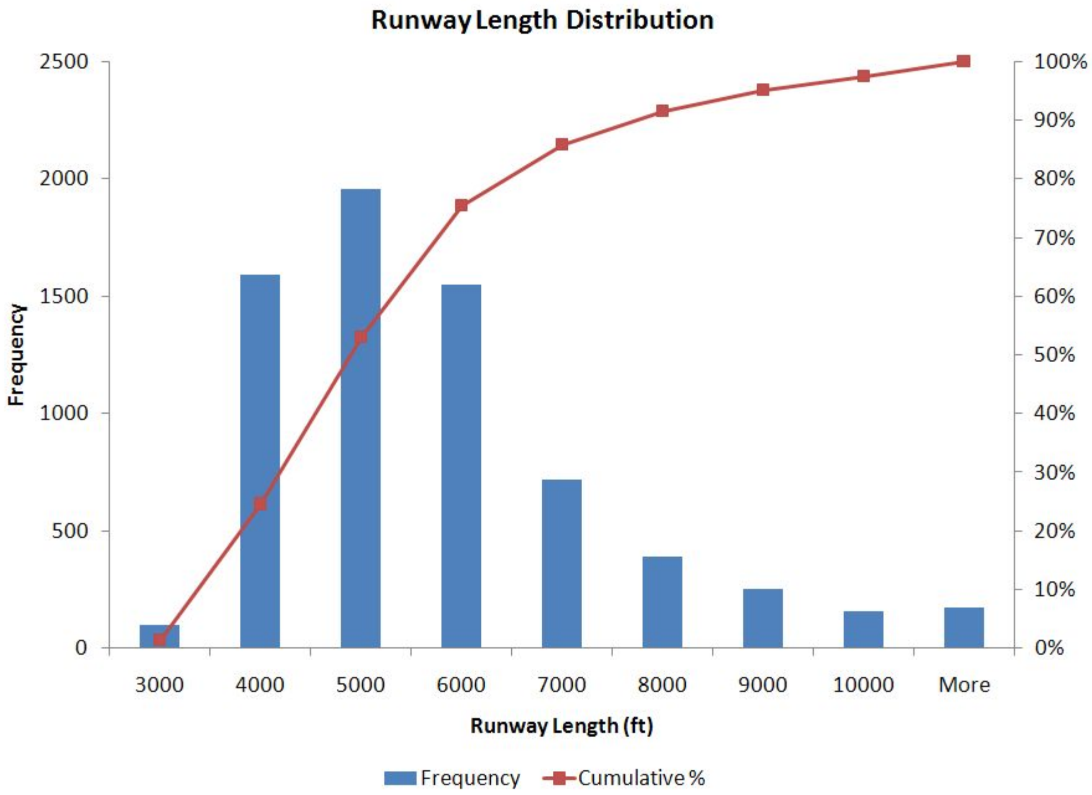

Figure 4-18: Runway Length Distribution.

Approach Lighting System

In order to provide precision approach, the airport must meet a minimum runway and approach lighting requirement. The Medium Intensity Runway Light (MIRL) is required for landing minima less than ½ mile and 200ft. The Medium Intensity Approach Lighting System with Runway Alignment Indicator (MALSR) is required for ½ mile and 200ft minima and recommended for ¾ mile and 300 ft minima. Other lighting might be needed as specified by TERPS such as Simplified Short Approach Lighting System (SSALS), Approach Lighting with Sequenced Flashing Lights (ALSF), and Approach Lighting System (ALS). For this analysis, cost for installing MIRL and MALSR will be considered. In addition, a Precision Approach Path Indicator (PAPI) is needed to provide visual vertical descend guidance information during the approach to the runway.

A comprehensive instrument approach procedure study conducted by the Nebraska Department of Aeronautics (NDOA) offers useful insights on the cost of approach lighting system (NDOA, 1999). The study suggests the equipment costs for MIRL, MALSR, and PAPI are $80,000, $200,000 and $30,000 in Year 1999 dollars.

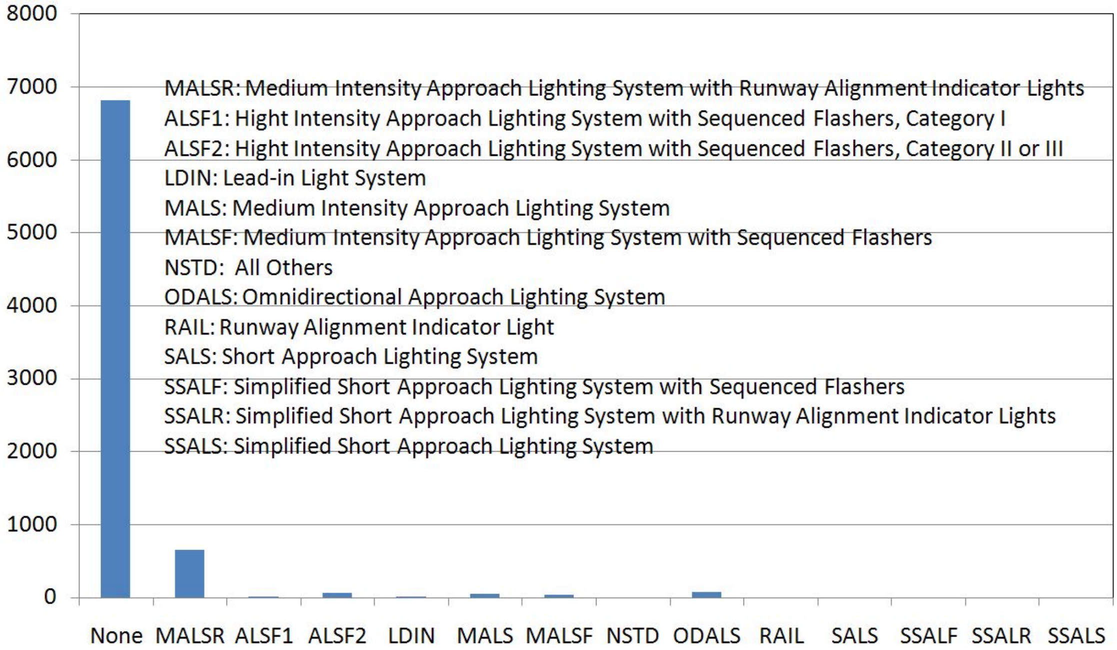

Figure 4-19: Approach Lighting System Availability (7,764 Runways at 2,286 Airports).

Figure 4-19 shows the approach lighting system availability of the full 2,286 SATS compatible airports. Among the 2,286 airports, 547 airports have at least one MALSR system. This result indicates 75% of the airports would require new MALSR installation if precision approach were offered. It is assumed that the same amount of MIRL is needed as well. Since there are 686 airports equipped with Instrument Landing System (ILS) where MALSR and MIRL should be deployed, most of the remaining SATS airports will need MALSR and MIRL. According to the FAA, there are approximately 900 MALSRs in the NAS (FAA, 2008).

Figure 4-20 shows the status of runway glide slope indicator availability. It indicates that 55% of the current runway does not have the PAPI Indicator. Among the 2,286 airports, 1,425 airports have at least one PAPI indicator. Therefore apart from the 676 ILS airports, there are approximately 750 SATS airports with PAPI available. It means around 50% of the remaining SATS (apart from ILS airports) airports will need

PAPI installed. This number is very different from FAA report (FAA, 2008) which indicates there are 693 PAPIs in the NAS between 2005 and 2006.

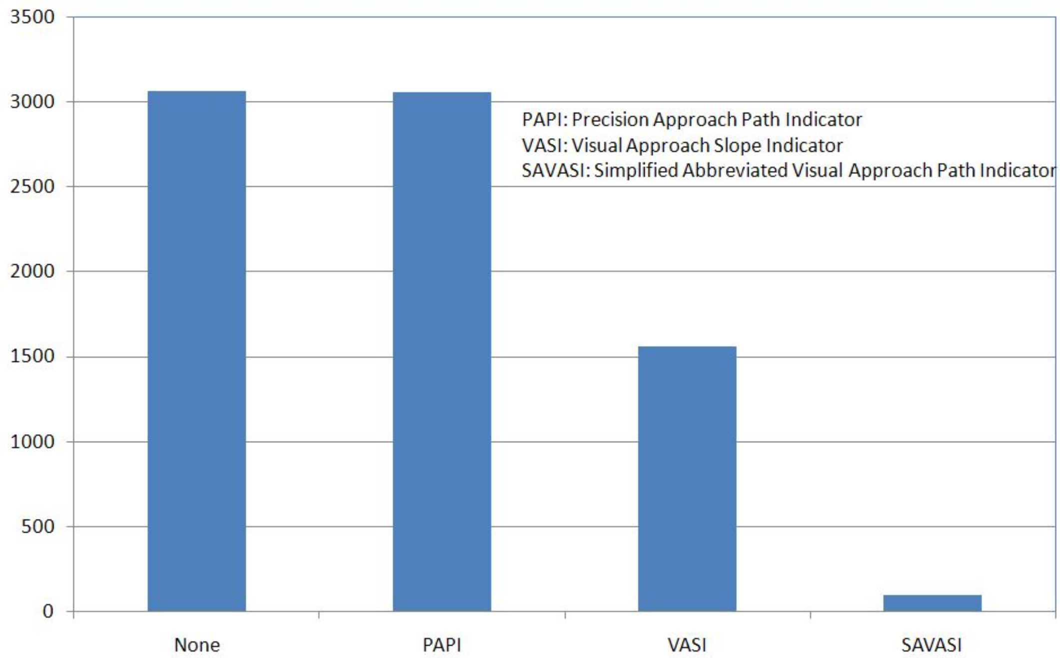

Figure 4-20: Runway Glide Slope Indicator Availability.

Approach Fix Installation and WAAS Approach Publication

GPS WAAS precision approach procedure requires installing a final approach fix. Installing a Fan Marker on the final approach course to set up the final approach fix costs around $25,000 according to the NDOA study (NDOA, 1999). AOPA estimates the cost of mapping and publishing WAAS approach is around $50,000 per runway (AOPA, 2006). This cost should include approach fix installation and procedure design.

RPZ / OFZ / RSA

The cost of RPZ / OFZ / RSA is the most expensive investment in the estimation. The amount of investment depends on many parameters such as the dimension of current RPZ/ OFZ/ RSA, terrain, land acquisition, obstruction removal, and environmental concerns. Figure 4-21 shows the FAA RPZ dimension requirements. Some of the RPZ / OFZ /RSA construction costs can be found in the literature (Table 4-5).

Approach Visibility Minimums 1/	Facilities Expected To Serve	Dimensions			
		Length L Feet (meters)	Inner Width W_1 feet (meters)	Outer Width W_2 feet (meters)	RPZ acres
Visual And Not lower than 1-Mile (1 600 m)	Small Aircraft Exclusively	1,000 (300)	250 (75)	450 (135)	8.035
	Aircraft Approach Categories A & B	1,000 (300)	500 (150)	700 (210)	13.770
	Aircraft Approach Categories C & D	1,700 (510)	500 (150)	1,010 (303)	29.465
Not lower than ¾-Mile (1 200 m)	All Aircraft	1,700 (510)	1,000 (300)	1,510 (453)	48.978
Lower than ¾-Mile (1 200 m)	All Aircraft	2,500 (750)	1,000 (300)	1,750 (525)	78.914

1/ The RPZ dimensional standards are for the runway end with the specified approach visibility minimums. The departure RPZ dimensional standards are equal to or less than the approach RPZ dimensional standards. When a RPZ begins other than 200 feet (60 m) beyond the runway end, separate approach and departure RPZs should be provided. Refer to Appendix 14 for approach and departure RPZs.

Figure 4-21: Runway Protection Zone Dimension (Source: FAA AC 150/5300-13).

Table 4-5: RSA Construction Cost.

Airport	Cost
New Castle Airport[10]	$110,000
Phoenix Deer Valley[11]	$5,000,000
Hector International[12]	$3,000,000
Morgantown[13]	$4,000,000

[10] Delaware River & Bay Authority, 2008
[11] Arizona Department of Transportation – Aeronautics Division, 2006
[12] Hector International Airport, 2008
[13] City of Morgontown meeting minutes, 2006

The Engineered Material Arresting Systems (EMAS) offers an alternative for full RSA. FAA Order 5200.9 (FAA, 2004) describes in detail the life cycle cost-benefit analysis to compare the financial feasibility of a full RSA and an EMAS system. This dissertation will not elaborate on EMAS and therefore it is assumed full RSA is always more cost-effective than EMAS.

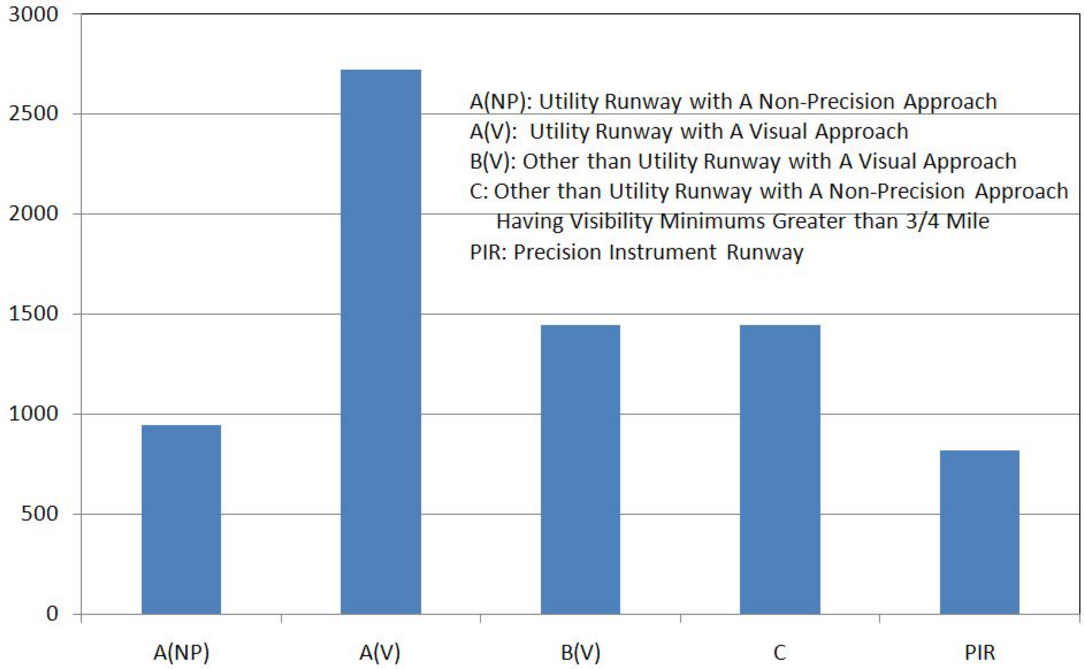

Figure 4-22: Number of Runways for Each FAR Part 77 Category.

There are 614 airports with at least one runway used as precision instrument runway. There are 535 airports with the best available runway categorized as ‘C’. This suggests that almost all the GA airports except ILS airports will need RPZ/OFZ/RSA improvement (Figure 4-22).

Pavement Marking

Figure 4-23 shows the current runway marking conditions. Non-precision runway marking is sufficient for GPS WAAS approach (FAA AC150/5300-13, Change 10, 2006). There are 1,842 airports with at least one runway marked as precision or non-precision instrument runway. It implies only about 400 SATS compatible airports need additional marking.

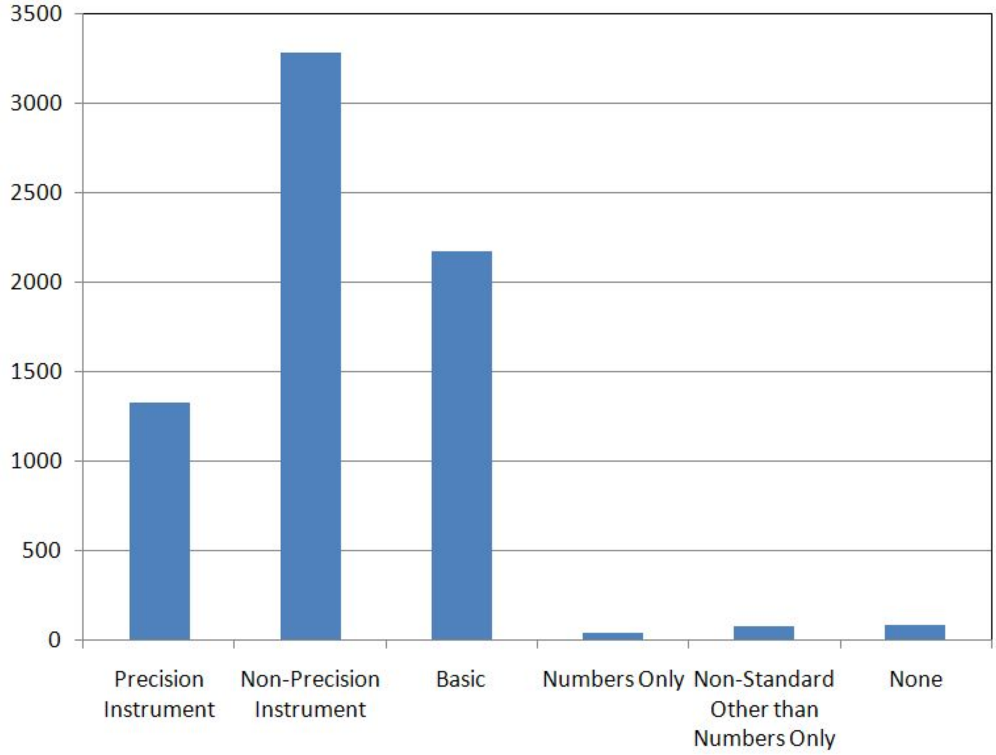

Figure 4-23: Runway Marking Conditions.

An airport master plan study (Danielson Airport, 2007) estimates the cost of non-precision runway marking to be $20,000. This is a relatively small investments compared with other costs.

Maintenance cost

Maintenance cost for taxiway can be found in Daniel's comprehensive cost-benefit study (Daniel, J.I., 2002). The following maintenance costs are considered:

1. Light bulb replacement - $400 / year
2. Center-line maintenance - $100 / year
3. Snow removal - $300 / year
4. Mowing around lights - $480 / year
5. In-ground lights - $300 / year
6. Ramp sign lights (assuming 4 signs) - $1,000 / year
7. Total $2,580 / year, (3,872 in 2008 dollars)

The same paper considers cracks seals as well. The linear cracks should be sealed every two years after the 10th year of construction. It is estimated as $300 for the first year, $500 the third and $600 the fifth and so on. Those costs should not exceed $1,000. It can be argued that runway maintenance cost will increase as a result of more aircraft activities. It requires further analysis and will not be covered in the cost-benefit analysis.

Airborne equipment

The Garmin GNS-480 is the first Gamma-3 WAAS-certified GPS/Comm navigator which costs around $10,750 (Wikipedia, 2008). Its recently upgrade GNS-530 represents the most up-to-date all-in-one GPS/Nav/Comm avionics. The price of a GNS-530 unit is around $15,000 to $20,000 plus $500 per year subscription fee for Jeppesen database updates (Sarasota Avionics International, 2008) (Philip Greenspun, 2002), .

Figure 4-24: Garmin GNS-530.

The fleet size for different scenarios is estimated by TSAM (assuming 15% repositioning flights). Table 4-6 shows the TSAM VLJ fleet estimates for each scenario. Table 4-7 compares the TSAM estimates with other VLJ fleet projections in the industry.

Table 4-6: Fleet Size Estimation using TSAM.

Year	ILS Case	LLM Case	Top 1,200 Case	Full Case
2008	1,724	2,259	2,259	2,556
2009	1,783	2,318	2,318	2,615
2010	1,843	2,378	2,378	2,674
2011	1,902	2,438	2,496	2,694
2012	1,902	2,556	2,556	2,853
2013	1,961	2,615	2,615	2,913
2014	2,020	2,674	2,674	3,031
2015	2,080	2,734	2,734	3,091
2016	2,140	2,853	2,853	3,209
2017	2,199	2,913	2,913	3,269
2018	2,318	3,031	3,031	3,388
2019	2,378	3,091	3,091	3,510
2020	2,438	3,209	3,150	3,566
2021	2,496	3,269	3,269	3,685
2022	2,615	3,388	3,388	3,804
2023	2,675	3,510	3,510	3,923
2024	2,734	3,646	3,646	4,041

Table 4-7: Comparison with Other Estimation (CRA, 2006).

Source	Year	Annual Production Rate	Total Fleet Size
TSAM	2014	N/A	3,031
FAA	2017	450-500	4,950
Rolls Royce	2024	395	7,500
Honeywell	2016	500	4,500 – 5,500
Teal Group	2014	140	1,265
Forecast International	2014	386	3,476

Travel Time Savings

The most important benefit brought by LLM capability is the travel time savings. By operating from small to medium community airports, the SATS system frees from the constraint of the hub-spoke system currently employed by commercial airlines. Therefore passengers can enjoy direct flights from origin to destination. The resultant travel time savings compose the primary benefit of the SATS system. Economic impact brought by aviation activities are not considered in this analysis.

Using TSAM model, air taxi travel time savings can be assessed when different airport sets are available.

Figure 4-27 and Figure 4-28 demonstrate travel time savings for different airport set (Business trip and Non-Business trip).

The benefits of travel time savings can be converted to monetary terms using the Value of Time (VOT) concept. FAA published a guideline on economic values used in airport cost-benefit analysis. The Value of Time is specified in the guideline as $45 per hour for business trip and $31.5 per hour for non-business trip (FAA, 2004) in Year 2000 dollars (Table 2-3). Figure 4-25 and Figure 4-26 illustrate the inflation rate reported by the Bureau of Labor Statistics and Woods & Poole economics data. The Woods & Poole data is used in the trip generation and distribution module of the TSAM model. Therefore to be constant, an average inflation rate of 3% observed from Woods & Poole is applied. A recommended discount rate of 7% is used (FAA, 1999).

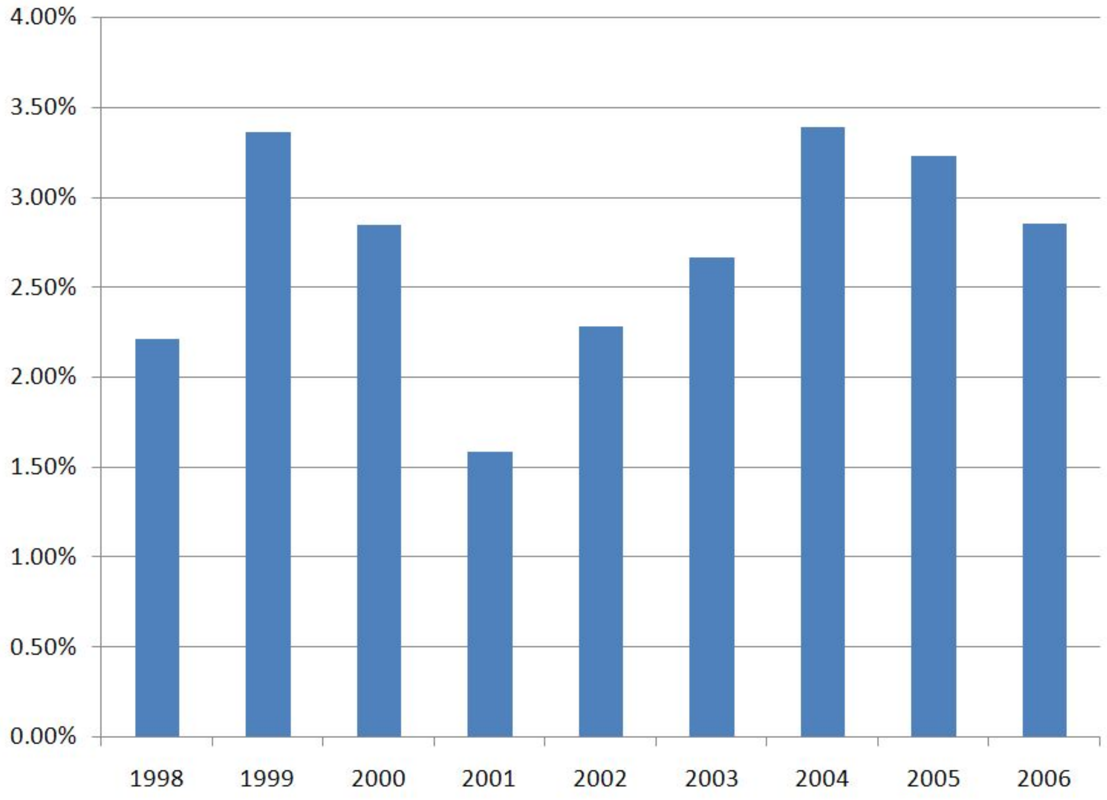

Figure 4-25: Consumer Price Index (CPI).

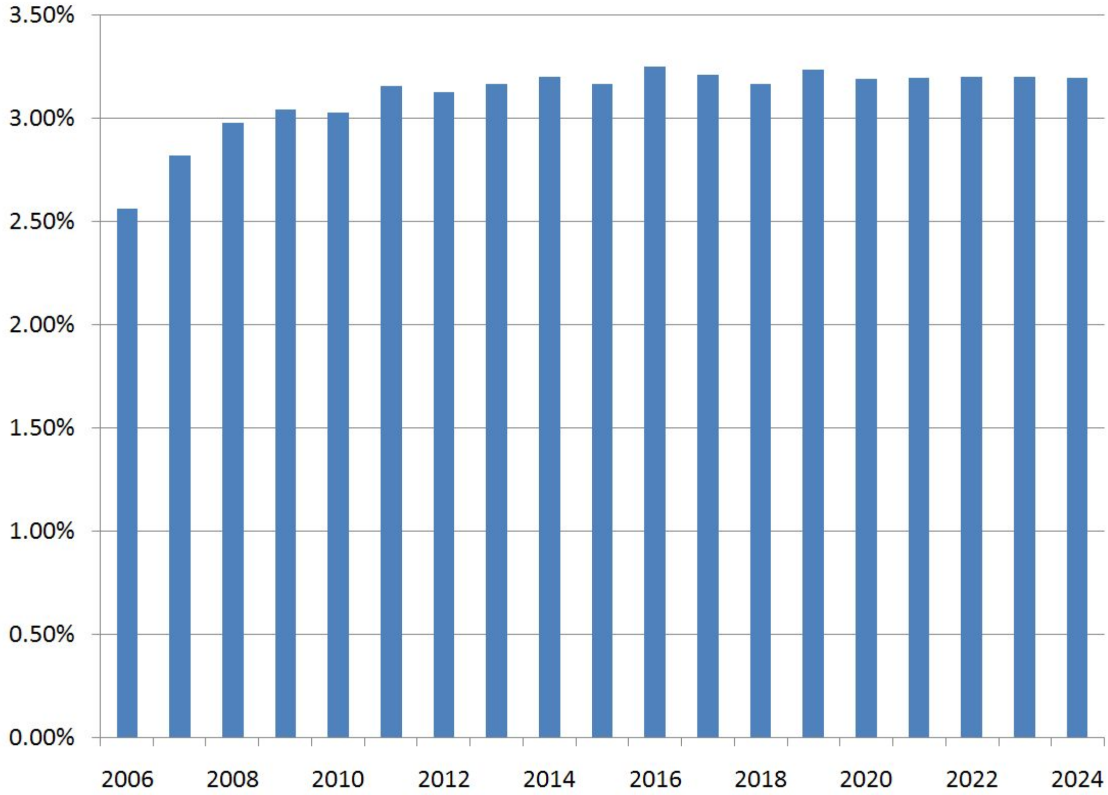

Figure 4-26: Inflation Rate Used in Woods & Poole Economics Data.

Several scenarios are analyzed. The first scenario assumes only ILS precision approach equipped airports are available for SATS traffic (676 airports). The second scenario utilizes the top 1,200 airports that generate the highest demand based on the TSAM estimation. In the third study, the number of airports increases to 1,868 airports where WAAS approach is feasible assuming Glide Path Angle is less than 5 degrees, displace threshold less than 1,000 feet and offset course less than 5 degrees. In the last case, all the SATS compatible airports are used (2,286 airports). The SATS demand and travel time savings of each scenario are estimated using TSAM.

Figure 4-27: Travel Time Savings with Different Airport Set – Business Trip.

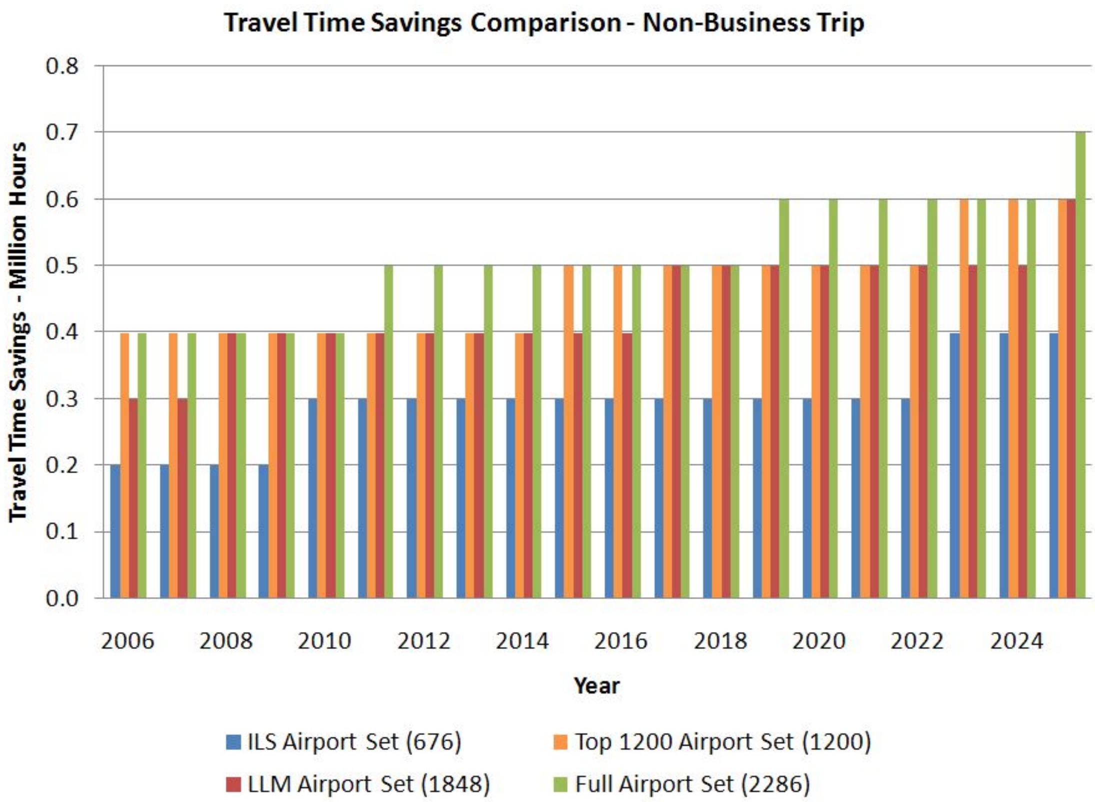

Figure 4-28: Travel Time Savings with Different Airport Set – Non-Business Trip.

Base Year, Salvage Value, and Life Cycle

It is assumed that there is no salvage value after the program.Base year is assumed to be year 2008 and the life cycle to be 15 years.

Cost and Benefit Estimates

VOT analysis

The Value of Time is specified in the guideline as $45 per hour for business trip and $31.5 per hour for non-business trip (FAA, 2004) in Year 2000 dollars. An inflation rate of 3% is used. Table 4-8 shows the discounted value of time for each year up to year 2022.

Table 4-8: Discounted Value of Time for Each Year ($ / hr).

Year	Business Trip VOT	Non-Business Trip VOT
2008	57.00	39.90
2009	58.71	41.10
2010	60.48	42.33
2011	62.29	43.60
2012	64.16	44.91
2013	66.08	46.26
2014	68.07	47.65
2015	70.11	49.08
2016	72.21	50.55
2017	74.38	52.06
2018	76.61	53.63
2019	78.91	55.24
2020	81.28	56.89
2021	83.71	58.60
2022	86.22	60.36

Table 4-9: Total Travel Time Savings using SATS (Business Trip, in Million Hours).

Year	ILS Case	LLM Case	Top 1,200 Case	Full Case
2008	10.8	12.8	13.0	14.3
2009	11.1	13.2	13.4	14.7
2010	11.5	13.6	13.8	15.2
2011	11.9	14.0	14.3	15.7
2012	12.2	14.5	14.8	16.2
2013	12.6	15.0	15.3	16.7
2014	13.0	15.4	15.7	17.3
2015	13.4	15.9	16.2	17.8
2016	13.9	16.4	16.8	18.4
2017	14.3	17.0	17.4	19.0
2018	14.8	17.6	17.9	19.7
2019	15.3	18.1	18.5	20.3
2020	15.8	18.7	19.1	21.0
2021	16.4	19.4	19.8	21.8
2022	17.0	20.1	20.6	22.6

Table 4-10: Total Travel Time Savings using SATS (Non-Business Trip, in Million Hours).

Year	ILS Case	LLM Case	Top 1,200 Case	Full Case
2008	0.2	0.4	0.4	0.4
2009	0.2	0.4	0.4	0.4
2010	0.3	0.4	0.4	0.4
2011	0.3	0.4	0.4	0.5
2012	0.3	0.4	0.4	0.5
2013	0.3	0.4	0.4	0.5
2014	0.3	0.4	0.4	0.5
2015	0.3	0.5	0.4	0.5
2016	0.3	0.5	0.4	0.5
2017	0.3	0.5	0.5	0.5
2018	0.3	0.5	0.5	0.5
2019	0.3	0.5	0.5	0.6
2020	0.3	0.5	0.5	0.6
2021	0.3	0.5	0.5	0.6
2022	0.3	0.5	0.5	0.6

Table 4-11: Total Travel Time Savings Benefits (in Million Dollars).

Year	ILS Case	LLM Case	Top 1,200 Case	Full Case
2008	624	746	757	831
2009	660	791	803	880
2010	708	839	852	936
2011	754	890	908	1,000
2012	796	948	968	1,062
2013	847	1,010	1,030	1,127
2014	899	1,067	1,088	1,201
2015	954	1,139	1,155	1,272
2016	1,019	1,210	1,233	1,354
2017	1,079	1,290	1,320	1,439
2018	1,150	1,375	1,398	1,536
2019	1,224	1,456	1,487	1,635
2020	1,301	1,548	1,581	1,741
2021	1,390	1,653	1,687	1,860
2022	1,484	1,763	1,806	1,985

Taxiway Cost

It is assumed the cost of taxiway is around $300 per foot in 2006 dollars. It can be converted to $318.27 per foot in 2008 dollars.

1. ILS case: It is assumed that ILS airports have required parallel taxiway system
2. LLM airport set case: (1,848 – 676) * 25% * 318.27 * 5,000 = $466,265,550
3. Top 1,200 GA airport case: (1,200 – 676) * 25% * 318.27 * 5,000 = $208,466,850
4. Full airport set case: (2,286 – 676) * 25% * 318.27 * 5,000 = $640,518,375

Approach Lighting Cost

The Nebraska Department of Aeronautics study suggests the equipment costs for MIRL, MALSR, and PAPI are $80,000, $200,000 and $30,000 in Year 1999 dollars. So the cost of MIRL, MALSR, and PAPI are $104,382, $260,955 and $39,143.

1. ILS case: It is assumed that ILS airports have required lighting system
2. LLM airport set case: (1,848 – 676) * 104,382 + (1,848 – 676) * 260,955 + (1,848 – 676) * 50% * 39,143 = $451,112,762
3. Top 1,200 GA airport case: (1,200 – 676) * 104,382 + (1,200 – 676) * 260,955 + (1,200 – 676) * 50% * 39,143 = $201,692,054
4. Full airport set case: (2,286 – 676) * 104,382 + (2,286 – 676) * 260,955 + (2,286 – 676) * 50% * 39,143 = $619,702,685

Approach Procedure Mapping and Publication

The Aircraft Owners and Pilots Association estimates average cost of a WAAS approach per runway to be $50,000 (Year 2006). It can be converted to $53,045 in 2008 dollars.

1. ILS case: It is assumed that ILS airports have required final approach fix
2. LLM airport set case: (1,848 – 676) * 53,045 = $62,168,740
3. Top 1,200 GA airport case: (1,200 – 676) * 53,045 = $27,795,580
4. Full airport set case: (2,286 – 676) * 53,045 = $85,402,450

RPZ/OFZ/RSA

It is assumed each RPZ/OFZ/RSA project cost $5,000,000 on average (Year 2006). It is converted to $5,304,500 in 2008 dollars.

1. ILS case: no additional improvement needed.
2. LLM airport set case: (1,848 – 676) * 5,304,500 = $6,216,874,000
3. Top 1,200 GA airport case: (1,200 – 676) * 5,304,500 = $2,779,558,000
4. Full airport set case: (2,286 – 676) * 5,304,500 = $8,540,245,000

Marking

1. ILS case: no additional marking needed
2. LLM airport set case: no additional marking needed
3. Top 1,200 airport case: no additional marking needed
4. Full airport case: (2,286 – 1,842) * 20,000 = $8,880,000

Airborne equipment cost

Assuming the GNS-530 receiver costs $17,500 per unit in 2008, the airborne cost is calculated as follows:

1. ILS case: No GNS-530 is required
2. LLM case: 2,259 * 17,500 = $39,532,500
3. Top 1,200 GA airport case: 2,259 * 17,500 = $39,532,500
4. Full case: 2,556 * 17,500 = $44,730,000

Annual cost

Annual cost includes maintenance cost and Jeppensen approach chart cost.

1. ILS case: No maintenance or subscription cost needed
2. LLM case: (1,848 – 676) * 25% * 3,872 + 2,259 * 500 = $2,263,996
3. Top 1,200 GA airport case: (1,200 – 676) * 25% * 3,872 + 2,259 * 500 = $1,636,732
4. Full airport set case: (2,286 – 676) * 25% * 3,872 + 2,556 * 500 = $2,836,480

Resurfacing costs are ignored because they are relatively small compared with other costs.

This section discusses the cost and benefit estimates for the LLM capability. Infrastructure costs including parallel taxiway, approach lighting system, RSA / RPZ / OFZ, pavement makings, and maintenance costs. The airborne parts focuses on the certified GPS WAAS receiver.

The benefit of the LLM capability considers travel time savings for both business and non-business trip purpose.

The cost and benefit estimates are based on the current airport infrastructure status. Investments are incurred when the analyzed airports do not have the required infrastructures.

The detailed cost-benefit results will be presented in the Chapter 7.

Chapter 5 : External Cost Analysis

A Preliminary Assessment of Airport Noise and Emission Impacts Induced by Small Aircraft Transportation System Operations

Yue Xu, Hojong Baik and Antonio Trani

Virginia Polytechnic Institute and State University, Blacksburg, VA, 24060

Presented at the 6th AIAA Aviation Technology, Integration and Operations Conference (ATIO), AIAA-2006-7736, Wichita, Kansas, Sep. 25-27, 2006.

[Abstract] This paper evaluates potential noise and emission impacts associated with an advanced Small Aircraft Transportation System (SATS). Specifically, the analysis presented in this paper quantifies possible noise and emission contributions of advanced single-engine and multi-engine piston-powered aircraft and very light jet-powered aircraft. The noise impact analysis is carried out using the standard Federal Aviation Administration (FAA) Integrated Noise Model (INM). The emission influence is modeled using the FAA Emission and Dispersion Modeling System (EDMS). The noise signature and emission parameters of a new generation Very Light Jet (VLJ) are modeled in our analysis. Major emission pollutant level is estimated at 2,286 airports. Noise contours studies are conducted at five airport noise impact spanning both metropolitan and rural General Aviation (GA) airports.

Sensitivity analysis is conducted to evaluate influence of the fleet composition and advanced approach procedures in the present and future years.

Introduction

The Small Aircraft Transportation System (SATS) is a concept proposed by the National Aeronautics and Space Administration (NASA) to employ advanced small aircraft (propeller and jet-powered) to satisfy point-to-point, on-demand air transportation services using existing underutilized airports. The SATS program represents a joint effort by government, industry and academia to improve the intercity mobility of various communities in the country. Part of the SATS program goals is to develop aircraft technologies and four operational technical capabilities to make this a reality. From the beginning of the program, SATS proponents identified noise impacts as critical to the acceptance to the concept. To understand potential noise impacts at airports, the Virginia Tech Air Transportation Systems Lab developed a Transportation System Analysis Model (TSAM) to assess impacts of SATS in the National Airspace System (1). TSAM uses county-level socio-economic data to forecast the number of intercity trips in the United States. The model uses proven transportation engineering methods to predict the number of travelers selecting among various modes of transportation (i.e., auto, airline, and other technologies like Very Light Jets and piston-powered aircraft operating as air-taxis). The demand for on-demand air transportation services has been evaluated at 2,286 SATS technology enabled airports. The demand function at each airport is characterized in terms of daily person-trips and daily flight arrivals and departures.

This paper presents the evaluation of noise and emission impacts performed typical SATS enabled airports using the TSAM model. The noise impacts of SATS operations are assessed at five representative airports using the standard Federal Aviation Administration (FAA) Integrated Noise Model (INM) version 6.1c. The emission influences are modeled at 2,286 SATS compatible airports using the standard FAA Emission and Dispersion Modeling System (EDMS).

Three representative SATS aircraft are modeled in our study: 1) a new generation Very Light Jet (VLJ) aircraft, 2) an advanced technology Single-Engine (SE), piston-powered aircraft, and 3) an advanced technology Multi-Engine (ME), piston-powered aircraft. The VLJ aircraft is modeled as a new vehicle with advanced low-thrust, medium by-pass ratio turbofan engines in INM. VLJ aircraft have relatively slow approach and takeoff speeds (belonging to approach speed group A) and the Sound Exposure Level (SEL) curves have been adjusted to account for lower thrust produced by VLJ engines. The emission matrices of the VLJ including Carbon Monoxide (CO), Total Hydrocarbon (THC), Non-Methane Hydrocarbons (NMHC), Nitrogen Oxides (NOx) and Sulfur Oxides (Sox) are modeled using regression analysis. The advanced Single-Engine and Multi-Engine aircraft are substituted by aircraft with analogous features.

SATS noise impacts are evaluated at five general aviation airports: Manassas, Virginia (HEF), Blacksburg, Virginia (BCB), Danville, Virginia (DAN), Teterboro, New York (TEB) and Goodland Municipal Airport, Kansas (GLD). These airports were selected among 2,286 airports modeled in TSAM because they represent a good cross-section of airport operations, aircraft mix, runway configurations and proximity to population centers. The INM required population and street files are obtained from the U.S. Census Bureau Census 2000 (2) and TIGER 2000 (3) database. The topographical features around these airports are integrated using 3-second elevation data. Baseline scenarios without SATS activities are executed based on average daily traffic operations and using the based aircraft fleet mix reported by Airnav (4). Day and night operations of different aircraft groups are estimated using the General Aviation and Air Taxi Activity (GAATA) report (5). The GAATA categorizes general aviation aircraft into five major groups and this categorization is applied in our analysis: Single-Engine Piston-Powered, Twin-Engine Piston-Powered, Single-Engine Turbo-Propeller, Twin-Engine Turbo-Propeller and Jet. VFR and IFR flight paths are constructed using the U.S. Terminal Procedural charts (6) and pilot anecdotal information. Runway utilization estimation is gathered from various sources including wind data, tower observations, pilot anecdotal information and the FAA Aviation Systems Performance Metrics (ASPM) (7) reports. Local operation information is used wherever available.

SATS emission impacts are assessed at the full spectrum of 2,286 airports that satisfy typical operational requirements of SATS aircraft. A current small aircraft, the Cessna Citation Sovereign is modified to simulate emission rates of the VLJ. Fuel flows of the VLJ are estimated by back engineering analysis of the VLJ drag polar. Regression analysis reveals a linear relationship between thrust and NOx emission rate. Two general assumptions are made (Table 5-1). The total taxi and queuing time is a function of the number of operations per runway per day. The number of operations of each five major aircraft groups applies a weighted average of the number of based aircraft and operation statistics reported by the GAATA.

Table 5-1: General Assumptions.

Assumptions for Total Taxi and Queuing Time and Local Traffic Assignment		
Number of Operations per Runway per Day (N)	Total Taxi and Queuing Time (minutes)	Percentage of Reported Local Traffic Assumed to be LTO Cycles / TGO Operations
N >200	30	80% / 20%
200> N >100	20	50% / 50%
N < 100	10	20% / 80%

GAATA Aircraft Categories and Operation Statistics (Year 2000)	
Aircraft Type	Reported Number of Annual Landings (in millions)
Piston – 1 Engine	27.59
Piston – 2 and More Engines	2.52
Turboprop – 1 Engine	1.37
Turboprop – 2 and More Engines	2.63
Jet Engines	2.18

LTO: Landing and Takeoff
TGO: Touch and Go
GAATA: General Aviation and Air Taxi Activity

Future scenarios with SATS operations are executed based on the SATS demand estimated by the Virginia Tech TSAM Model. A typical SATS aircraft, the Very Light Jet (VLJ) is developed as a new aircraft and engine model in both INM 6.1c and EDMS 4.2 to represent emerging very light jet technologies such as the Eclipse Aviation 500 and the Cessna Mustang. Representative aircraft selected to model single-engine and multi-engine SATS aircraft are shown in Table 5-2. All SATS operations are assumed to be itinerary.

Table 5-2: Representative Aircraft.

Baseline Scenario		
Aircraft Category	Representative Aircraft for Noise Analysis	Representative Aircraft for Emission Analysis
Single-Engine Piston-powered	1985 single-engine FP Prop	Cessna 172 Skyhawk
Multi-Engine Piston-powered	Beech Baron 58P	Aztec
Single-Engine Turboprop	Beech T34 Mentor	400A Hustler
Multi-Engine Turboprop	Cessna Citation II	Cessna 441 Conquest II
Jet	Mitsubishi Mu-300 Diamond	Mitsubishi Mu-300 Diamond
SATS Scenario		
Single-Engine small aircraft	Cessna 206H Stationair	Socata Tampico
Multi-Engine small aircraft	Cessna Conquest II	Cessna Conquest II
		Beech King Air 350
Very Light Jet	Very Light Jet	Very Light Jet

Literature Review

Since the late 1950s, air transportation noise has generated controversy from many communities around airports. This concern led to the passage of legislation by Congress and thus regulations by the aviation administration. In the early years, noise at airports was surveyed by continuously monitoring sound exposure levels at places of interests. An important survey revealed that when the sound levels exceed 65 decibels, people report a noticeable increase in annoyance (8). This survey assigned additional weight to sounds at night and this measurement became the standard aviation noise measurement: Day/Night Average Sound Level (DNL).

The increased number of noise studies around airports prompted the development of the Integrated Noise Model (INM) by the Federal Aviation Administration (FAA) in 1978. This model is the standard tool accepted by the federal government to conduct

Federal Aviation Regulation (FAR) Part 150 noise compatibility planning and FAA Order 1050 environmental analysis (9). It is an average value model to quantify annual noise influences using the concept of an ‘average annual day’. The average annual day comprises various typical long-term average conditions. The DNL is one of the 16 noise metrics supported by INM. The aircraft profile and noise calculation algorithms are based on three documents (10): the Society of Automotive Engineers (SAE) Aerospace Information Report (AIR) SAE-AIR-1845 (11), SAE-AIR-1751 (12) and SAE Aerospace Recommended Practice (ARP) SAE-ARP-866A (13). Noise impacts are reported by contour areas, population affected and noise level at points of interests.

In addition to the single-airport analysis tool INM, Metron Aviation developed the Noise Integrated Routing System (NIRS) under contract to the FAA to address large-scale aviation noise modeling over multi-state regions (14). The first version of NIRS was introduced in 1998 and new capacities are integrated continuously. The NIRS model allows users to specify tracks in three dimensions or follow standard tracks. Traffic elements that cause principle noise impacts can be identified. It also provides comparisons of noise impacts across alternative airspace routing designs. Noise analysis results of NIRS are presented by noise comparison maps and tables.

The INM models fixed-wing aircraft operation noise impacts and helicopter and rotorcraft are modeled by the Heliport Noise Model (HNM) and the Rotorcraft Noise Model (RNM). The HNM is based on the INM but it is able to model more complicated helicopter flight activities. The RNM, developed by NASA, is capable of developing approach and departure noise abatement procedures to promote civilian use of rotorcraft (15).

As a military counterpart of the INM, Noisemap has been used to model sound exposure in the vicinity of military air bases. Another military aircraft noise analysis model called the Military Operating Area and Range Noise Model calculates noise from subsonic military aircraft over Military Training Routes (MTRs), Military Operating Areas (MOAs) and Special Use Airspaces (such as ranges).

One common feature of all models is the evaluation of noise exposure due to multiple aircraft activities. Single aircraft flyover noise levels can be evaluated using Menu 10 or an updated version Sound Exposure Level Calculator (SELCal). There are papers on enhancements to INM and INM error analysis in the literature. This paper will not address these issues.

Emission represents another primary environmental concern introduced by air transportation besides noise. Research on aviation emission effect is driven by increasing air transportation activities and environment protection awareness since the 1960s.

The most widely applied terminal area emission analysis model in U.S. is the Emission and Dispersion Modeling System (EDMS) developed by FAA. In 1998, FAA revised its policy on air quality modeling procedures to identify EDMS as the required model to perform air quality analyses for aviation sources instead of a preferred model. It utilizes Aircraft Engine Emission Databank provided by ICAO to estimate emissions during five phases, namely idle, takeoff, climb out, approach and touch-and-go operations. The model integrates several Environment Protection Agency's (EPA) model to estimate aircraft, Ground Support Equipment (GSE) and Auxiliary Power Unit (APU) emissions. Pollutants such as Total Hydrocarbons (THC), Non-Methane Hydrocarbons (NMHC), Volatile Organic Compounds (VOC), Oxides of Nitrogen (NOx), and PM2.5 are calculated. EPA's state-of-art dispersion model AERMOD, along with its supporting modules, is integrated for dispersion analysis. As detailed airport information is required for dispersion analysis, air quality evaluation will concentrate on emission only. The EDMS does not estimate emission beyond mixing height. Model's default mixing height is 3,000 ft and user can modify this value according to local condition.

For altitude beyond 3,000 ft, a method developed by Boeing Company called Boeing Method (BM2) has been widely applied (Baughcum, 1996). Different from the EDMS, BM2 emission estimation is based on power levels instead of set mode points. It is essentially a curve fitting method that plot emission indices and fuel flow on a logarithmic basis and make a series of linear fits between pair of mode points.

Similarly, EUROCONTROL developed a Toolset for Emission Analysis (TEA) (Eurocontrol Experiment Center, 2003) including an emission module Advanced Emission Model (AEMIII), a contrail formation prediction tool CONTRAIL, and a meteorological database. The AEMIII applies the same emission estimation philosophy in the vicinity of the airport as the EDMS and extends the analysis to en-route. En-route emission analysis is based on aircraft fuel burn calculated using the Base of Aircraft Data

(BADA). Emission rate and fuel flow from the ICAO databank is adapted to the atmospheric condition using the BM2. The model output agrees well with the historical data (Carlier, 2004).

Modeling

The SATS population considered in the analysis includes proposed very light jets weighing less than 3,181 kg (7,000 lb). The new generation of VLJ aircraft is best represented by the Eclipse 500, recently achieved FAA type certification. The analysis also considers representative turboprop aircraft. Examples of existing vehicles in this category are the Raytheon B300, Pilatus PC-12, and Cessna Caravan. New generation of high-performance single engine piston-powered aircraft (Cirrus SR-22 and Lancair 400) are also considered in the analysis. In the INM and EDMS, single engine and multi-engine turboprop are modeled using similar aircraft (i.e., substitution method) because the powerplants and flight characteristics of new generation SE piston-powered aircraft are similar to modern aircraft in the same category today.

Noise and Emission Characteristics of the Very Light Jet

The VLJ modeled in the analysis models a pressurized aircraft with six people (including pilots), cruises at 676 km/hr (365 knots) and has a range of 2,037 km (1,100 nautical miles) with four occupants using National Business Aviation Association (NBAA) IFR reserves . The VLJ is equipped with two medium by-pass ratio turbofans producing 1100-lbf takeoff thrust at sea level static conditions.

The noise profile of the very light jets is created in the INM as they represent new generation of small jets powered by substantially lower thrust engines than their current corporate jet counterparts. The most similar corporate jet model in the INM aircraft database is the Cessna Citation II. This aircraft is selected as the baseline vehicle and then substantially modified to represent the expected acoustic and performance characteristics of the VLJ. Table 5-3 shows the general characteristics of the new VLJ aircraft created in the INM aircraft database. The noise profile characteristics are presented in Figure 5-1. Compared to first generation corporate jets, very light jets are expected to have relatively small noise footprints. Advances in engine noise reduction technology and the low thrust of the new generation turbofan engines are likely to make these aircraft very quiet.

Another advantage of the VLJ is their relatively modest approach and takeoff speeds. The 3,000 kg VLJ prototype in this analysis has an approach speed of 166 km/hr (90 knots) at maximum landing weight.

To model VLJ emission characteristics, thirteen current corporate jet engine emission rate tables are extracted to establish basic statistical model between four emission indices (HC, CO, NOx, SOx). A liner relationship can be established between the NOx emission rate and the fuel flow. In the absence of published VLJ CO and HC emission rates, two assumptions are evaluated: same rate as current light jet corporate jets and a reduced (80%) rate. Cessna Citation Sovereign is used as a prototype to model VLJ. SOx emission rate is assumed to be consistent with current light corporate jets. The estimated emission parameters of the VLJ can be found in Table 5-4.

Table 5-3: General Parameters of the Modeled Very Light Jet Employed to Derive Noise Signature.

Category	Parameters	Value
Weight Summary	Maximum Takeoff Weight	2,579 kg (5,640 lbs)
	Maximum Landing Weight	2,431 kg (5,360 lbs)
	Maximum Landing Distance	762 meters (2500 ft.)
Engine	Number of Engines	2
	Static Thrust	1100 lb
Maximum Climb Jet Coefficients	E	900 lb
	F	-1.99614 (lb/kt)
	Ga	6.15000e-02 (lb/ft)
	Gb	-2.40502e-6 (lb/ft^2)
	H	0
Maximum Takeoff Jet Coefficients	E	900 lb
	F	-2.21793 (lb/kt)
	Ga	6.83330e-2 (lb/ft)
	Gb	-2.67224 (lb/ft^2)
	H	0

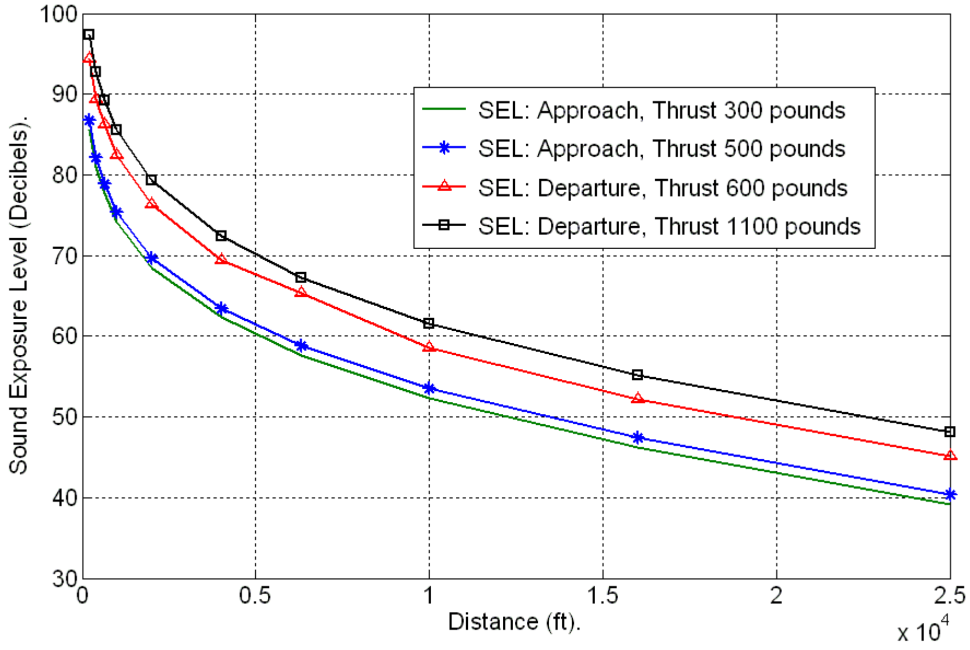

Figure 5-1: Sound Exposure Level (SEL) Curves of Very Light Jet Modeled.

Table 5-4: Estimated Emission Parameters of the VLJ.

Phase	CO (g/Kg)	CO (80%) (g/Kg)	HC (g/Kg)	HC (80%) (g/Kg)	NOx (g/Kg)	SOx (g/Kg)	Time in Mode (mins)	Fuel Flow (Kg/s)
Takeoff	0.81	0.648	0.09	0.072	6.77	0.54	1.08	0.140
Climb out	0.97	0.776	0.10	0.080	10.93	0.54	0.97	0.120
Approach	5.23	4.184	0.14	0.112	2.20	0.54	4.52	0.030
Idle	42.3	33.84	5.94	4.752	1.60	0.54	0.22	0.012

SATS Noise and Emission Impact Analysis

Using the predicted noise footprints and emission indices of the very light jets and using aircraft substitution methods for SATS single and multi-engine, piston-powered aircraft, we derived airport noise contour maps and emission levels using the SATS demand function predicted by TSAM.

Throughout the analysis, we attempted a consistent level of detail across all five airports. Baseline operations without SATS are obtained from Airnav website, FAA records and tower records when available. SATS operations are estimated by the TSAM model and added to the baseline operations to study the potential environmental impact. Sensitivity analyses are conducted with different fleet composition and glide path angles for the year 2014. Two SATS fleets are tested in the analysis: 1) 100% of the SATS aircraft comprising very light jets and, 2) 30% single-engine, 30% multi-engine and 40% very light jet, comprising the SATS scenario. SATS scenarios are executed for 3, 4 and 5 degrees of Glide Path Angle (GPA) to evaluate the influences of advanced approach technology to mitigate noise and emission impacts.

Noise Case Studies

Five case studies are conducted at airports located at both metropolitan area and rural communities. The selected airports are representative of the spectrum of airports with future SATS enabled technologies. In our analysis, we identify specific operating patterns at those airports. When the local information is unavailable, assumptions are generally made based on pilot anecdotal experience. It is important to point out that general aviation traffic at non-towered airports is more difficult to characterize than commercial traffic at towered airports. Many general aviation airports lack accurate records on the number of both local and itinerant general aviation operations. At many of these small airports the Terminal Area Forecasts (TAF) data is generally unreliable.

Our analysis starts with several general assumptions made regarding the percent of day and night operations. The GAATA report states that 85% of the general aviation flights in the U.S. occur during daytime conditions. Similarly, 15% of the general aviation flights occur at night. The ratio of Visual Flight (VFR) and Instrument Flight Rules (IFR) is typically 85/15 in the NAS. Flight paths are constructed using U.S. Terminal Procedure Charts to model IFR arrivals and departures. VFR arrival and departure tracks are modeled using standard airport flight patterns. Three types of aircraft compose the SATS fleet: Single-Engine, Multi-Engine aircraft and Very Light Jet. The base scenario considers 37.5% of approach, 37.5% of departures and 25% of touch-and-go operations except for jets. Jets only operate 50% approach and 50% departures. All SATS operations are assumed to be itinerant operations with equal number of arrivals and departures. The general characteristics of the five airports studied are shown in Table 2. A brief description of each airport follows.

Teterboro Airport (TEB), Teterboro, New Jersey. Teterboro airport represents one of the busiest General Aviation (GA) airports in the country with substantial corporate jet operations. TEB is projected to attract a substantial number of SATS operations due to its proximity to the New York Metro Area. A total of 81 business jets are based at TEB today (16) demonstrating its high attractiveness to corporate aviation. To model the existing operations at TEB, an aircraft substitution scheme is developed based on the INM recommendations for various aircraft present in the INM database. Runway utilizations are obtained from the FAA Aviation System Performance Metrics (ASPM).

*Manassas Regional Airport / Harry P.Davis Field (HEF), Manassas, Virgini*a. The Manassas airport represents another busy GA airport. The airport has a fleet comprised of mostly single-engine aircraft (304) with 57 multi-engine aircraft and 10 business jets. Using the TSAM model, we forecast moderate SATS/VLJ traffic due to the proximity to the Northern Virginia and Washington DC areas. HEF was selected because its aircraft mix is different than that of TEB. Yet its location helps us understand SATS noise issues in metropolitan areas.

The GAATA landing statistics are used as the primary source to estimate operations of each aircraft group. Based on wind analysis, operations on each runway end are assigned as 80% on runway 16L/34R (1,737 meters) and 20% on 16R/34L (1,128 meters).

Danville Regional Airport (DAN), Danville, Virginia. Danville represents rural areas where SATS operations could improve the accessibility to air transportation and thus promote economic development. The airport has a fleet population very similar to the national average with 3 jet-powered, 5 multi-engine and 40 single-engine piston aircraft. The GAATA landing statistics and verbal communications with the airport control tower personnel are the main sources to estimate operations of each aircraft group. Runway utilization information is provided by observations from the airport tower personnel. A standard left-turn VFR is modeled corresponding to the flight practices at the airport.

Virginia Tech / Montgomery Executive Airport (BCB), Blacksburg, Virginia. Blacksburg represents another example of a rural airport with potential use of SATS on-demand services due to the proximity to a large State University and its Corporate Research Center. BCB is located around hilly terrain and represents a candidate to benefit from lower landing minima capability developed by the SATS Program (Highway-in-the-Sky Instrumentation, synthetic vision systems, and optimal energy steep approaches coupled with Wide Area Augmentation System navigation). Wind data, field observations and pilot anecdotal information provide detailed information about aircraft models, flight paths, number of operations on each path and runway utilization (17). Each flight path has been verified by local pilots flying at the airport. Eleven types of aircraft are modeled at the airport including a gyro-copter and a Robinson 22 helicopter. The gyro-copter and helicopter are created and added to the INM database using user-defined profiles and noise curves.

Renner Field / Goodland Municipal Airport (GLD), Goodland, Kansas. Goodland Municipal airport represents a third rural airport with potentially very low SATS demand. Based on the TSAM model airport demand estimation, the vast majority of 2,286 airports modeled are projected to have low to very low SATS demands. The GAATA landing statistics are used as the primary source of information to estimate operations at this airport. A detailed wind rose analysis using the FAA Airport Design software program AD42 is conducted for each runway end to derive runway utilization. The wind data is obtained from National Virtual Data System (NVDA) Local Data Publication (18) which provides weather information at 260 airports in the U.S.

Emission Impacts Analysis

SATS emission impacts are analyzed at 2,286 SATS enabled airports. Compared with noise analysis, the emission analysis attempts to apply national statistics instead of local information. The estimate uses FAA landing facility database to derive airport baseline operations. The same aircraft categorization is followed and one aircraft in

EDMS database is chosen to represent the group. Aircraft emission indices in the EDMS are obtained from ICAO Aircraft Engine Emission Databank. Aircraft emission is the product of annual operations and emission rate. Emissions of Group Support Vehicles and Auxiliary Power Units are estimated by the integrated MOBILE 6.

Results and Conclusions

In general, the analysis shows that SATS aircraft (including very light jets) could be good to community neighbors. Compared to existing twin-engine jet aircraft, SATS aircraft are expected to have smaller noise and emission footprints. Table 5-5 and Table 5-6 list the INM and EDMS model results with the forecasted operations in the year 2014. Figure 5-2 through Figure 5-6 show 65DNL contour maps at all five airports studied with a nominal three-degree GPA approach. Sensitivity analysis shows a mixed fleet generates slightly higher noise levels around airports. The changes to GPA seem to have very little impact on the noise exposure around the airports.

Table 5-5: INM Noise Analysis Impacts (65 DNL Contours, Year 2014).

Airport ID	GPA (Degrees)	Baseline Scenario		SATS Scenario VLJ Only		SATS Scenario Mixed SATS Fleet	
		Population Influenced	Area (Acres)	Population Influenced	Area (Acres)	Population Influenced	Area (Acres)
TEB	3	5,050	1,517.7	5,746	1,624.9	6,027	1,692.6
	4	5,050	1,517.7	5,627	1,617.4	6,027	1,687.1
	5	5,050	1,517.7	5,627	1,612.5	5,998	1,683.7
HEF	3	0	188.4	0	273.7	0	303.7
	4	0	188.4	0	269.4	0	300.4
	5	0	188.4	0	266.0	0	297.8
DAN	3	0	67.9	0	82.1	0	81.9
	4	0	67.9	0	81.8	0	81.8
	5	0	67.9	0	81.7	0	81.7
BCB	3	0	84.7	0	89.3	0	93.7
	4	0	84.7	0	89.2	0	93.7
	5	0	84.7	0	89.1	0	93.6
GLD	3	0	115.0	0	116.0	0	116.0
	4	0	115.0	0	116.0	0	116.0
	5	0	115.0	0	116.0	0	116.0

Table 5-6: EDMS Emission Impacts.

Compared to the Baseline (No SATS)	Takeoff	Climb-out	Approach	Idle
Maximum Increase (%)	2886.299	11151.81	230949.9	361055.5
Average Increase (%)	0.98	3.80	78.80	123.19

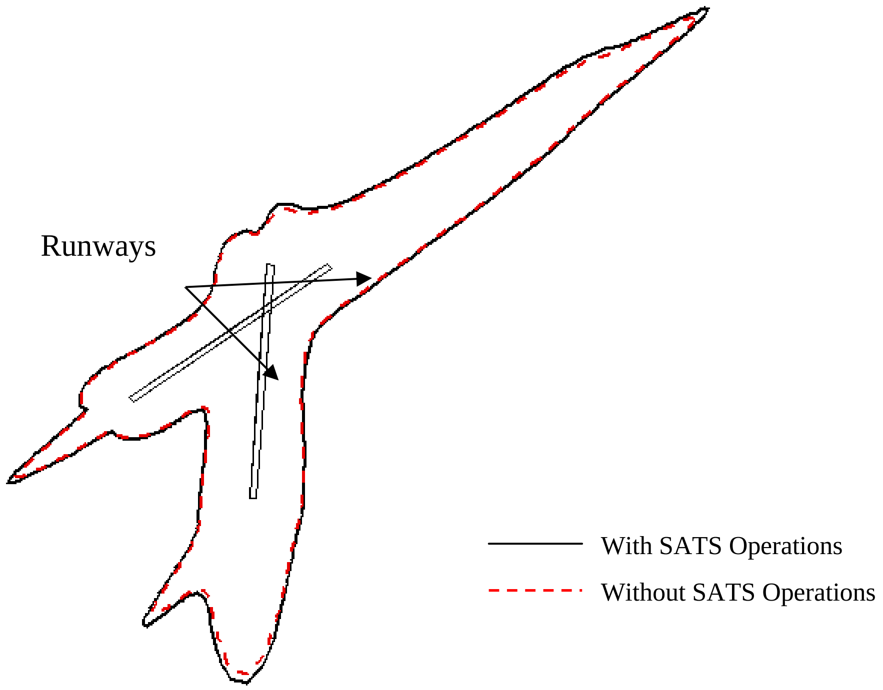

Figure 5-2: Teterboro Airport Noise Contour Maps (65DNL).

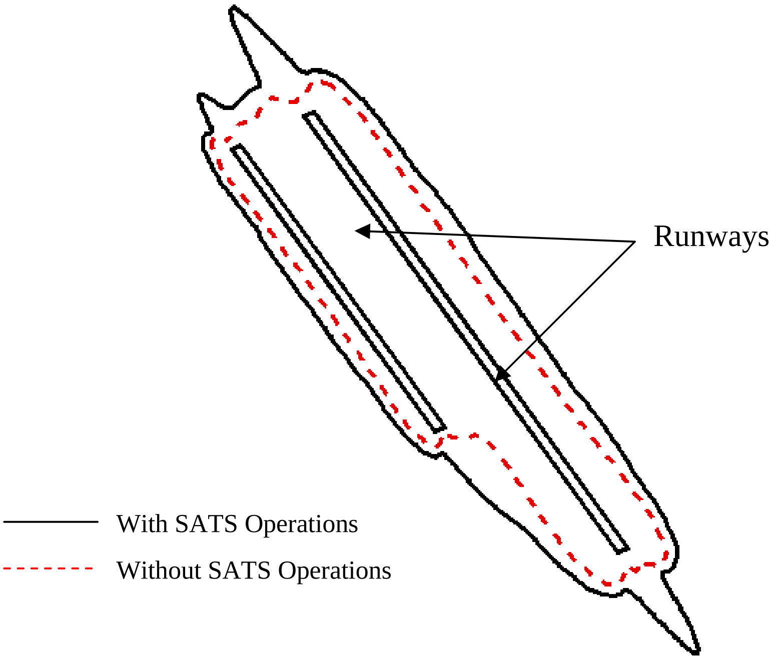

Figure 5-3: Manassas Airport Noise Contour Maps (65DNL).

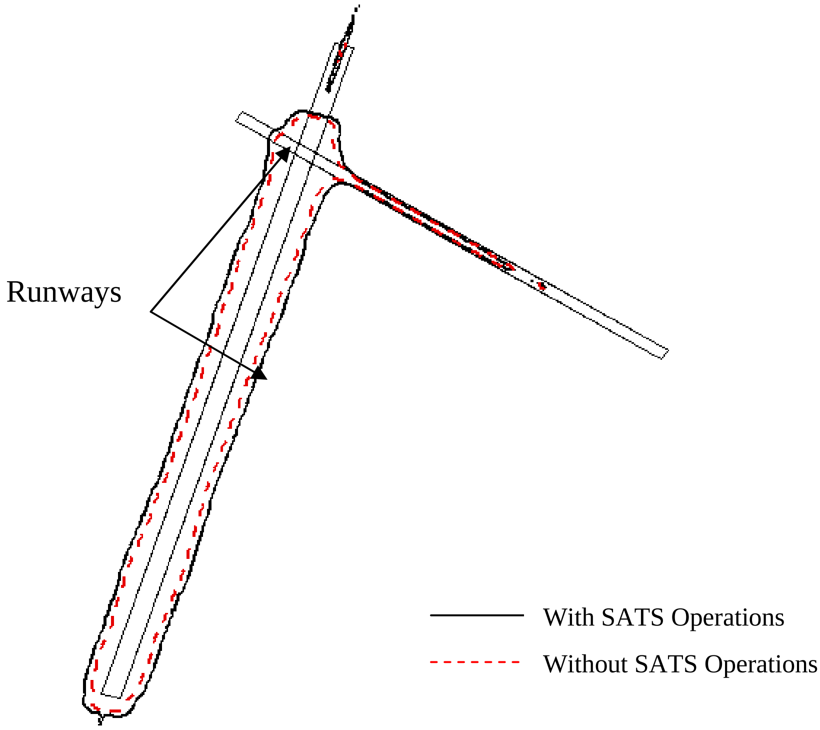

Figure 5-4: Danville Airport Noise Contour Maps (65DNL).

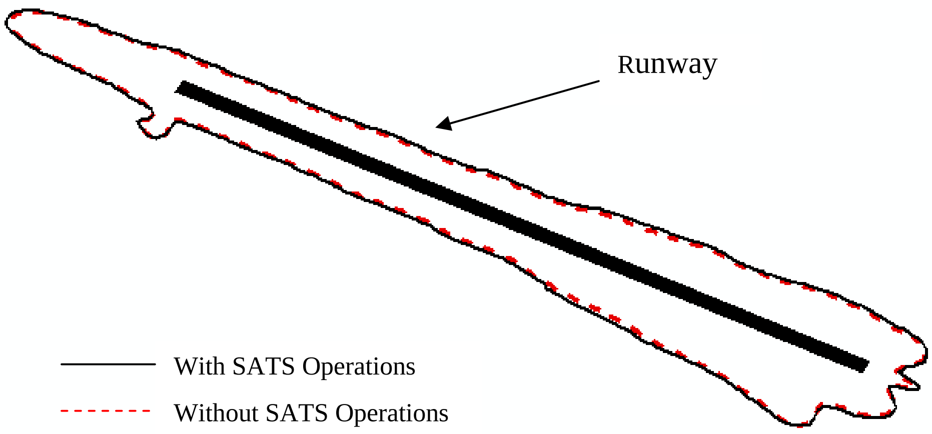

Figure 5-5: Blacksburg Airport Noise Contour Maps (65DNL).

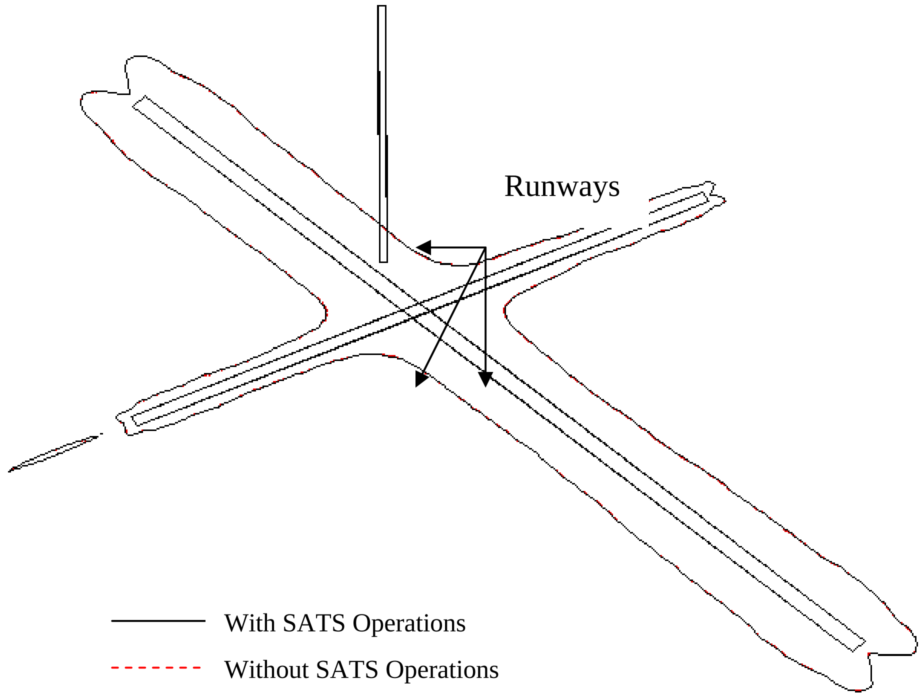

Figure 5-6: Goodland Airport Noise Contour Maps (65DNL).

In the analysis presented for the year 2014, a three percent annual growth in the GA operations has been assumed. Population density and distribution around the airports for the year 2014 are assumed to be the same as the Census 2000 survey. SATS demand is estimated using the TSAM model. The analysis shows that the additional SATS operations increase the 65DNL noise contours from 7% at TEB to less than 0.9% at a rural airport like GLD. The noise impact at a terrain-challenged airport like BCB is estimated to be 5%. At TEB an estimated five percent more populations would be affected after the introduction of SATS on-demand operations. It is important to put in context these numbers. If TEB enacts a ban to stage 2 corporate jets today, an estimated 11% reduction in the size of the 65DNL is possible. Estimate of major pollutant productions using the EDMS indicate a slight increase in CO, THC and NMHC and a significant increase in NOx and SOx. This result suggests careful environmental evaluation during deployment of the SATS system.

Recommendations

This paper presents a first assessment of the potential environmental impacts of SATS operations at various airports in the U.S. The following recommendations could be pursued to improve the analysis presented in the paper.

- This study represents a "pessimistic" scenario in that all SATS operations are assumed to be additional operations above the current GA activities at each airport. This is the result of a new paradigm on-demand services possible with the introduction of VLJ and advanced SE and ME piston aircraft. SATS aircraft including the VLJs will probably have a replacement effect on existing aircraft technologies used today. Thus some of the older and noisier aircraft operating at these airports could probably be replaced by more environmentally friendly SATS aircraft. The authors are currently studying this replacement effect.

- The information on reliable local and itinerant operations at some rural airports is scarce. This is an issue that needs to be improved across NAS. The FAA Terminal Area Forecast (TAF) is unreliable and thus local information is preferred for airport impact studies. General survey and statistical data sources are preferred for nation-wide impact analyses. The analysis conducted in our study mixes local and general airport operational data as a best effort to model credible operations at each airport. Future efforts could use other data sources. For example, real-time or historical radar track data will be helpful in building flight paths and improve the estimation of runway utilization.

- The noise and emission signature of very light jets and new technologies should be continuously validated along with the flight test progress of the new SATS aircraft. At least two new generation VLJs will be certified in 2006 (the Eclipse 500 and the Cessna Mustang). Future noise impact studies will benefit from actual aircraft noise certification data.

- A more comprehensive nation-wide noise analysis is desired to address, at the national level, the impacts of SATS aircraft deployment. A method with virtual airports and virtual runways is being developed at Virginia Tech to address this concern. The method considers typical runway configurations and parametric SATS demands as two explanatory variables. Local adjustments will be made to account for specific local effects at all 2,286 airports or at other sets of airports studied.

- Commercial operation impacts should be included in future studies.

Acknowledgments

This study has been supported by the National Consortium of Aviation Mobility (NCAM) under a grant with the Virginia SATS Alliance. Thanks are due to Stuart Cooke and Jeff Viken (NASA Langley Research Center) for their constructive criticism and technical oversight. We would like to thank Senanu Ashiabor, Howard Swingle and Chad Ackley for their valuable input with the Blacksburg airport data. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. government or an airport authority.

References

1: Trani, A.A., H. Baik, H. Swingle, S. Ashiabor, N. Hinze, A. Seshadri, K. Murthy and Y. Xu, *SATS Transportation System Analysis Model*, Blacksburg, Virginia, May 2005.

2: U.S. Census Bureau, Census 2000 population file, http://www2.census.gov/census_2000/datasets/redistricting_file--pl_94-171/

3: U.S. Census Bureau, Census TIGER 2000/Line file, http://www.esri.com/data/download/census2000_tigerline/

4: Airnav website. http://www.airnav.com

5: Federal Aviation Administration, *General Aviation and Air Taxi Activity Survey*, Washington DC. 2002.

6: Federal Aviation Administration National Aeronautical Charting Office, *U.S. Government Flight Information Publication*, U.S. Terminal Procedure, 2003.

7: Federal Aviation Administration, Aviation System Performance Metrics (ASPM), http://www.apo.data.faa.gov/faamatsall.HTM

8: T.J. Schultz, Synthesis of Social Surveys on Noise Annoyance, *Journal of the Acoustical Society of America* 64(2) pp377-405, 1978.

9: Federal Aviation Administration Office of Environment and Energy, ATAC Corporation, Volpe National Transportation Systems Center Acoustics Facility, *The Integrated Noise Model 6.0 User's Guide*, September 1999.

10: Federal Aviation Administration Office of Environment and Energy, ATAC Corporation, Volpe National Transportation Systems Center Acoustics Facility, *The Integrated Noise Model 6.0 Technical Manual*, January 2002.

11: Society of Automotive Engineers, Committee A-21, Aircraft Noise, *Procedure for the Computation of Airplane Noise in the Vicinity of Airports*, Aerospace Information Report No. 1845, Warrendale, PA: Society of Automotive Engineers Inc., March 1986

Chapter 6 : DataComm Analysis

SIMMOD Simulation

Two basic inputs are required for SIMMOD: airport and airspace information plus demand. The airport ground infrastructure layout is built with the assistance of Google Earth satellite images and FAA airport diagram (Figure 6-1 to Figure 6-4). Airspace structure is derived from Performance Data Analysis and Reporting System (PDARS).

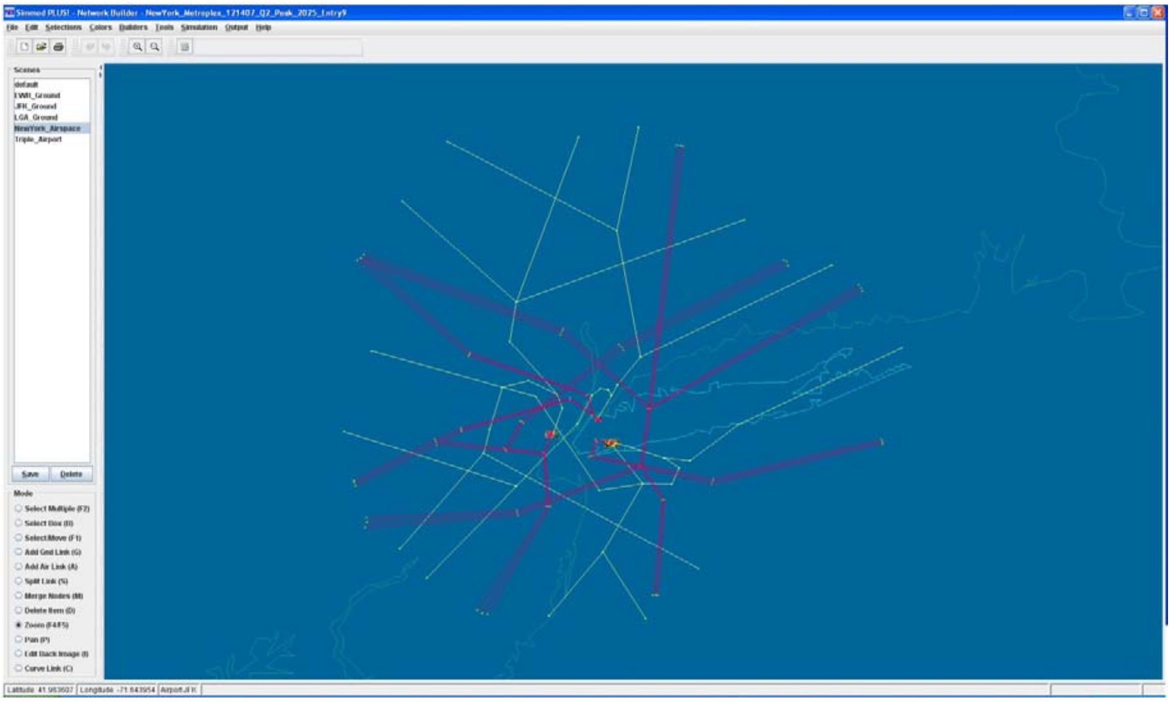

Figure 6-1: New York Area Airspace Structure.

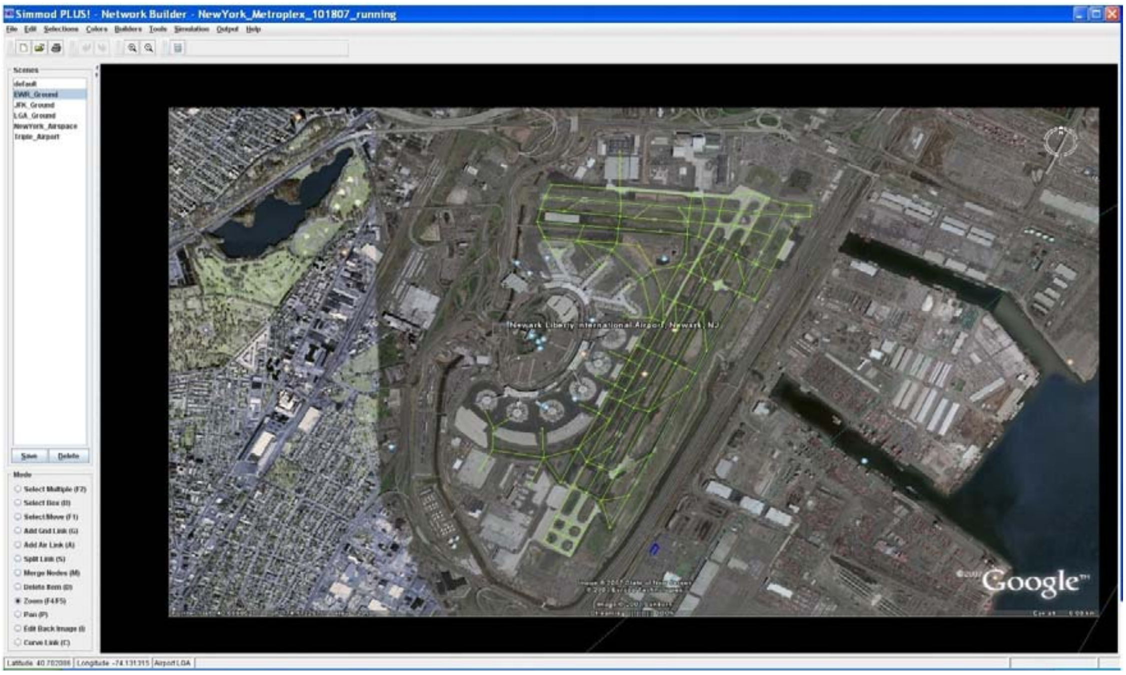

Figure 6-2: EWR (Newark Airport) Ground Infrastructure Layout.

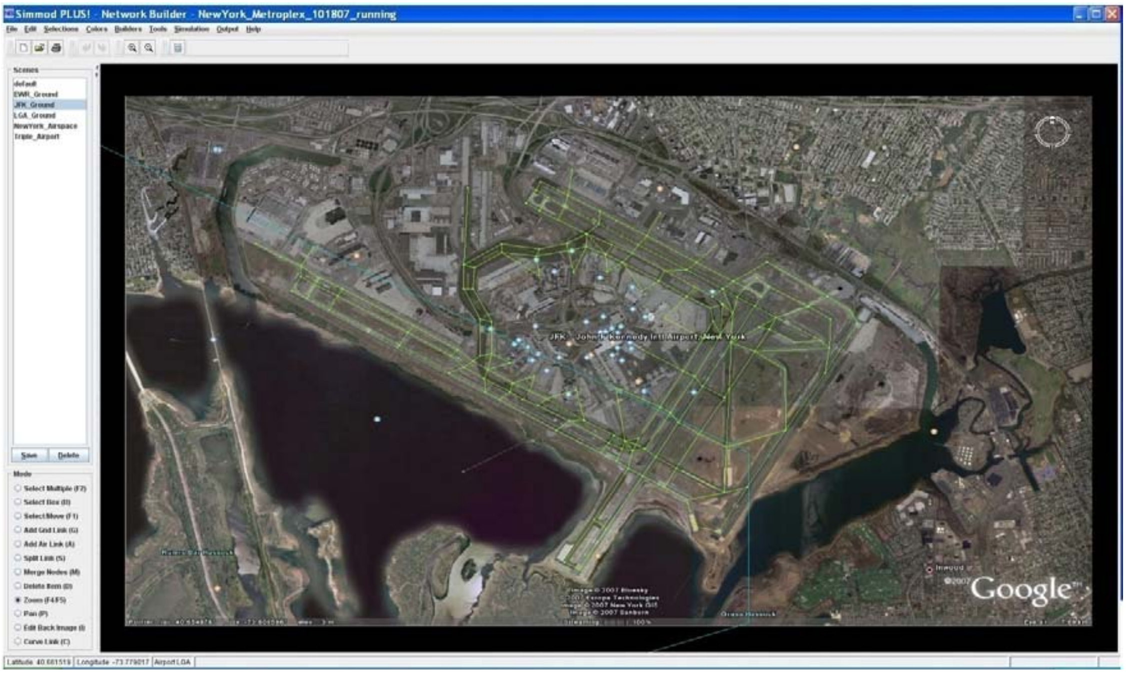

Figure 6-3: JFK (John F. Kennedy Airport) Ground Infrastructure Layout.

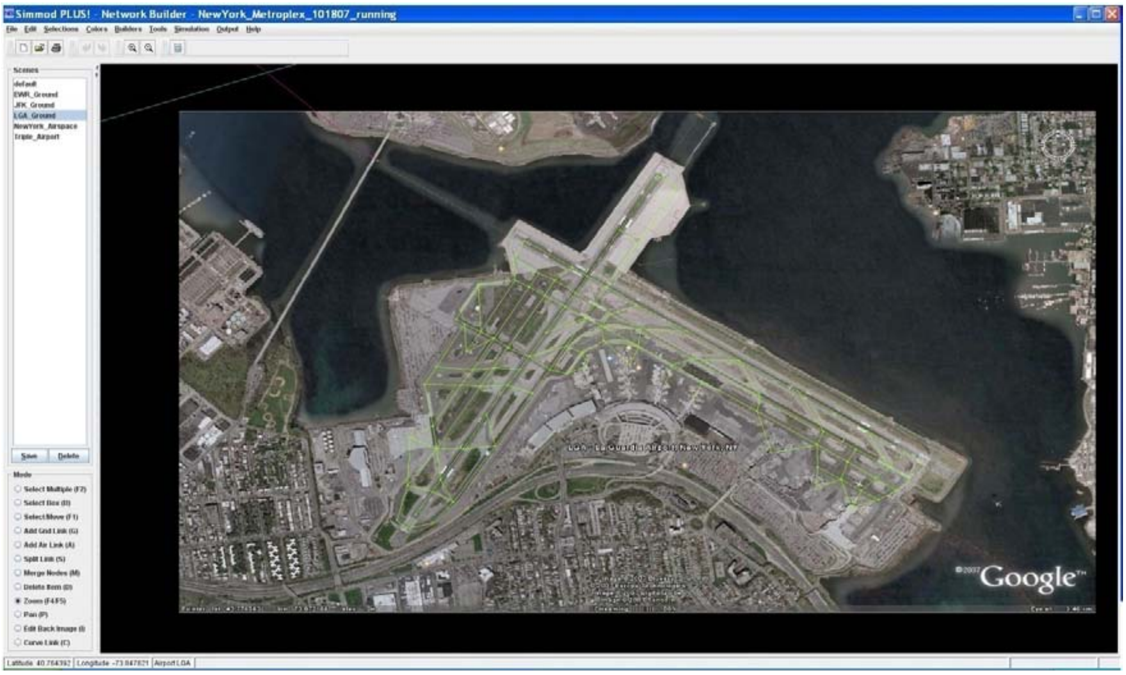

Figure 6-4: LGA (LaGuardia Airport) Ground Infrastructure Layout.

Observations from PDARS data are analyzed in order to derive the airspace structure, aircraft speed and separation matries. In SIMMOD, three sets of separation matries are used to establish the operation pattern around terminal area. Approach-to-approach separations are dominated by wake vortex characteristics. Departure-to-departure separations are primarily determined by the speed of the following aircraft. Arrival-followed-by-departure separations are applied to prevent encroachment. Separations for departure-followed-by-arrival are ignored as arrival aircraft have priority to land.

Defined by their wake vortex characteristics, aircraft are classified into Heavy, Large and Small group by FAA. A minimum separation value is set for each aircraft group and another. In DataComm analysis, observations from PDARS for each airport are analyzed to set up such a separation matrix. An interpolation point is created first and the time stamp that each aircraft crosses this point is recorded. Figure 6-5 illustrates the relative location of the interpolation point to real operations. Once the time sequence is derived, a distribution of time separation can be obtained. This distribution analysis is conducted for each combination of the aircraft group to establish minimum separation.

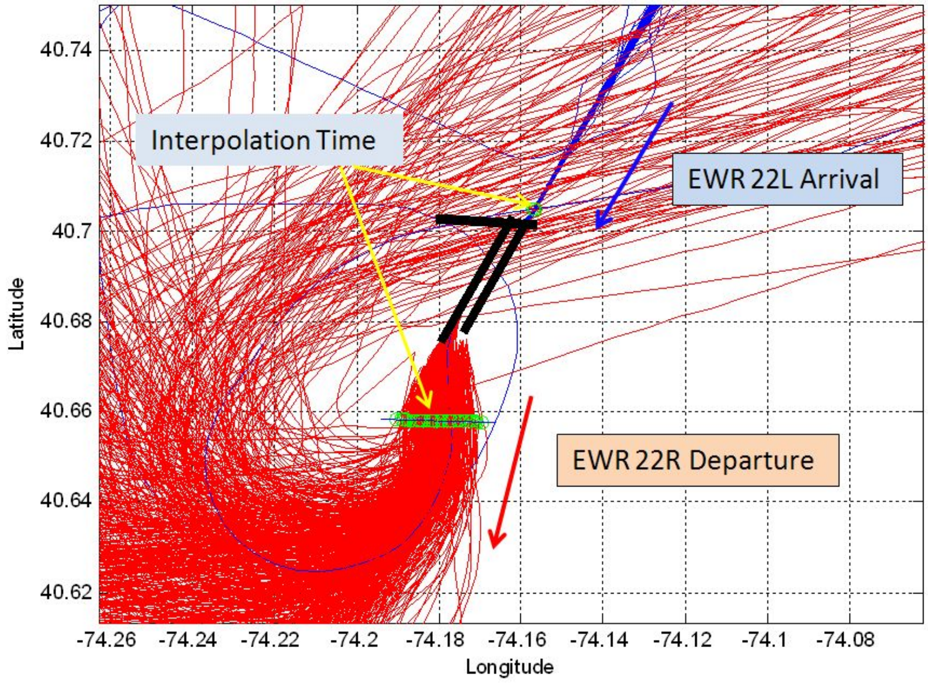

Figure 6-5: EWR Separation Matrix Estimation Using PDARS.

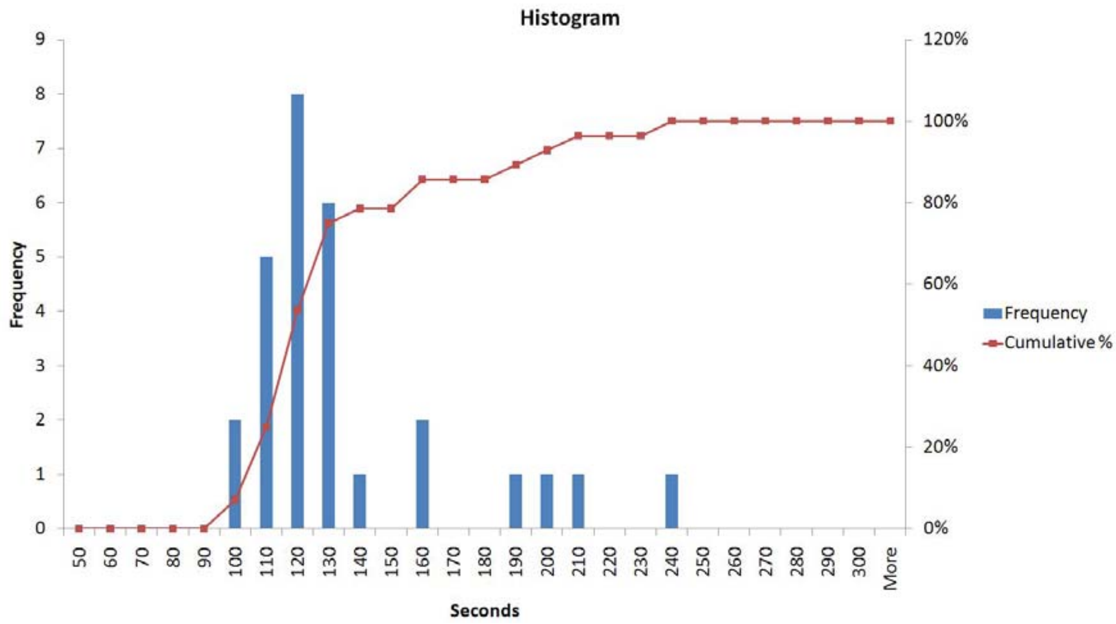

Figure 6-6: EWR Approach-Approach Separation (Heavy-Heavy).

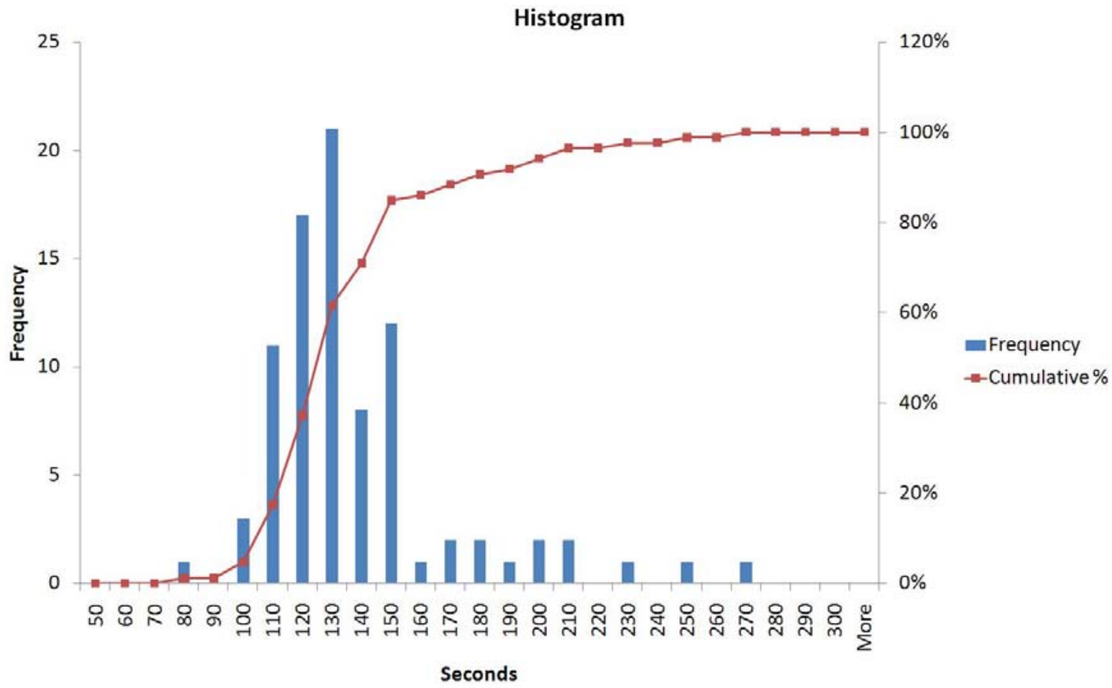

Figure 6-7: EWR Approach-Approach Separation (Heavy-Large).

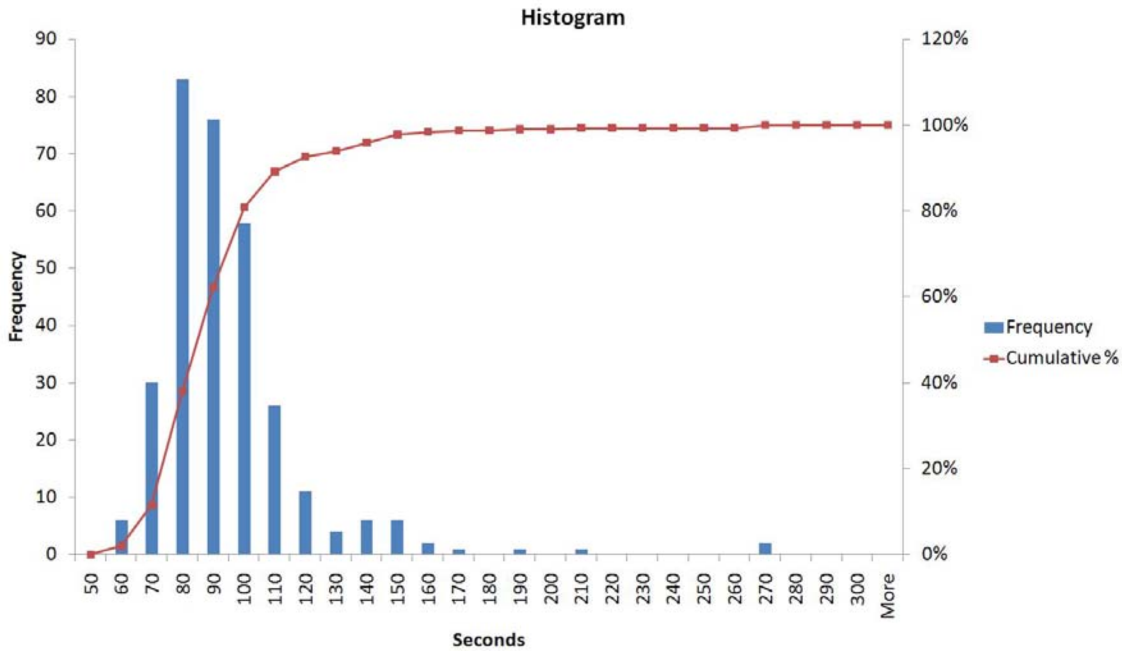

Figure 6-8: EWR Approach-Approach Separation (Large-Large).

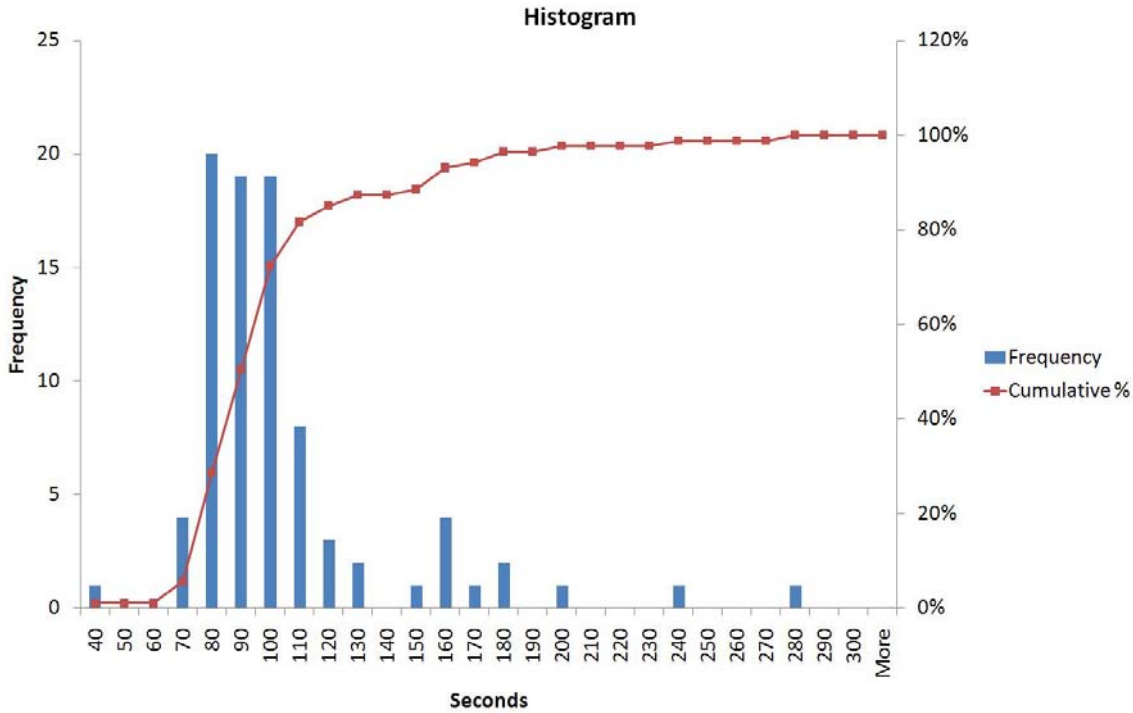

Figure 6-9: EWR Approach-Approach Separation (Large-Heavy).

Figure 6-6 to Figure 6-9 present the distribution of arrival-arrival separation between Large and Heavy group at EWR. The number of operations from Small aircraft is not significant enough to conduct this analysis. Ten percentile of the cumulative separation value, instead of the minimum value, is applied as to compensate abnormities or record error. The same method is applied for departure-departure separation matrix as well. In this way, Arrival-Arrival and Departure-Departure separation matrix for the three airports are derived from the real observations. Figure 6-10 to Figure 6-15 illustrate separation matrix of each airport.

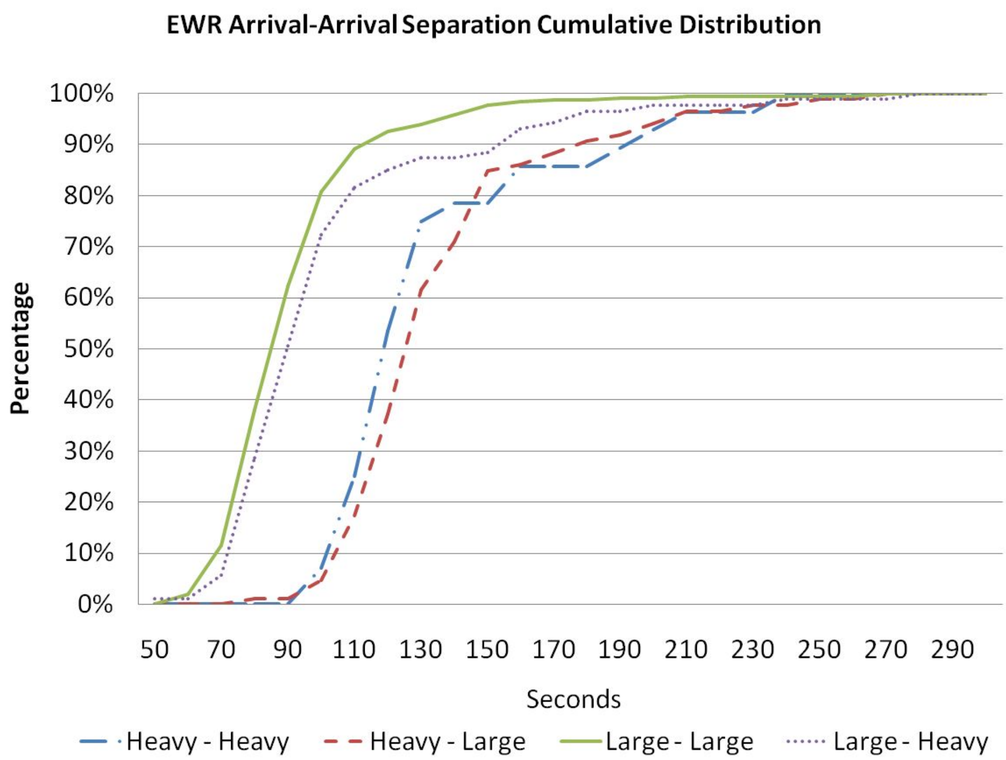

Figure 6-10: EWR Arrival-Arrival Separations.

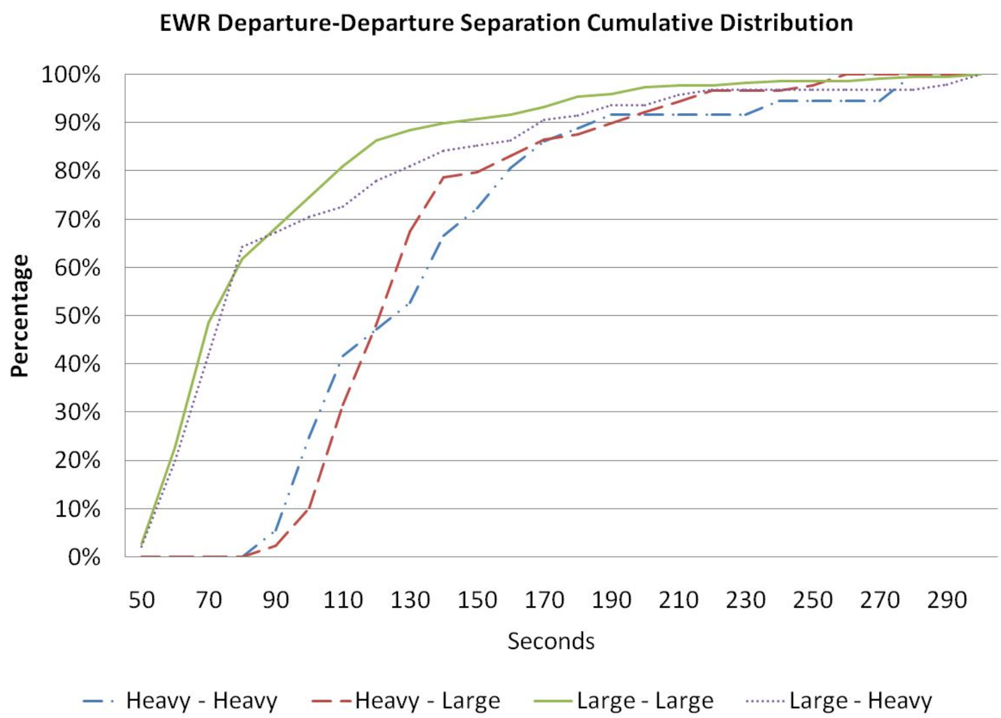

Figure 6-11: EWR Departure-Departure Separations.

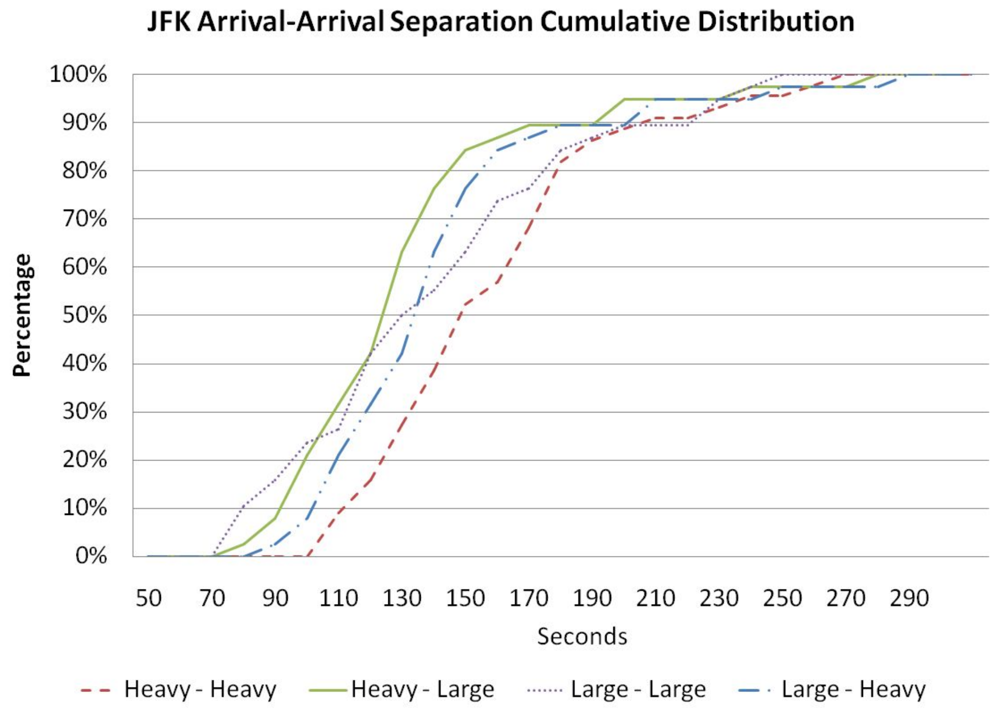

Figure 6-12: JFK Arrival-Arrival Separations.

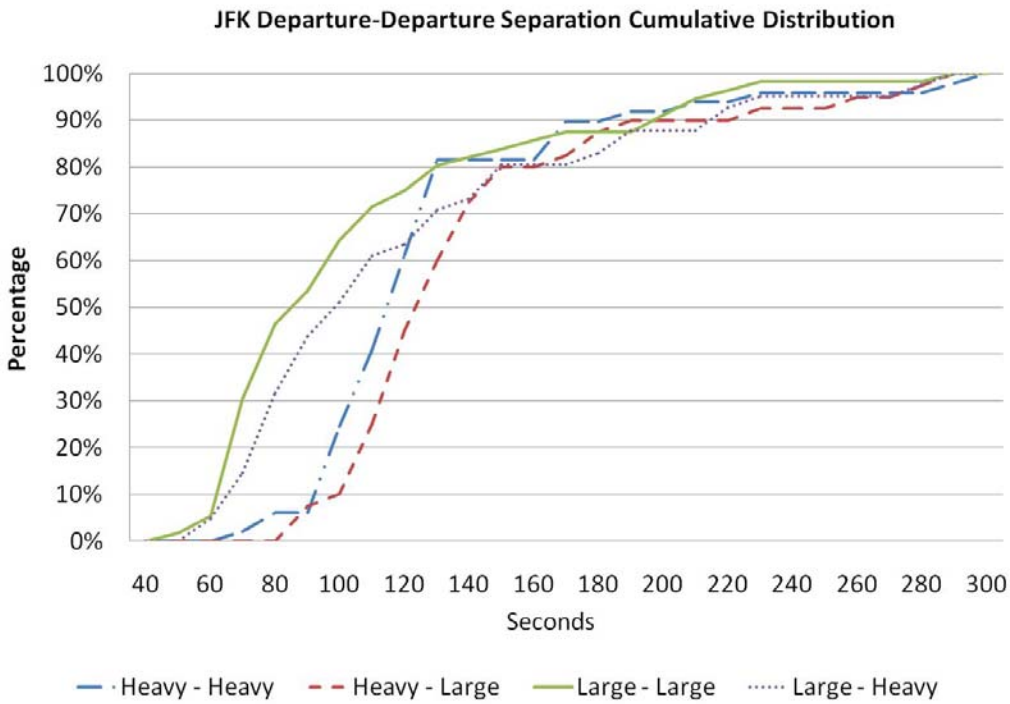

Figure 6-13: JFK Departure-Departure Separations.

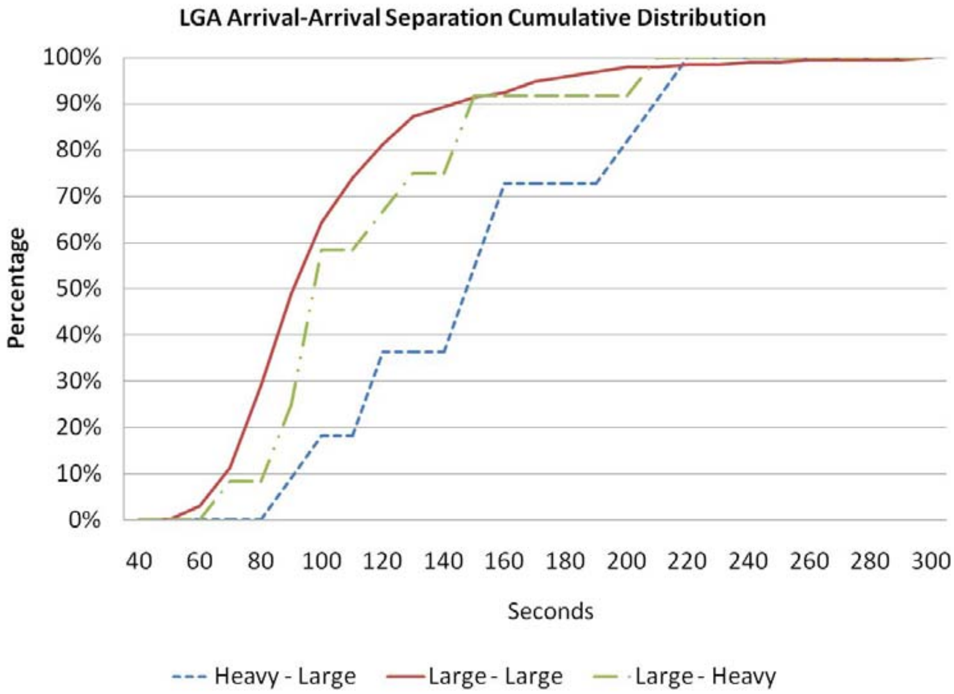

Figure 6-14: LGA Arrival-Arrival Separations.

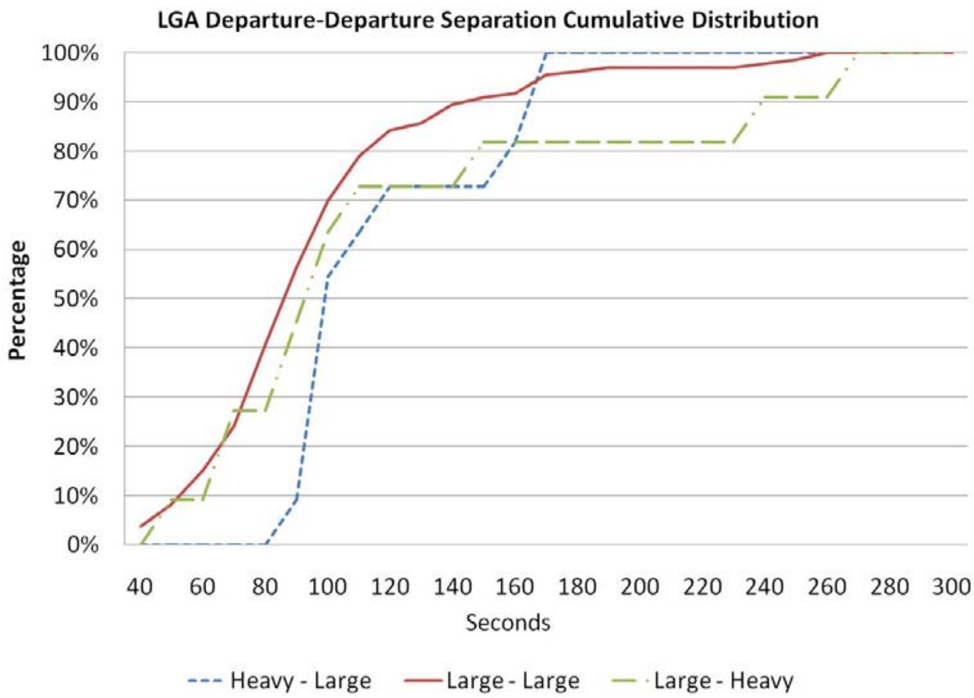

Figure 6-15: LGA Departure-Departure Separations.

LGA airport, different from EWR and JFK's parallel runway system, has two intersecting runways. Therefore, it is necessary to analyze arrival-departure separation. The same interpolation method is applied (Figure 6-16) and results (Figure 6-17) are employed in the simulation.

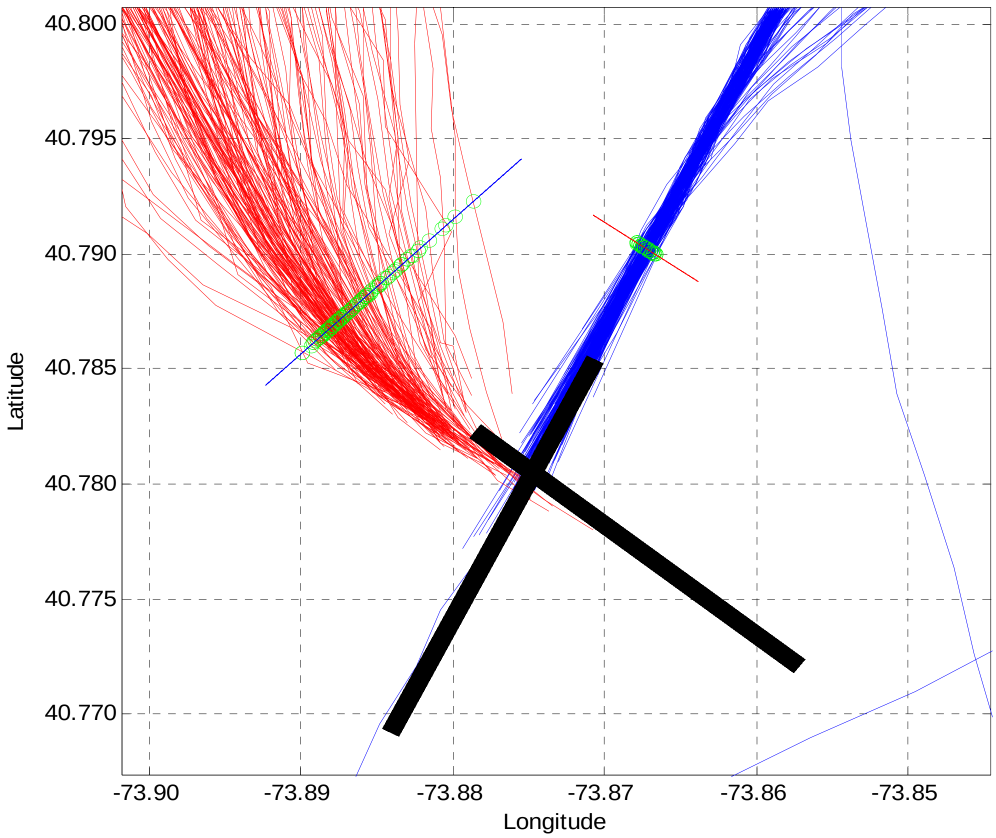

Figure 6-16: LGA Arrival-Departure Separation Matrix Estimation using PDARS.

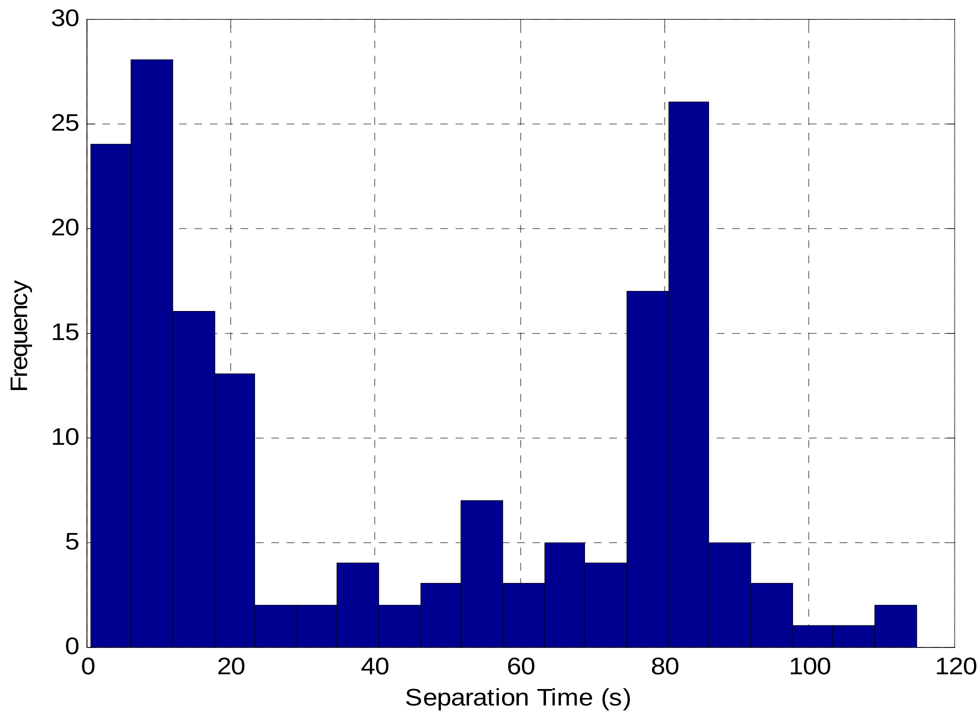

Figure 6-17: LGA Arrival-Departure Separation Matrix PDF Distribution.

In addition to separation standards, PDARS also provides data on node and link performance. For instance, approach speed and flight level from southwest bound and west bound to EWR is shown in Figure 6-18 and Figure 6-19. Based on this information, node location and altitude and link speed are defined in SIMMOD.

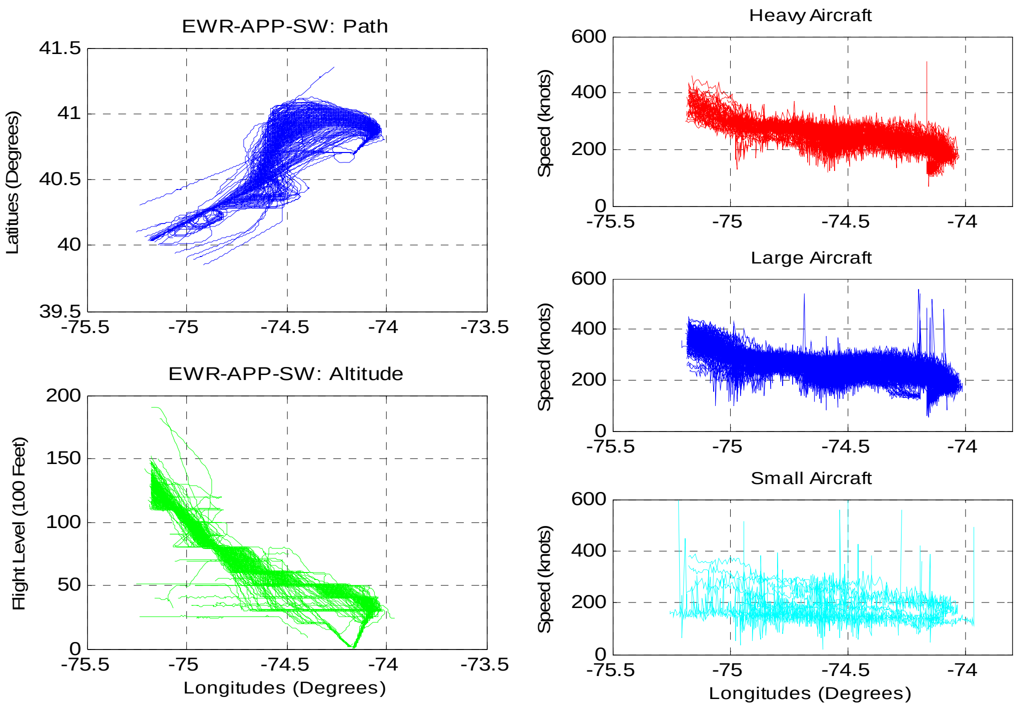

Figure 6-18: EWR Speed and Flight Level Profile – South West Bound.

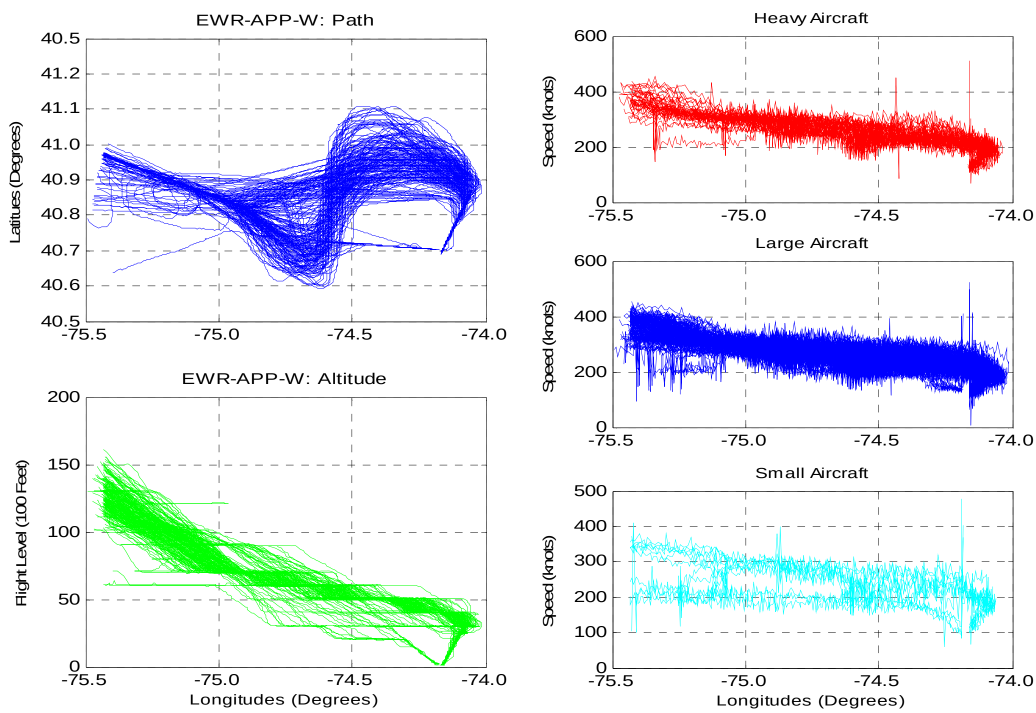

Figure 6-19: EWR Speed and Flight Level Profile – West Bound.

Chapter 7 : Results

LLM Analysis

The cost and benefit components are discussed in detail in Chapter 4. This section will present the results of the LLM cost-benefit analysis.

Cost-Benefit Analysis

Costs are converted to annual cost using the Capital-Recovery Factor (CRF). The CRF factor is calculated as:

$$CRF = \frac{i(1+i)^n}{(1+i)^n - 1} = \frac{0.07 \times (1+0.07)^{15}}{(1+0.07)^{15} - 1} = 0.11 \qquad (7\text{-}1)$$

Table 7-1: Total Annual Cost Estimation.

	ILS Case	LLM Case	Top 1,200 Case	Full Case
Taxiway	0	$466,265,550	$208,466,850	$640,518,375
Approach Lighting	0	$451,112,762	$201,692,054	$619,702,685
Procedure Design	0	$62,168,740	$27,795,580	$85,402,450
RPZ/OFZ/RSA	0	$6,216,874,000	$2,779,558,000	$8,540,245,000
Marking	0	0	0	$8,880,000
Airborne	0	$39,532,500	$39,532,500	$44,730,000
Total (Initial)	0	$7,235,953,552	$3,257,044,984	$9,939,478,510
Converted Annual Cost	0	$795,954,891	$358,274,948	$1,093,342,636
Maintenance and Subscription Cost	0	$2,263,996	$1,636,732	$2,836,480
Total (Annual)	0	$798,218,887	$359,911,680	$1,096,179,116

Total benefits are calculated in Table 4-11. Table 7-1 and Figure 7-2 illustrates the cost-benefit curve for the ILS airport set, the Top 1,200 airport set, the full airport set and the ILS airport set. The first set (ILS – Top 1,200 – Full) indicates that around 1,850 airports are worthwhile updating from a cost-benefit viewpoint whereas the second set (ILS – LLM – Full) suggests 1,750 airports. It is more cost-effective to update the top 1,200 airport set first as it generates benefits faster than the LLM set.

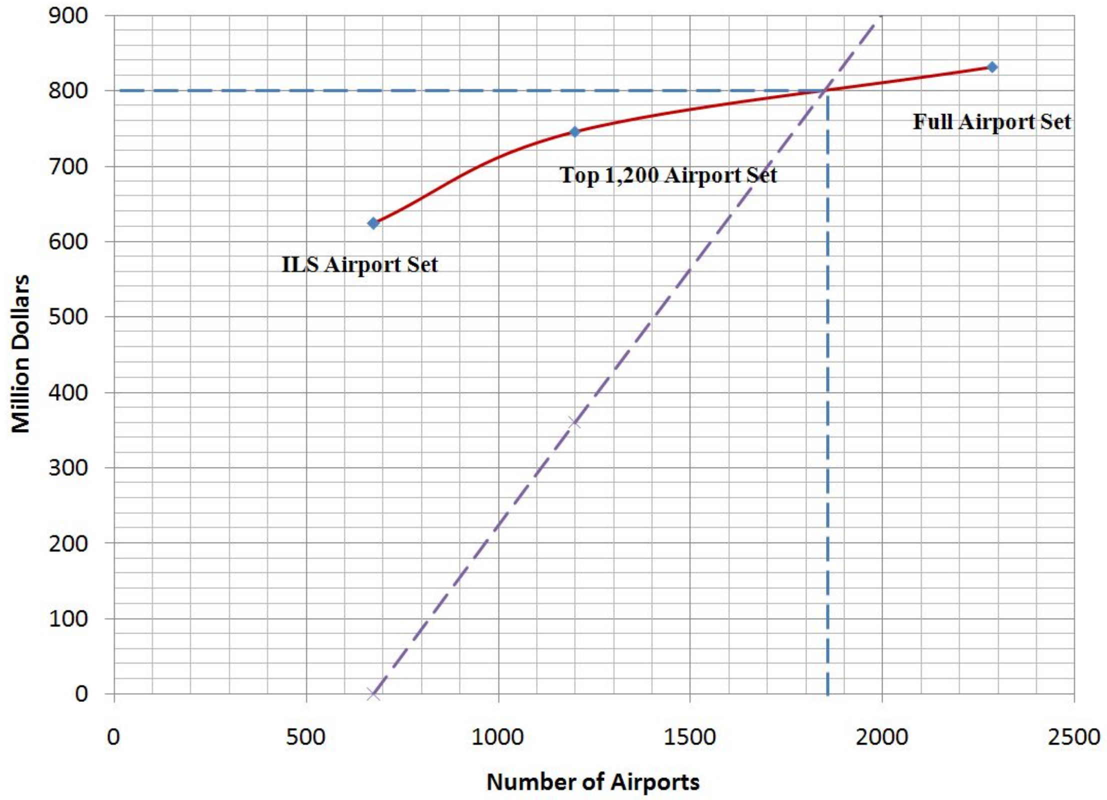

Figure 7-1: Cost-Benefit Curve (ILS – Top 1,200 GA – Full Airport Set Case).

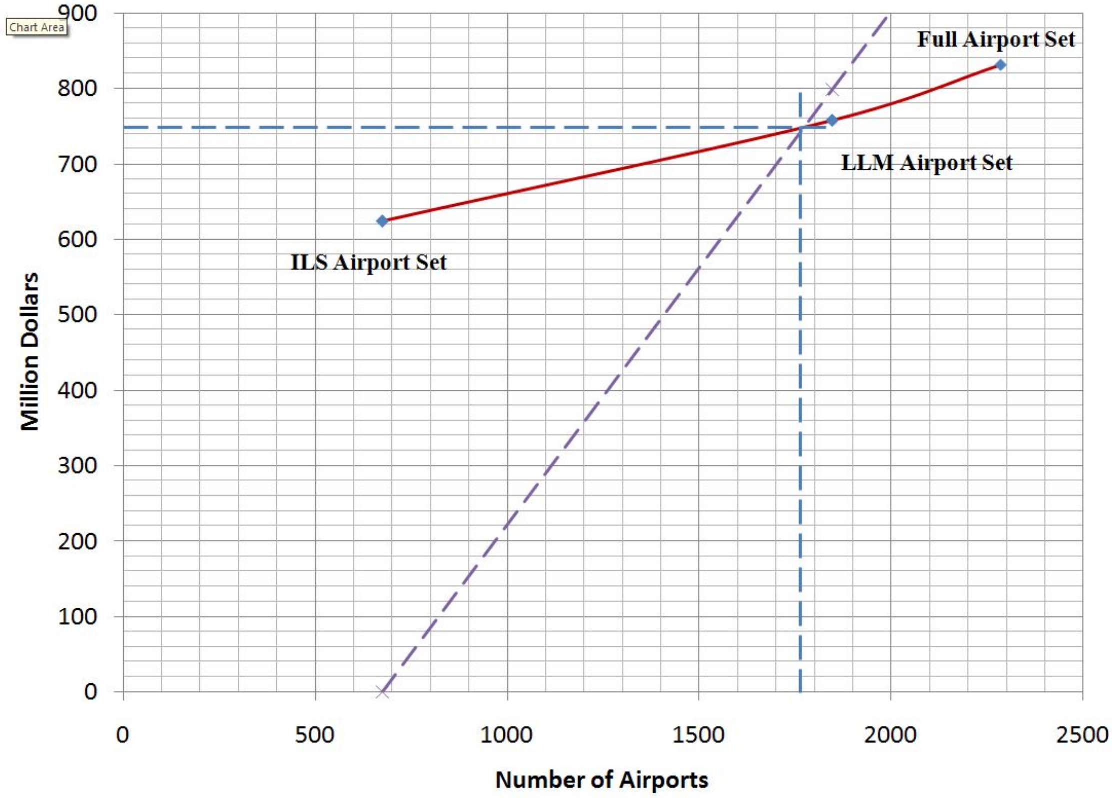

Figure 7-2: Cost-Benefit Curve (ILS – LLM – Full Airport Set Case).

Sensitivity Analysis

Table 7-1 shows that the driven cost factor is the RPZ/OFZ/RSA cost. Therefore a sensitivity analysis is conducted assuming average RPZ/OFZ/RSA cost is 25% higher or 25% lower.

- Assume RPZ/OFZ/RSA costs 25% higher

1. ILS case: no additional improvement needed.
2. LLM airport set case: (1,848 – 676) * 7,155,625 = $8,368,392,500
3. Top 1,200 GA airport case: (1,200 – 676) * 7,155,625 = $3,749,547,500
4. Full airport set case: (2,286 – 676) * 7,155,625 = $11,520,556,250

Table 7-2: Cost Estimates Assuming RPZ/OFZ/RSA Costs 25% Higher.

	ILS Case	LLM Case	Top 1,200 Case	Full Case
Taxiway	0	$502,495,000	$224,665,000	$690,287,500
Approach Lighting	0	$595,405,780	$265,597,260	$816,052,650
Procedure Design	0	$67,091,140	$29,996,380	$92,164,450
RPZ/OFZ/RSA	0	$8,368,392,500	$3,749,547,500	$11,520,556,250
Marking	0	0	0	$8,880,000
Airborne	0	$39,532,500	$39,532,500	$44,730,000
Total (Initial)	0	$9,572,916,920	$4,309,338,640	$13,172,670,850
Converted Annual Cost	0	$1,051,054,821	$473,142,218.69	$1,446,288,452.29
Maintenance and Subscription Cost	0	$2,263,996	$1,636,732	$2,836,480
Total (Annual)	0	$1,053,318,817	$474,778,951	$1,449,124,932

- Assume RPZ/OFZ/RSA costs 25% lower

1. ILS case: no additional improvement needed.
2. LLM airport set case: (1,848 – 676) * 4,293,375 = $5,031,835,500
3. Top 1,200 GA airport case: (1,200 – 676) * 4,293,375 = $2,249,728,500
4. Full airport set case: (2,286 – 676) * 4,293,375 = $6,912,333,750

Table 7-3: Cost Estimates Assuming RPZ/OFZ/RSA Costs 25% Higher.

	ILS Case	LLM Case	Top 1,200 Case	Full Case
Taxiway	0	$502,495,000	$224,665,000	$690,287,500
Approach Lighting	0	$595,405,780	$265,597,260	$816,052,650
Procedure Design	0	$67,091,140	$29,996,380	$92,164,450
RPZ/OFZ/RSA	0	$5,031,835,500	$2,249,728,500	$6,912,333,750
Marking	0	0	0	$8,880,000
Airborne	0	$39,532,500	$39,532,500	$44,730,000
Total (Initial)	0	$6,236,359,920	$2,809,519,640	$8,564,448,350

Converted Annual Cost	0	$684,718,797	$308,470,154.46	$940,330,392.36
Maintenance and Subscription Cost	0	$2,263,996	$1,636,732	$2,836,480
Total (Annual)	0	$686,982,793	$310,106,886	$943,166,872

Figure 7-3 and Figure 7-4 demonstrate the sensitivity of RPZ/OFZ/RSA cost. If the cost of RPZ/OFZ/RSA is 25% higher, the intersecting point of cost and benefit changes from 1,850 airports to 1,600 airports for the ILS – Top 1,200 – Full case. For the ILS – LLM – Full case, the range is from 1,750 to 1,550 airports. If the cost is 25% lower, almost all the SATS compatible airports are worthwhile upgrading.

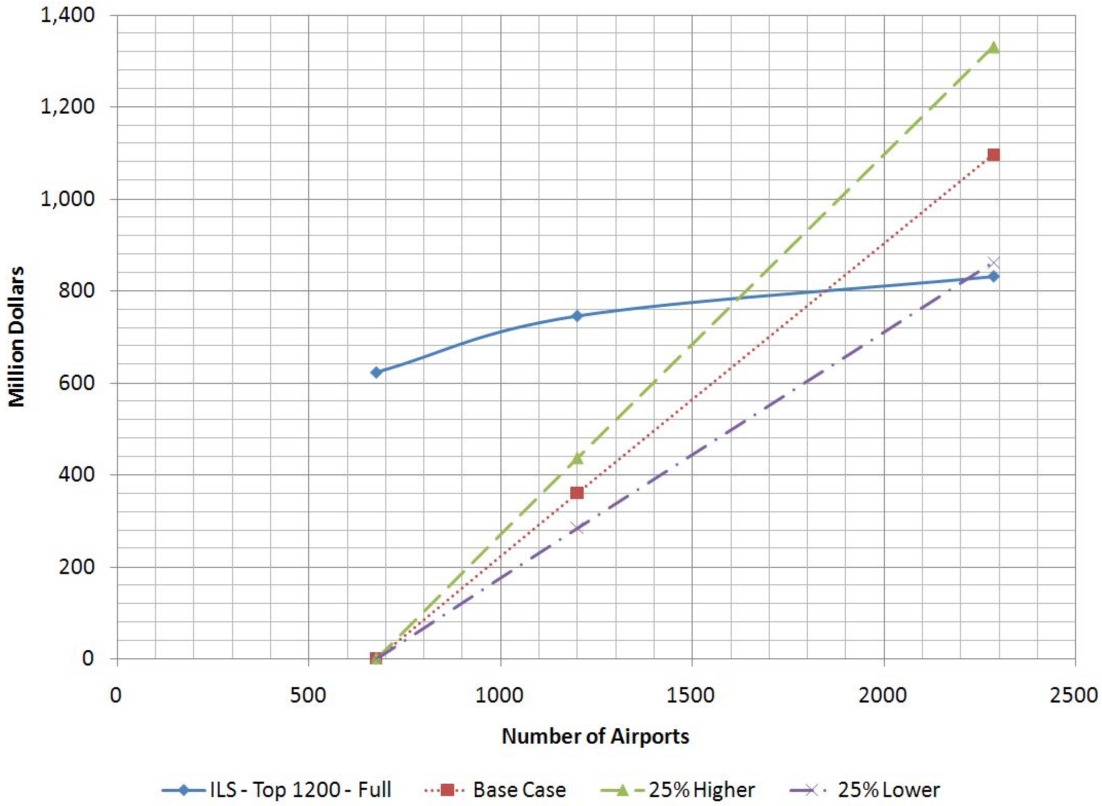

Figure 7-3: Sensitivity Analysis, (ILS – Top 1,200 GA – Full Airport Set Case).

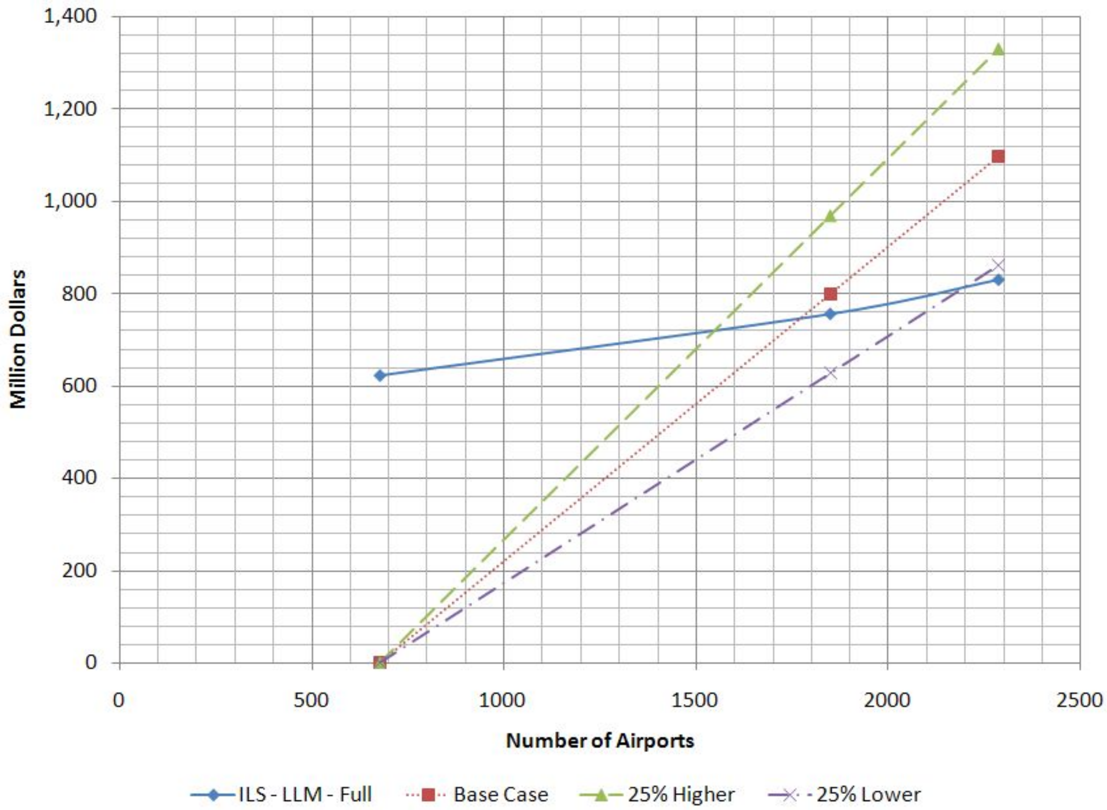

Figure 7-4: Sensitivity Analysis, (ILS – LLM – Full Airport Set Case).

Environmental Impacts

Using predicted noise footprints of the very light jets and using aircraft substitution methods for SATS single and multi-engine, piston-powered aircraft, airport noise contour maps are derived using the SATS demand function predicted by TSAM. Five case studies are conducted at airports located at both metropolitan area and rural communities. The selected airports are representative of the spectrum of airports with future SATS enabled technologies. In our analysis, we identify specific operating patterns at those airports. When the local information is unavailable, assumptions are generally made based on pilot anecdotal experience. It is important to point out that general aviation traffic at non-towered airports is more difficult to characterize than commercial traffic at towered airports. Many general aviation airports do not have accurate records on the number of both local and itinerant general aviation operations. At many of these small airports the Terminal Area Forecasts (TAF) data is generally unreliable. The results are shown in Table 7-4.

Table 7-4: Noise Case Study Results.

Airport ID	GPA (Degrees)	Baseline Scenario		SATS Scenario VLJ Only		SATS Scenario Mixed SATS Fleet	
		Population Influenced	Area (Acres)	Population Influenced	Area (Acres)	Population Influenced	Area (Acres)
TEB	3	5,050	1,517.7	5,746	1,624.9	6,027	1,692.6
	4	5,050	1,517.7	5,627	1,617.4	6,027	1,687.1
	5	5,050	1,517.7	5,627	1,612.5	5,998	1,683.7
HEF	3	0	188.4	0	273.7	0	303.7
	4	0	188.4	0	269.4	0	300.4
	5	0	188.4	0	266.0	0	297.8
DAN	3	0	67.9	0	82.1	0	81.9
	4	0	67.9	0	81.8	0	81.8
	5	0	67.9	0	81.7	0	81.7
BCB	3	0	84.7	0	89.3	0	93.7
	4	0	84.7	0	89.2	0	93.7
	5	0	84.7	0	89.1	0	93.6
GLD	3	0	115.0	0	116.0	0	116.0
	4	0	115.0	0	116.0	0	116.0
	5	0	115.0	0	116.0	0	116.0

In the emission analysis, Single-Engine aircraft are modeled using the Socata Tampico and Multi-Engine by both the Cessna Conquest II and the Beech King Air 350. All SATS operations are assumed to follow itinerary operations and no touch-and-go operations. Sensitivity analyses are conducted to estimate the impacts of different fleet mix and approach glide path angles in the noise contours around airports. Regression techniques are applied to estimate emission indices of the VLJ.

Estimates of major pollutant emissions using the EDMS suggest that the typical rural airport will experience an increase of 1 - 4% Carbon Monoxide (CO), 1 - 7% Total Hydrocarbon (THC) and 1 - 7% Non-Methane Hydrocarbons (NMHC), 5 - 60% Nitrogen Oxides (NOx) and 5 - 80% Sulfur Oxides (SOx). The typical urban airport will experience an increase of 1 - 4% Carbon Monoxide (CO), 2 - 8% Total Hydrocarbon (THC) and 2 - 8% Non-Methane Hydrocarbons (NMHC), 5 - 80% Nitrogen Oxides (NOx) and 5 - 90% Sulfur Oxides (SOx). The absolute amount of pollutants and relative percentage of increase due to predicted SATS operations are shown in Figure 7-5 to Figure 7-12.

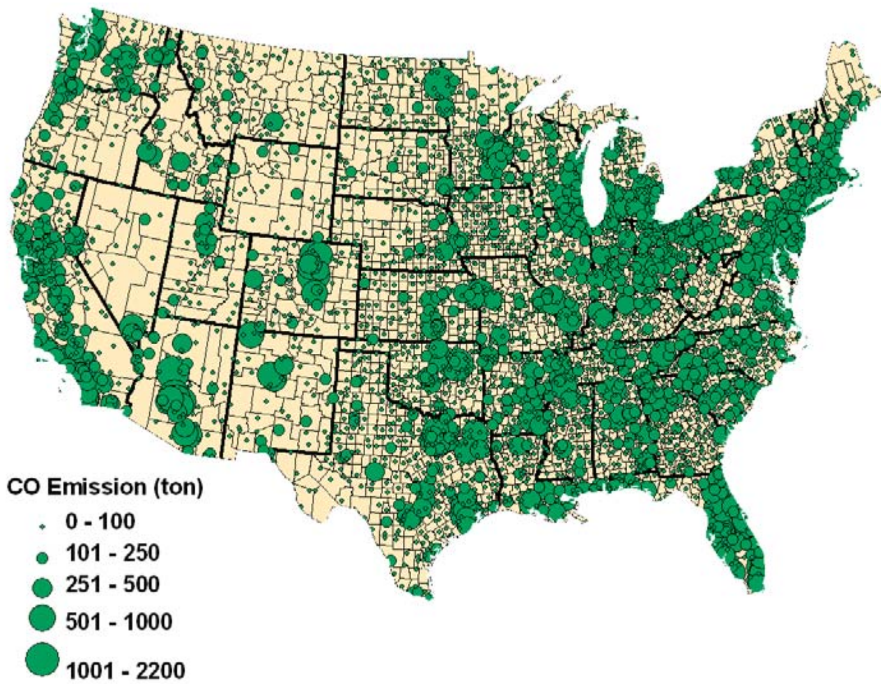

Figure 7-5: CO Emission Tonnage.

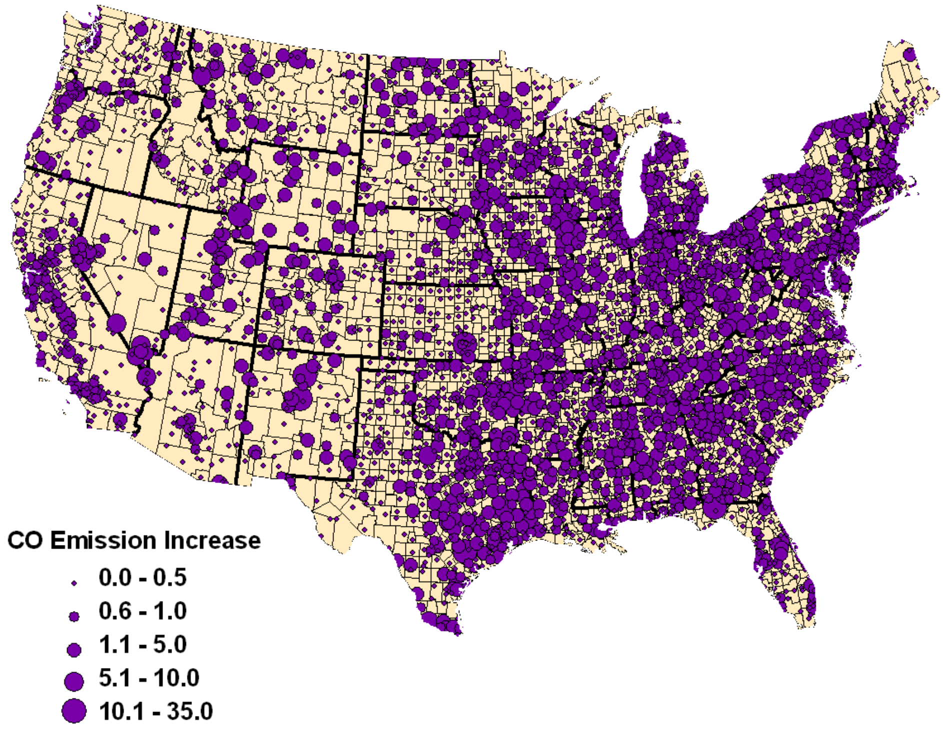

Figure 7-6: CO Emission Percentage of Increase.

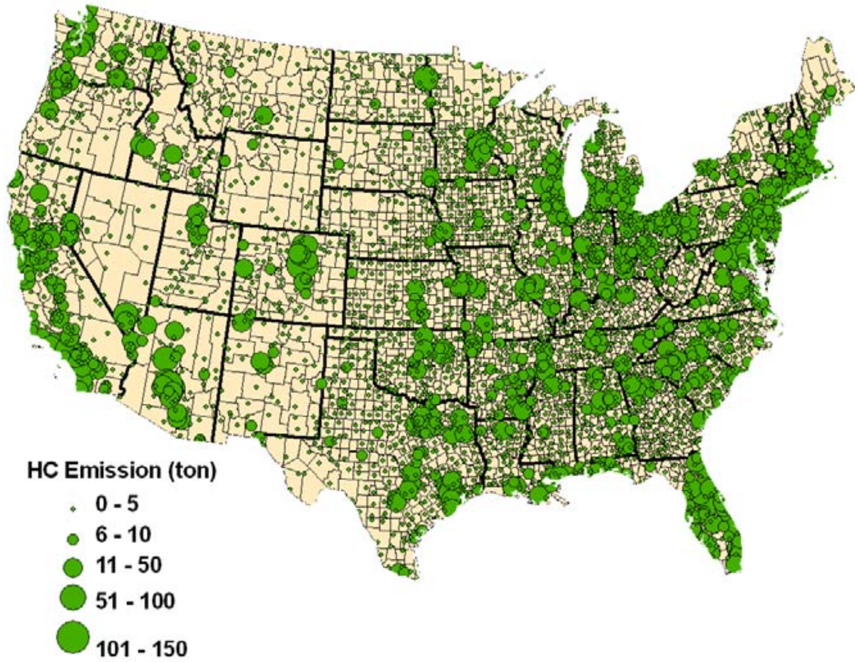

Figure 7-7: HC Emission Tonnage.

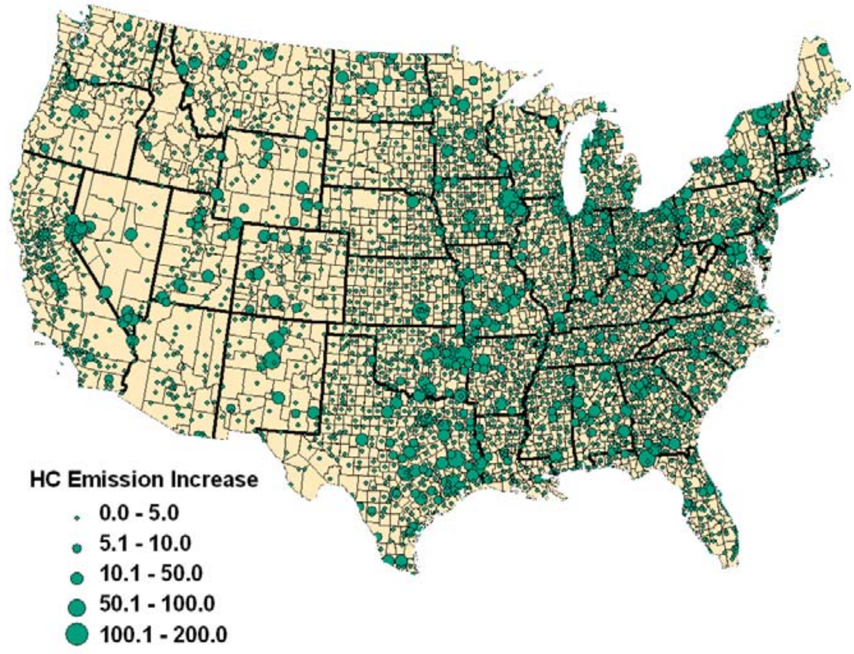

Figure 7-8: HC Emission Percentage of Increase.

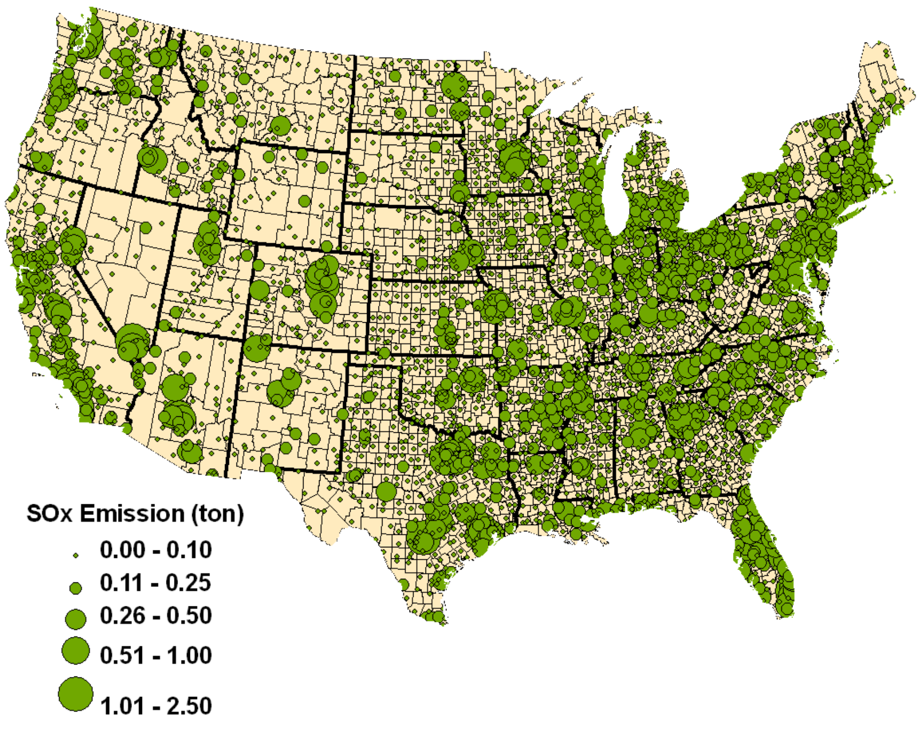

Figure 7-9: SOx Emission Tonnage.

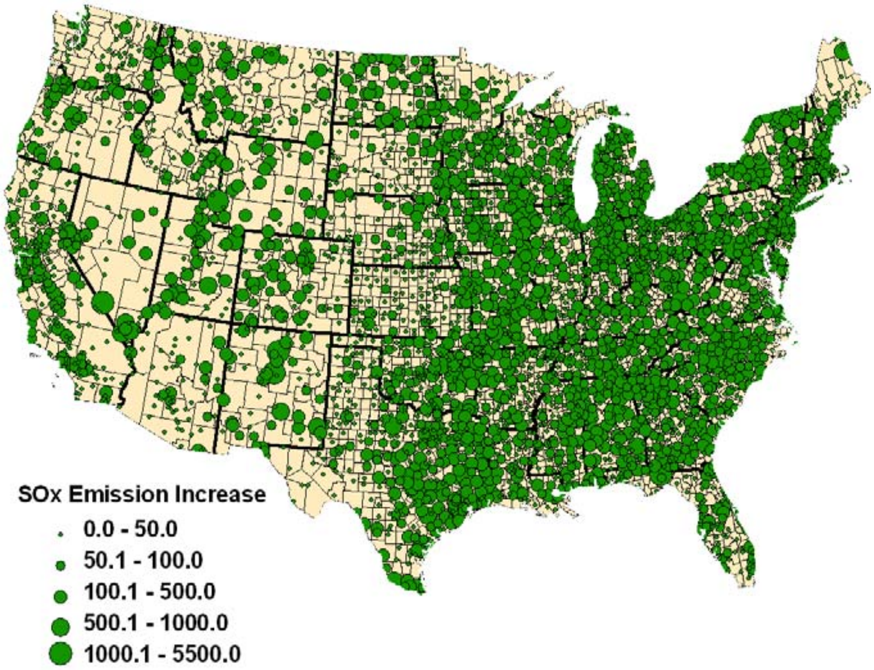

Figure 7-10: SOx Emission Percentage of Increase.

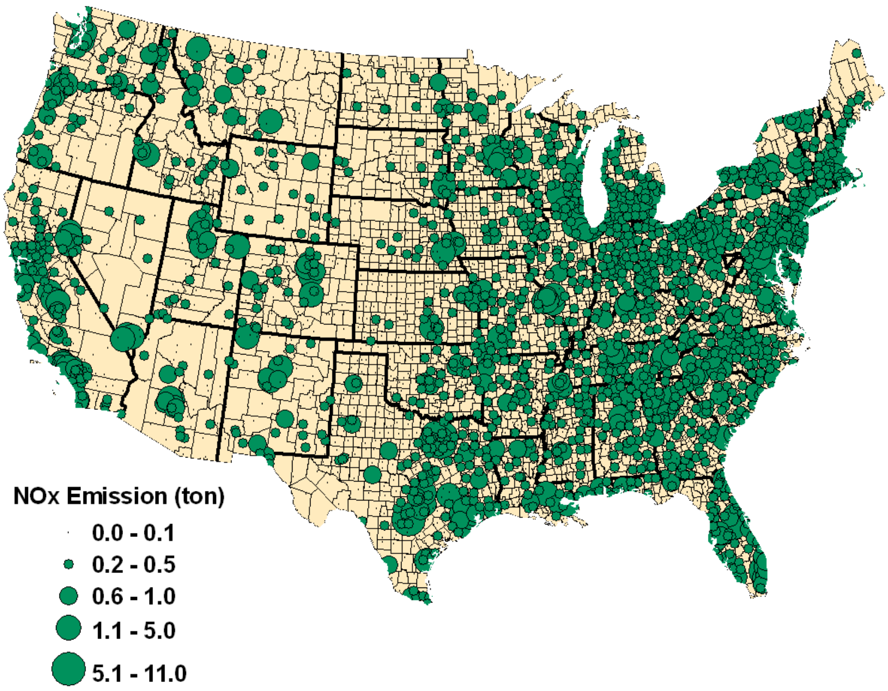

Figure 7-11: NOx Emission Tonnage.

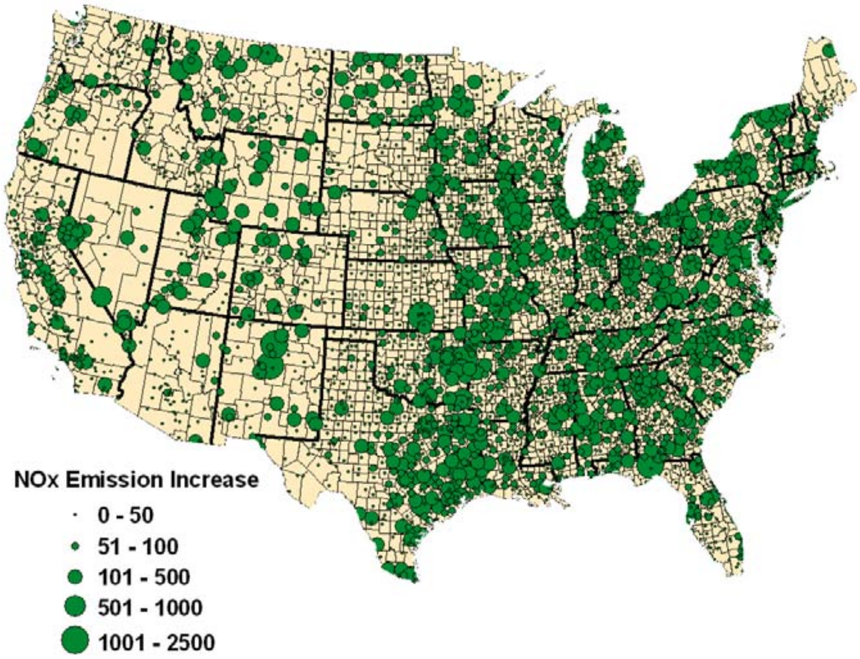

Figure 7-12: NOx Emission Percentage of Increase.

DataComm Analysis

DataComm case studies consist of two parts: delay and capacity improvement. The first part uses FAA Air Traffic Organization (ATO) current schedule (Year 2006) and future schedule projections of Year 2014, 2018, 2020, and 2025 as demand input. The second study applies computer generated demand instead of real schedule to assess capacity gains introduced by DataComm. The demand profile follows the Poisson distribution with assumed arrival or departure rate. The following sections will elaborate the results of both studies.

Delay Analysis

Currently the impact of DataComm is assessed by a reduction of inter-arrival delivery errors at arrival fixes (arrival post nodes). Three intrail separation values are simulated, that is, 9nm (without DataComm), 8nm (with DataComm) and 7nm (with DataComm). A two percent reduction in the final approach separation is also assumed as a result of DataComm. SIMMOD simulations are exercised using ATO current and future schedules. The outcome indicates that the maximum throughput of the New York terminal area is around 250 operations per hour (Figure 7-13, Figure 7-15 and Figure 7-17). Simulations are run for current and future scenarios. Year 2020 and 2025 demand scenarios fail to run because of excessive delay. It implies unless appropriate measures are taken, New York airports cannot handle the projected demand. Each data point in the following figures represents an average of 5 simulation iterations.

Current airport operation generates moderate delay (Figure 7-14). With the introduction of DataComm technology, however, delay increases compared with non-DataComm scenario. This is because DataComm is projected to benefit incoming traffic primarily. By introducing more arrivals, departure traffic on the ground find it more difficult to allocate sufficient gaps to takeoff as the arrival traffic always has the priority to land. Therefore, arrival capacity improvement brought by DataComm is offset by departure delays.

Future scenarios produce similar delay for both scenarios with and without DataComm technology (Figure 7-16 and Figure 7-18). This suggests DataComm is unable to improve overall supply capacity at near or over-saturated airspace.

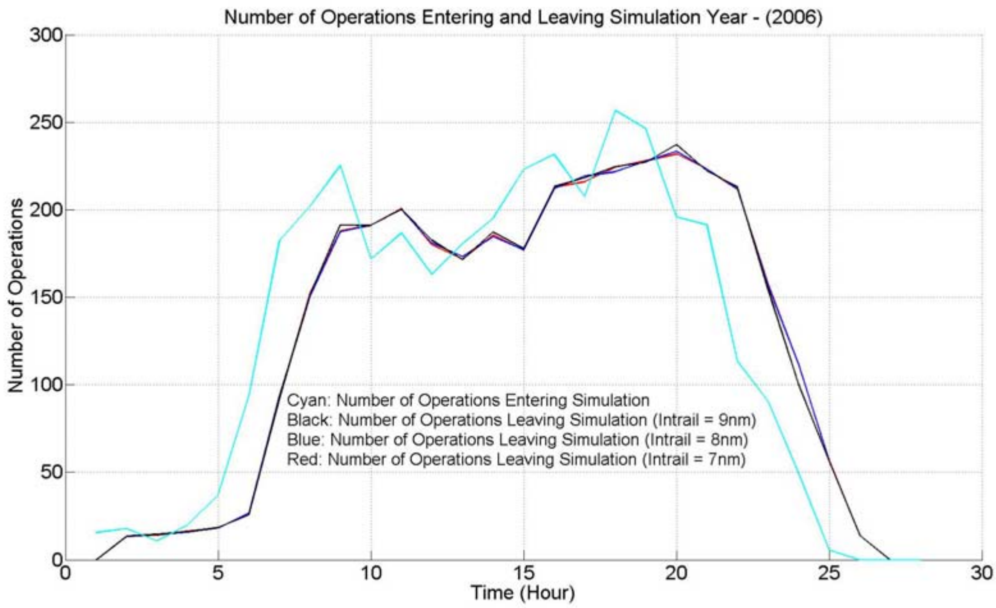

Figure 7-13: Number of Operations Entering and Leaving Simulation – 2006.

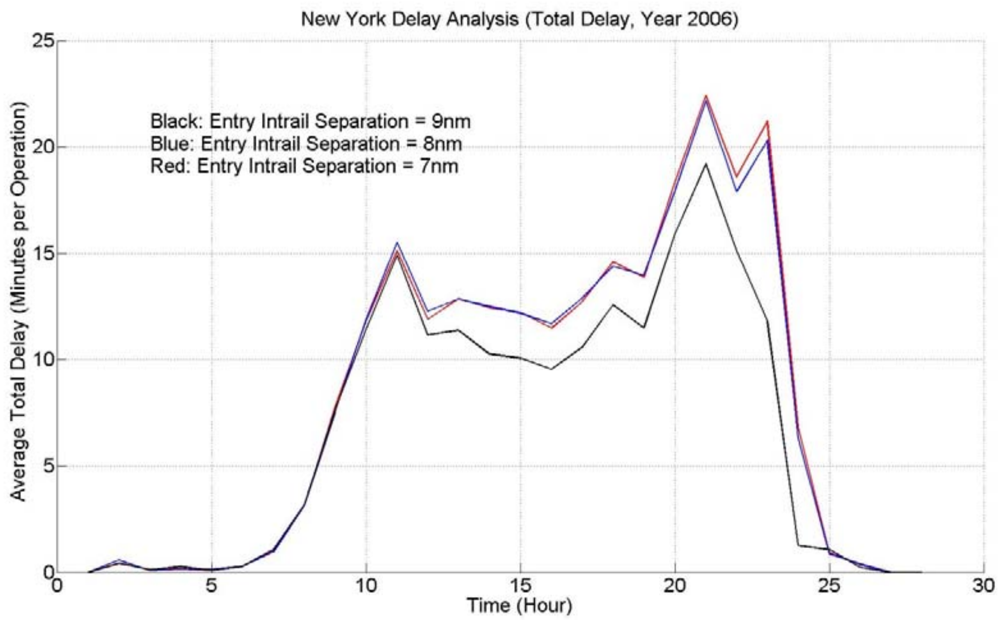

Figure 7-14: Average Total Delay – New York Area, Year 2006.

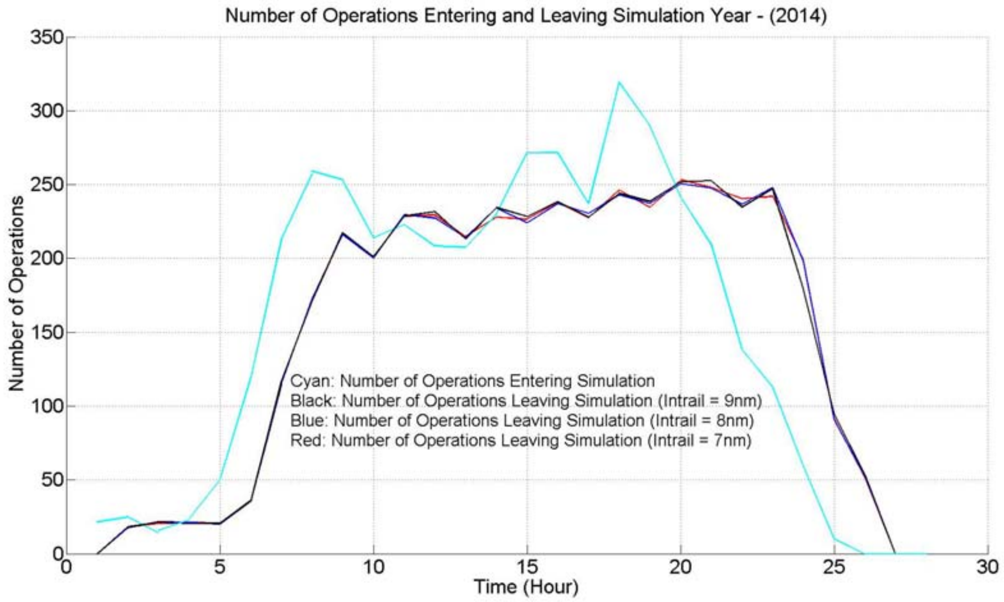

Figure 7-15: Number of Operations Entering and Leaving Simulation – 2014.

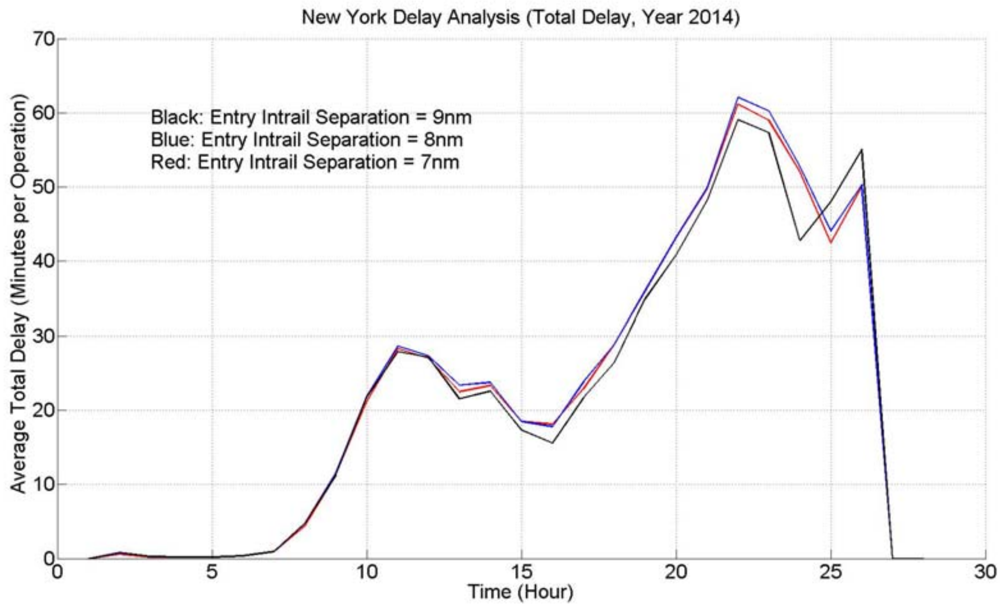

Figure 7-16: Average Total Delay – New York Area, Year 2014.

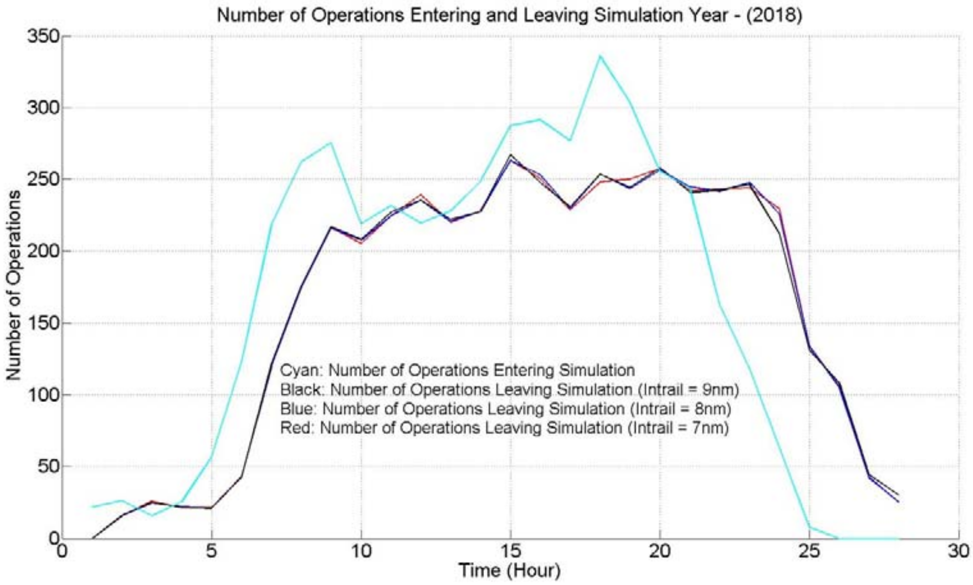

Figure 7-17: Number of Operations Entering and Leaving Simulation – 2018.

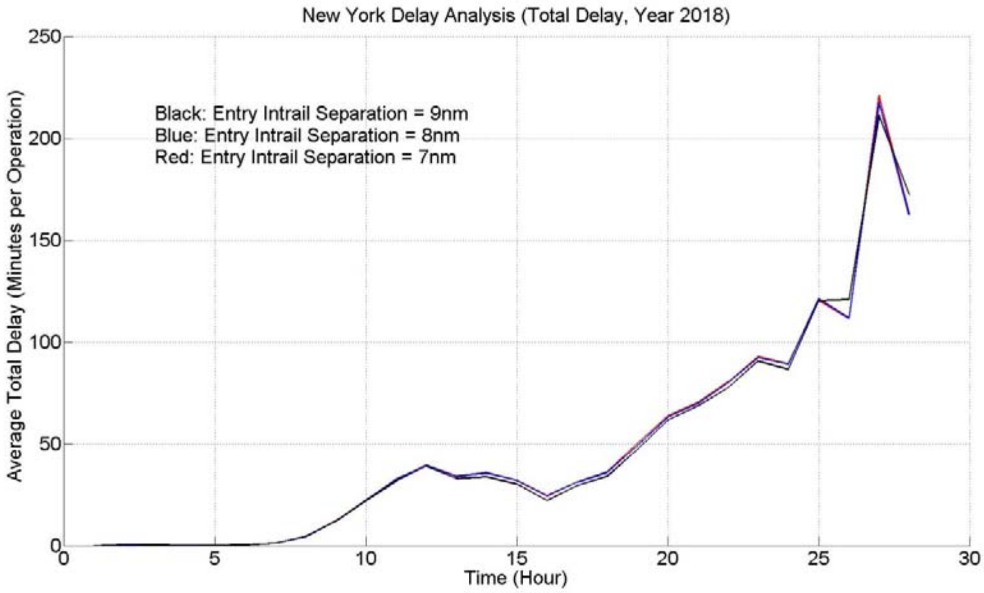

Figure 7-18: Average Total Delay – New York Area, Year 2018.

In addition, each airport in the New York area, John F. Kennedy Airport (JFK), La Guardia Airport (LGA) and Newark Liberty Airport (EWR), is simulated individually with current and future schedule. The outputs of the simulation outputs are listed in Appendix B.

Among the three focus airports, JFK generates the lowest delay, followed by EWR, and LGA produces the highest delay. The throughput of JFK is around 80-90 operations per hour. EWR is capable to accommodate 90-100 operations per hour. And at LGA, the capacity is around 70-80 operations per hour. This is consistent with the overall 250 operations per hour capacity for the whole terminal area.

At EWR, air delay is larger than ground delay (Figure 9-3, 25, and 29) and arrival delay is larger than departure delay (Figure 9-4, 26 and 30). At JFK, these two types of delays are comparable (Figure 9-15, 37, 41, 34, 38, and 42). However, ground and departure delays dominate at LGA (Figure 9-27, 49, 53, 46, 50 and 54). As LGA account for most of the overall delays in the New York area, total ground and departure delay are the higher compare with air and arrival delays.

The overall result suggests that DataComm is unable to improve supply capacity at near or over capacity airspace such as New York area. The dominant factor in such terminal area is the final approach separation and departure delays. A two-percent reduction at the final approach segment is not significant enough to introduce noticeable positive impact.

Capacity Improvement

Demand in this step is generated randomly using the Poisson distribution. A series of arrival and departure is generated and used as demand inputs. Delays of each scenario are recorded. Each data point represents an average of 5 iterations. Original separation matrix is maintained for baseline scenario as well as 9nm intrail separation at entry points. For DataComm cases, a two percent reduction is assumed. Two intrail separation values are examined, i.e. 7nm and 8nm. Separation matrix observed in PDARS (Figure 6-6 to Figure 6-9) is applied to each airport.

Table 7-5: Summary of Maximum Expected Arrival Capacity Improvement (VMC).

Airport	Baseline Arrival Capacity (5-min delay/acft)	Data Link Arrival Capacity (5-min delay/acft)	Change (%)
LGA	28.8	30.2	4.9
EWR	28.7	29.8	3.8
JFK	30.3	30.4	0.3

The observed improvements in arrival saturation capacity with data link vary from 0.3 to 4.9% (Table 7-5). The JFK simulation results show little improvement in reductions of in-trail distances below 8 nm (at metering fix). This is driven by the larger expected value of in-trail separation near the Final Approach Fix (FAF) compared to EWR and LGA (dictated by the large percent of heavy aircraft).

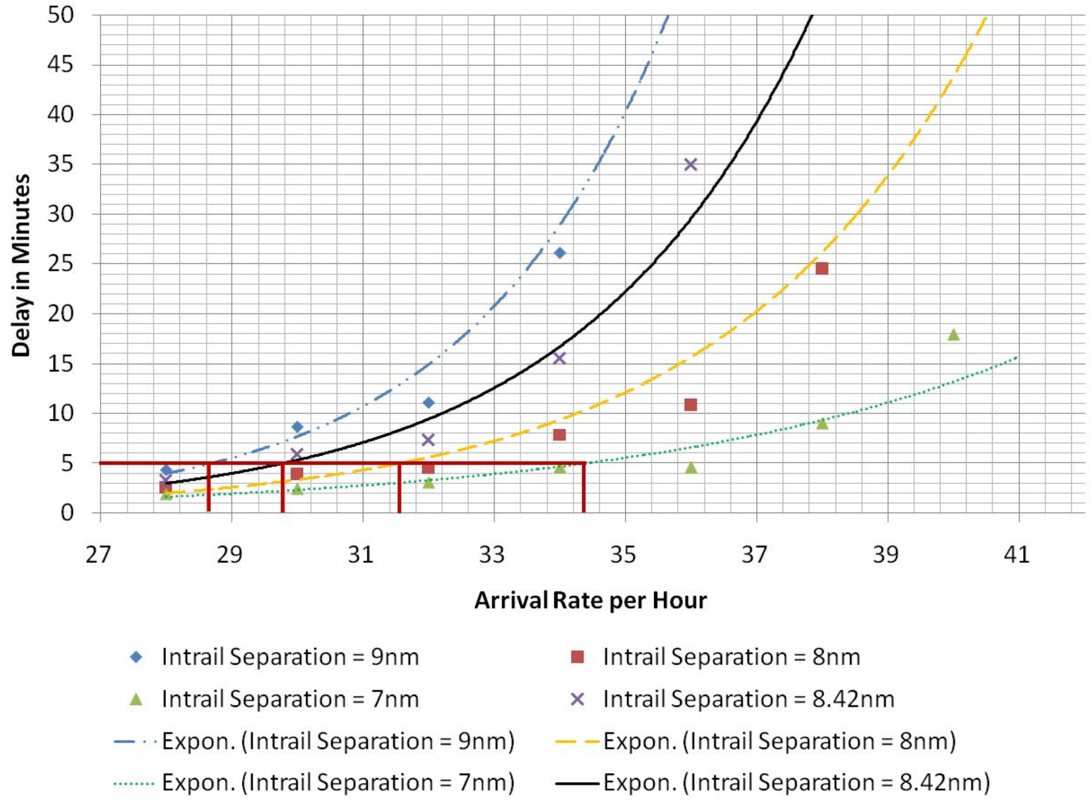

Figure 7-19: EWR Arrival Capacity Improvement Analysis.

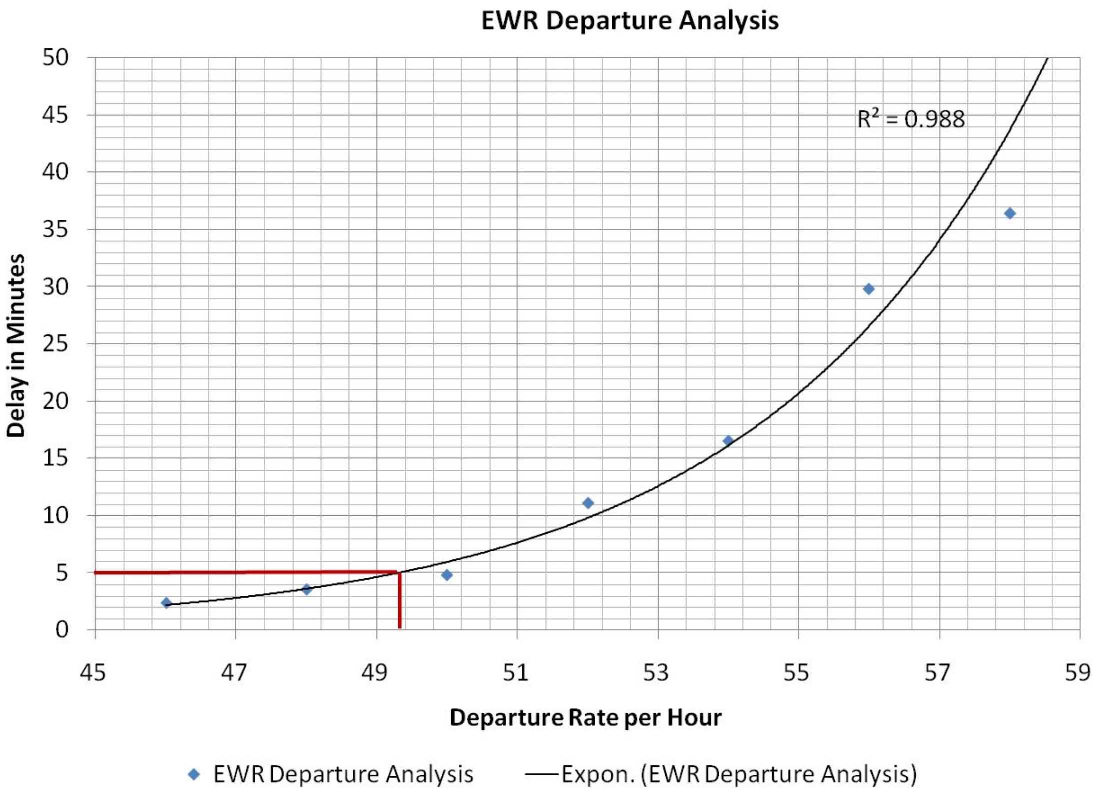

Figure 7-20: EWR Departure Capacity Analysis.

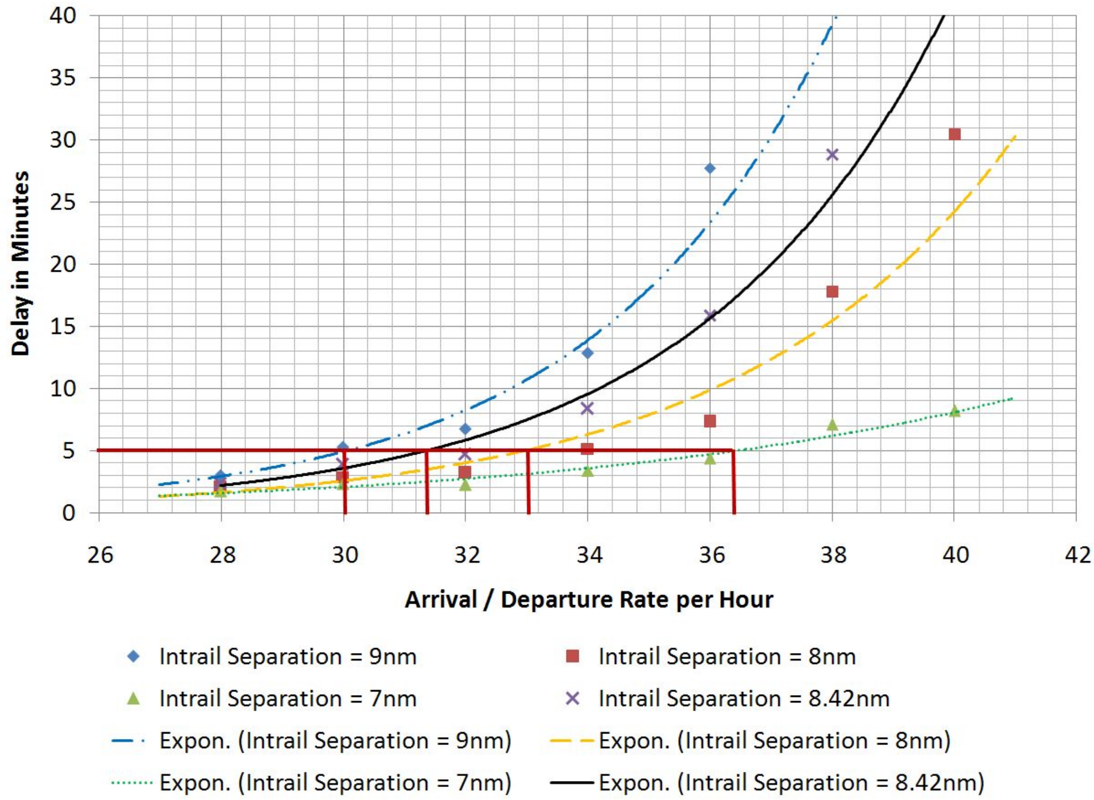

Figure 7-21: EWR Mixed Operation Capacity Improvement Analysis.

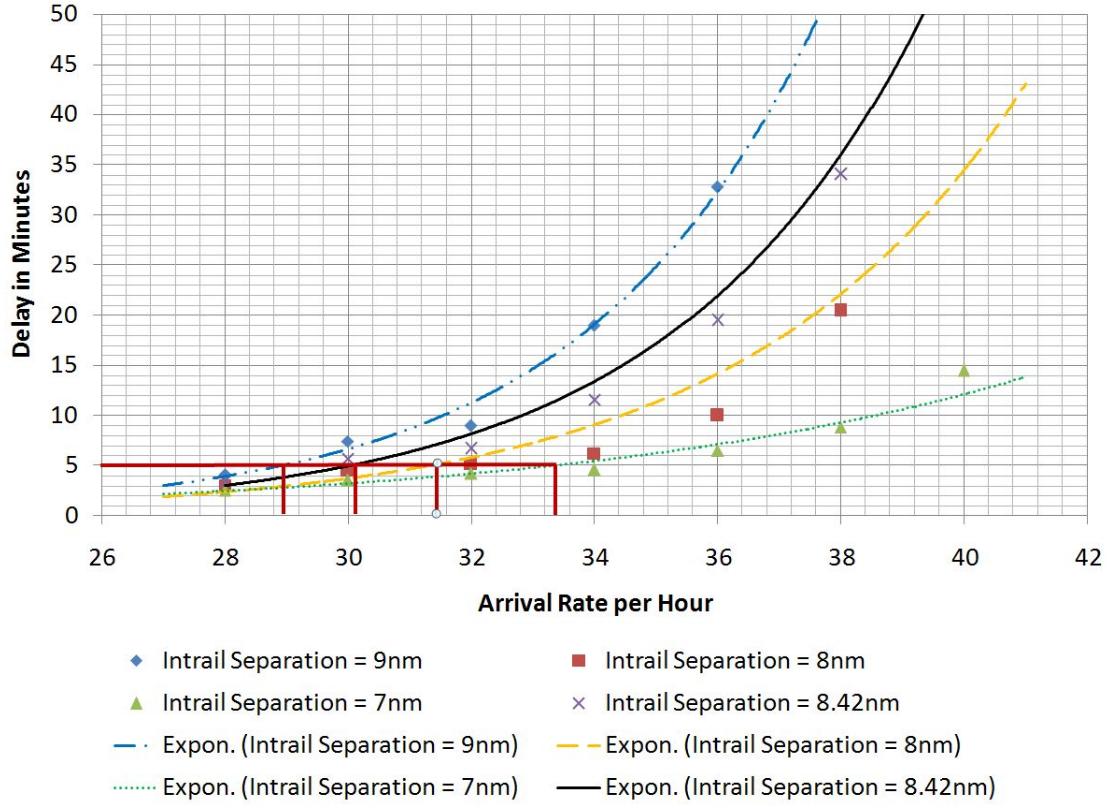

Figure 7-22: LGA Arrival Capacity Improvement Analysis.

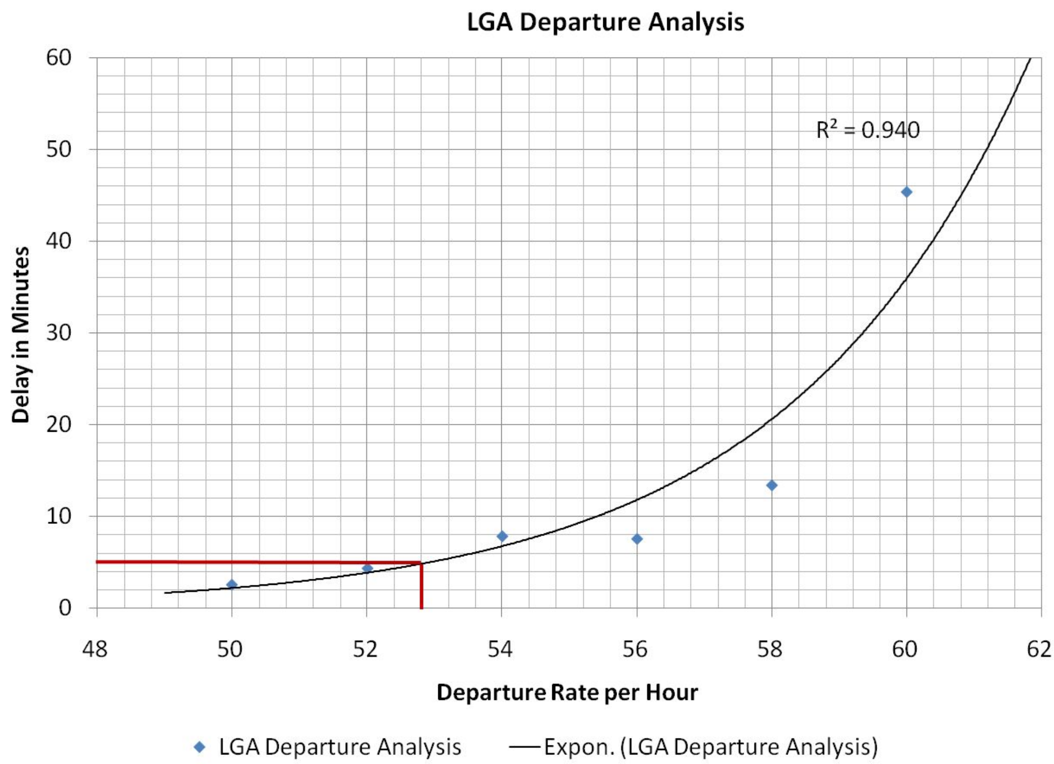

Figure 7-23: LGA Departure Capacity Analysis.

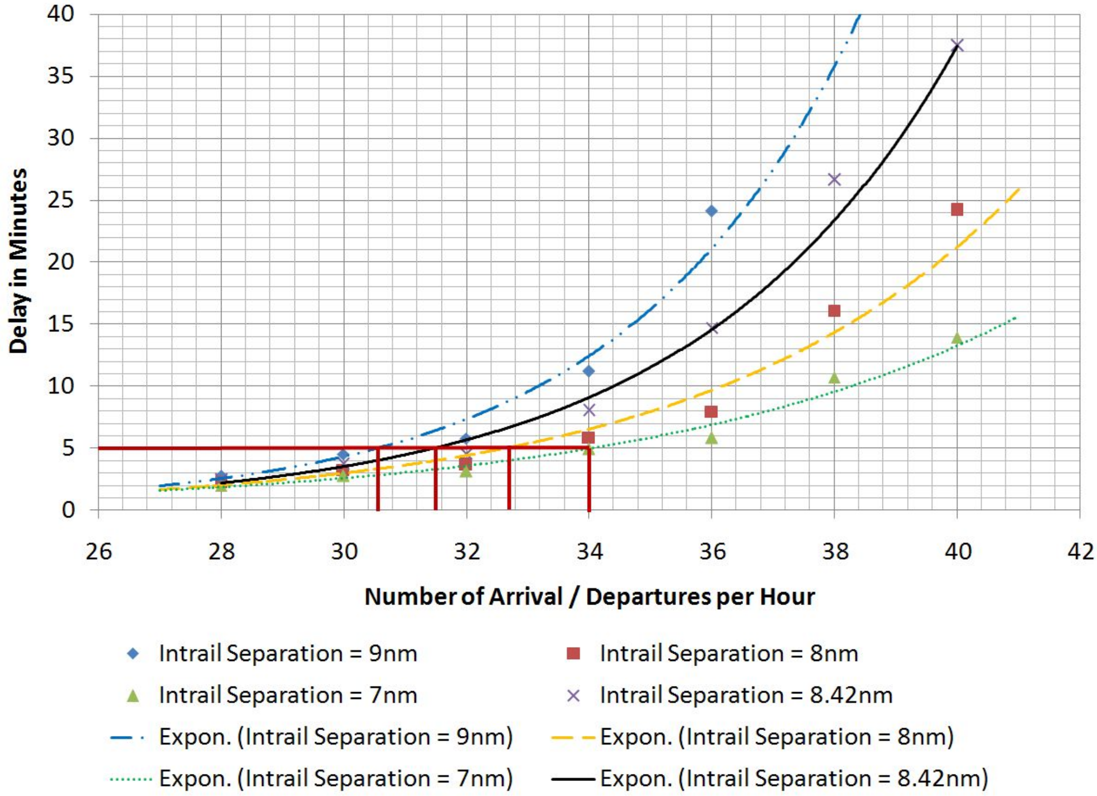

Figure 7-24: LGA Mixed Operation Capacity Improvement Analysis.

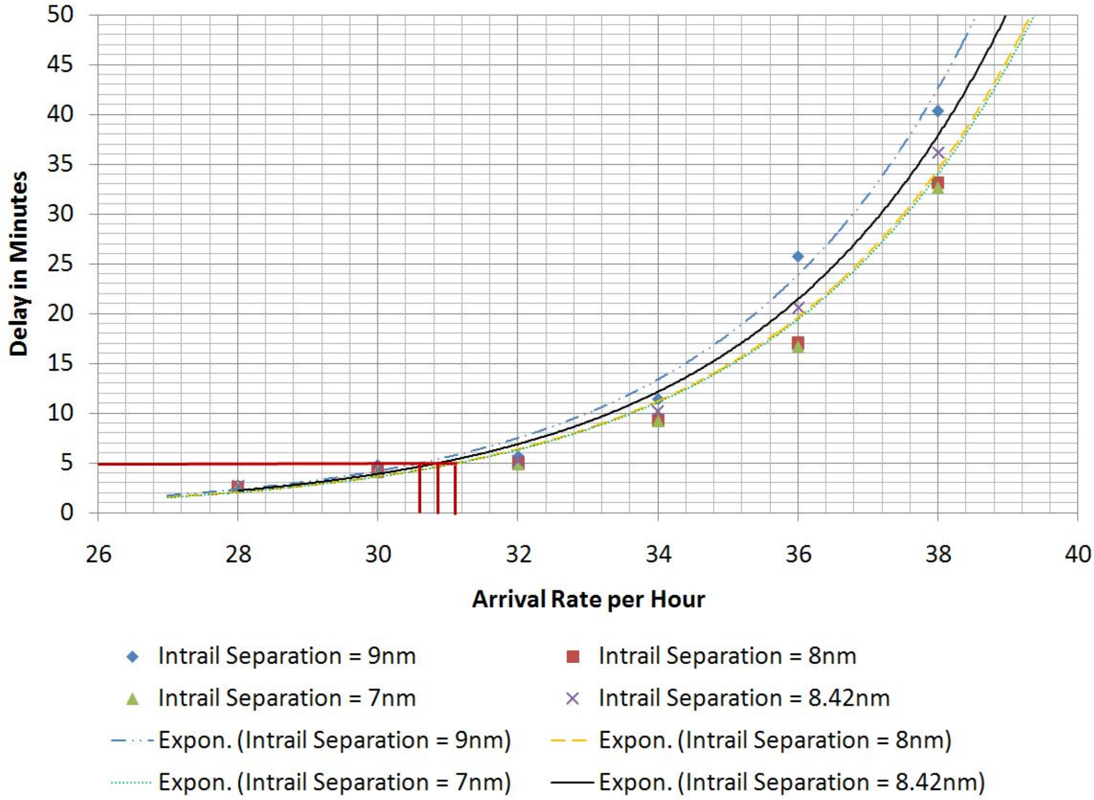

Figure 7-25: JFK Arrival Capacity Improvement Analysis.

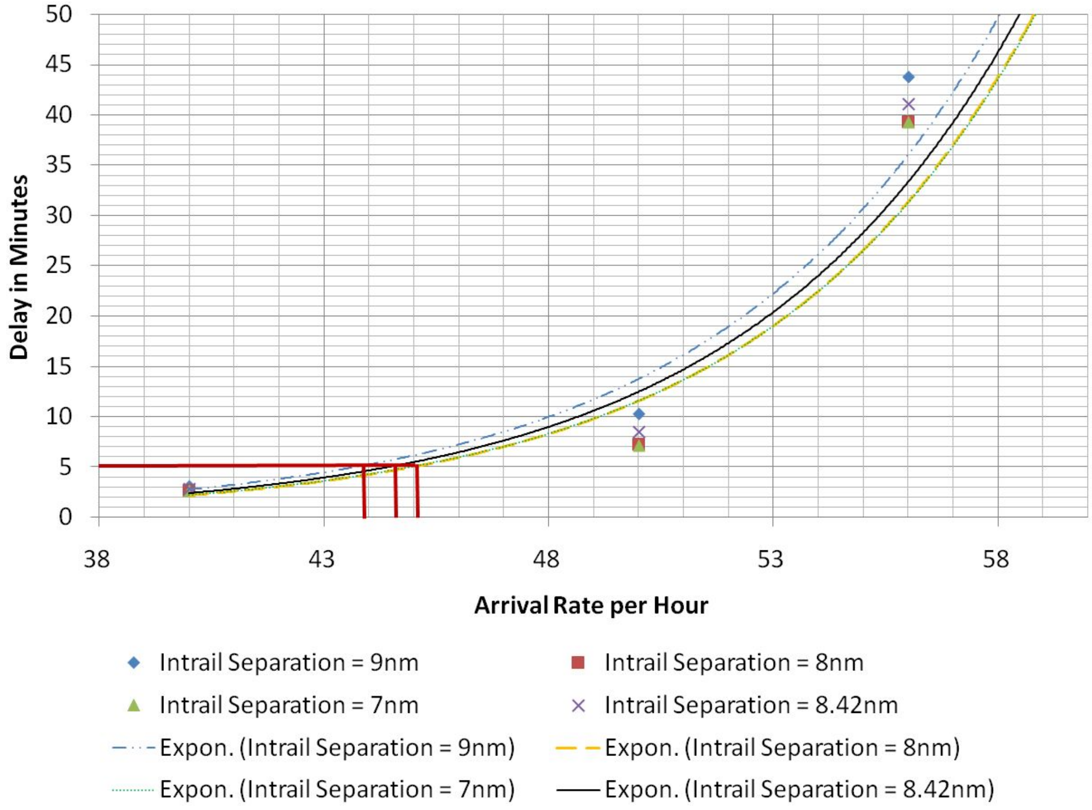

Figure 7-26: JFK Mixed Operation Capacity Improvement Analysis.

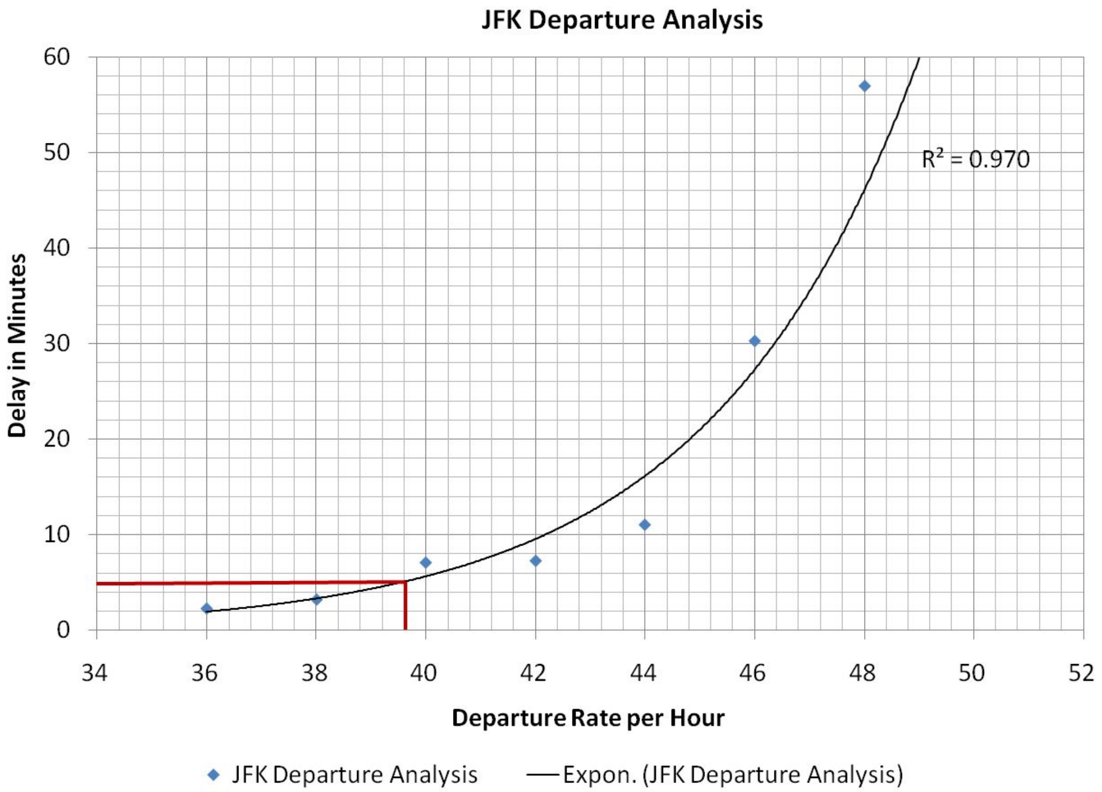

Figure 7-27: JFK Departure Capacity Analysis.

Consumer Surplus

The TSAM model is used to estimate demand at JFK, LGA and EWR for different delay scenarios. Demand is estimated at 5, 10, 30 and 60 minutes delay level.

The demand and delay at peak hours are used as the base case demand and delay (no improvement). The demand and delay scenario of the mixed operation simulation is applied (see Figure 7-21, Figure 7-24 and Figure 7-26). For example, peak hour delay at EWR is around 8 minutes (see Figure 9-2). The average number of peak hour operations is around 75 (see Figure 9-5). It suggests EWR operates at around 75 operations per hour and generates around 8 minutes delay. Assuming the number of arrivals and departures is the same, it translates to 37.5 arrivals and 37.5 departures per hour. Therefore the demand curve can be located using this operation point and the regression function. The same procedure is applied to LGA (Figure 7-29) and JFK (Figure 7-30).

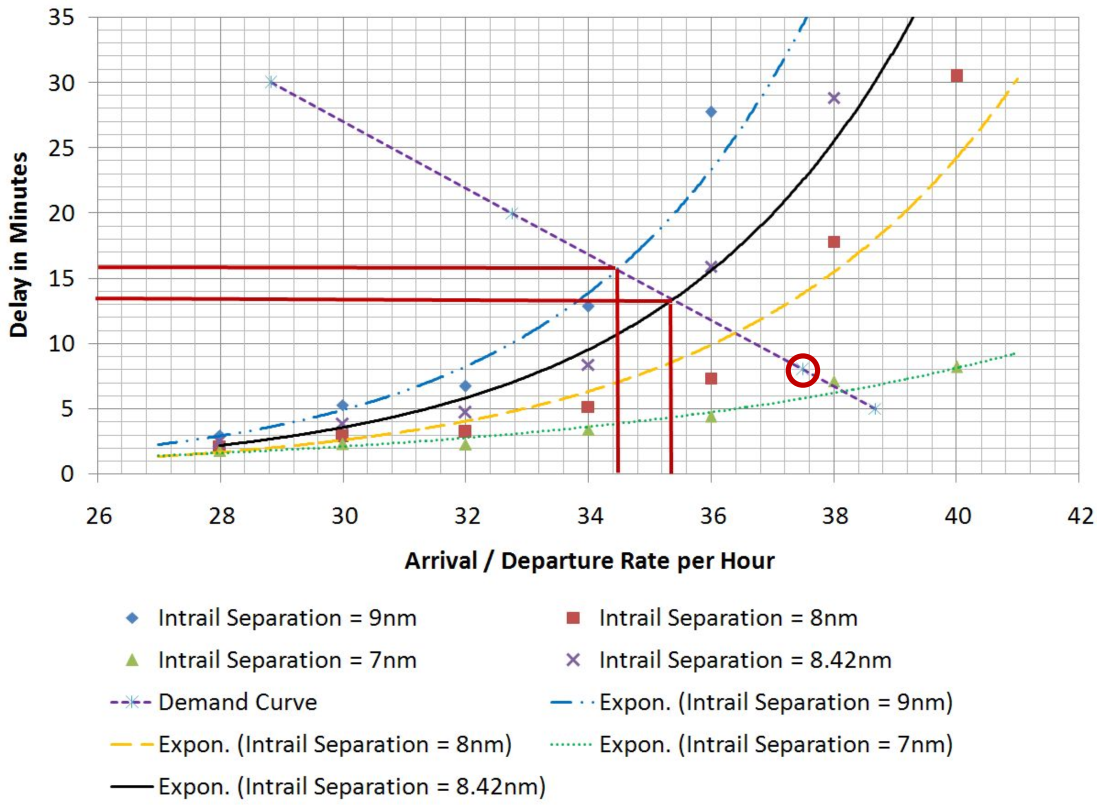

Figure 7-28: DataComm Consumer Surplus at EWR.

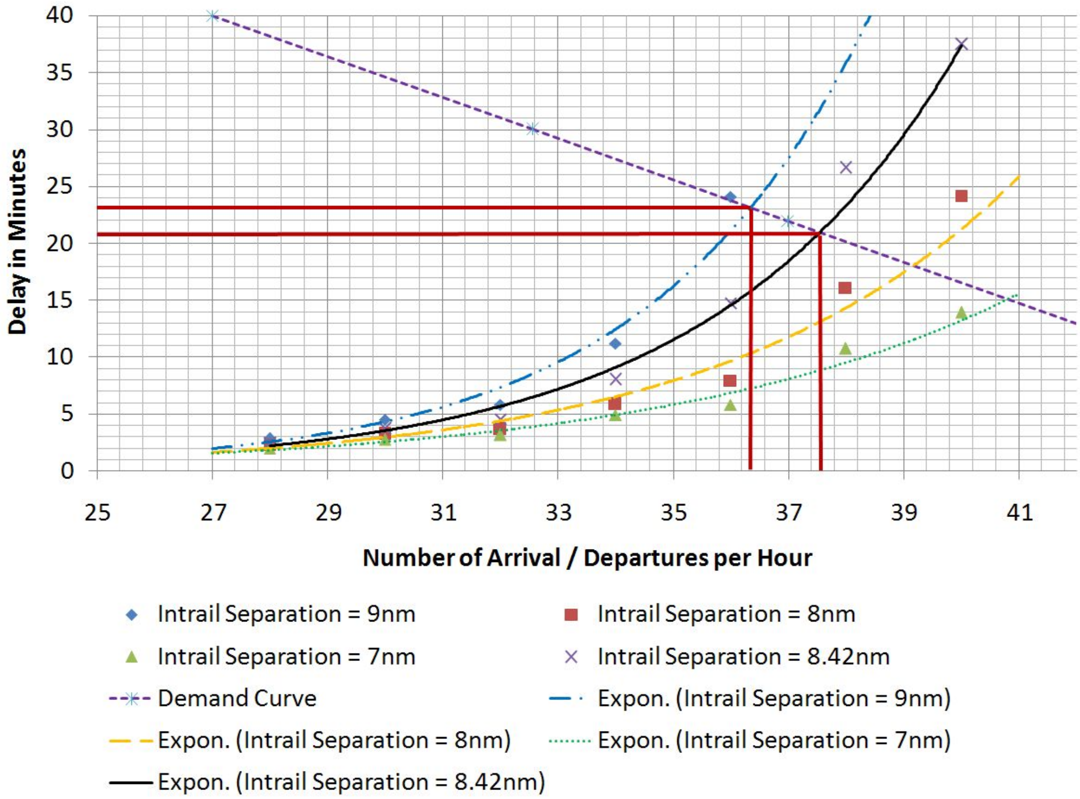

Figure 7-29: DataComm Consumer Surplus at LGA.

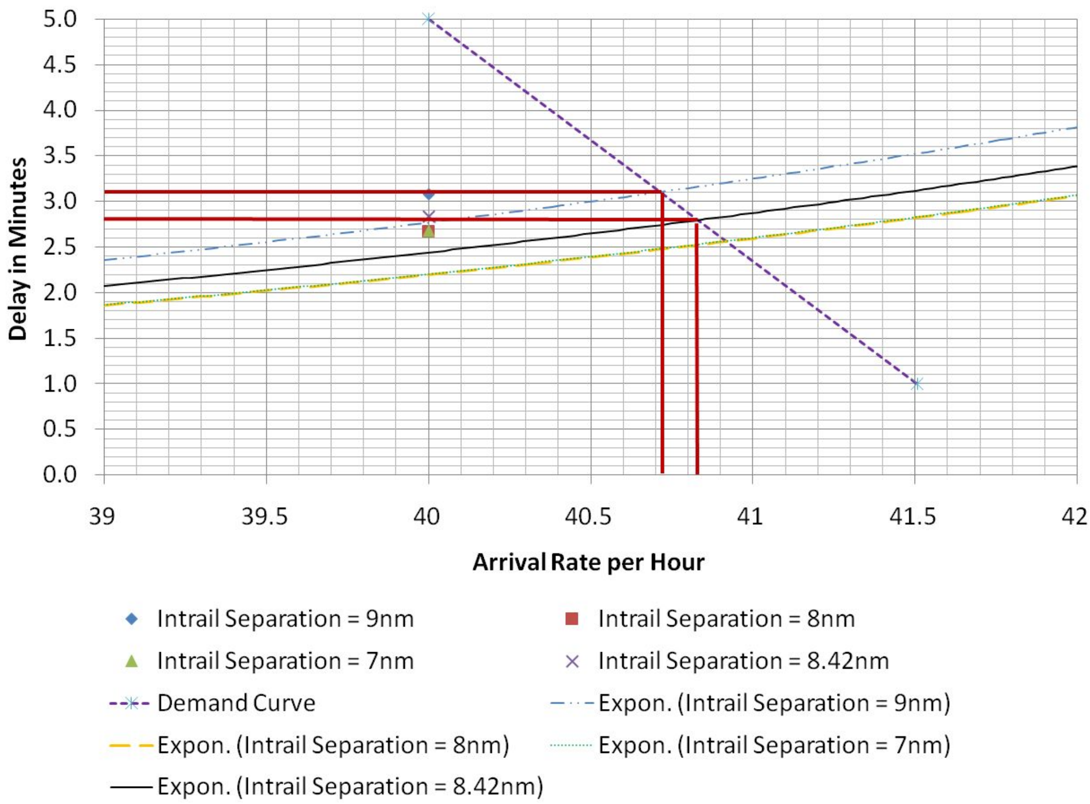

Figure 7-30: DataComm Consumer Surplus at JFK.

The simulation shows DataComm is able to reduc delay at the New York airports using computer generated demand. In order to estimate the introduced benefits, the components of delay costs must be defined. According to FAA, the cost of delay include airline costs (more aircraft operating hours), passengers (extra travel times), cargo shippers, and other people on the ground (FAA, 2001). In this analysis, the first two factors, airline operating costs and passenger time are considered. FAA provides a cost-benefit guildline that specifies values for important economic parameters (FAA, 2001). In this guideline, FAA suggests the air carrier operating costs to be $2,766 per hour (Year 2001) and passenger VOT $28.6 per hour (Year 2000). They can be converted to $3,207 and $34.15 per hour in terms of year 2006 dollars. In addition, other parameters including delay time savings and average number of passenger per operation can be estimated in Table 7-6. The average passenger per operation is estimated from the real operation record (The Port Authourity of NY&NJ, 2006).

Table 7-6: Travel Time Savings and Average Number of Passengers per Operation.

Airport	Time Savings (hr)	Average Number of Passenger per Operation
LGA	2 / 60	52.3
EWR	2.5 / 60	58.4
JFK	0.3 / 60	86.2

Using these values and the concept of consumer surplus (Figure 2-1), the benefit of DataComm to the three analyzed airports is estimated as follows:

User Benefits:

LGA: [(36.5 + 37.5) * 2 * 52.3 * 2/60 * 34.15] / 2 = $4,406

EWR: [(34.4 + 35.4) * 2 * 58.4 * 2.5/60 * 34.15] / 2 = $5,800

JFK: [(34.4 + 35.4) * 2 * 86.2 * 0.3/60 * 34.15] / 2 = $1,027

Airline Benefits

LGA: 37.5 * 2 * 2/60 * 3,207 = $8,018

EWR: 35.4 * 2 * 2.5/60 * 3,207 = $9,461

JFK: 35.4 * 2 * 0.3/60 * 3,207 = $1,135

Total Benefits

LGA: $4,406 + $8,018 = $12,424

EWR: $5,800 + $9,461 = $15,261

JFK: $1,027 + $1,135 = $2,162

Chapter 8 : Conclusion and Recommendations

Conclusions

Based on the two case studies described above, the proposed TSAM based cost-benefit analysis framework provides a solid tool to assist national aviation investment decision making. Relationship between demand and supply is established using TSAM, thus enables proper cost-benefit analysis. Impacts of new technology such as Lower Landing Minima and DataComm can be evaluated based on the travel time and cost change. In addition, the environmental impacts can be assessed using the standard FAA software package.

Recommendations

LLM Analysis

This case study represents a nation-wide cost-benefit study. More detailed study should be conducted if relevant information including terrain, obstacles, weather pattern**,** etc, is available. The following recommendations could assist further research efforts.

- Accessibility improvement should be evaluated using local weather information including ceiling and visibility statistics. Terrain should be considered in the design of GPS WAAS approach in addition to critical obstacle analysis.
- Hard-to-quantify benefits such as accident reductions and schedule productivity improvement are not modeled as they are not the focus of this dissertation. External costs are estimated in terms of noise contour and emission quantity. These cost components have not been converted to monetary terms. This is due to the wide variation of the value conversion in the literature. Wang and Santini (1995) conducted an extensive review on the economic values of emission pollutants. The surveyed values range from ten times greater to one tenth as much as median. If a consensus is reached in the future, these components should be integrated in the cost-benefit

analysis. Fuel savings result from flying more flexible approach path using WAAS are not considered. During the phase of infrastructure construction, the normal flow of traffic can be interferenced. This may introduce extra delay. This cost due to the construction is not considered.

- The cost-benefit analysis is built on the demand estimation capability of the TSAM model. The effectiveness of the CBA analysis is based on the accuracy of the projected demand. Future efforts should be taken to improve the precision of TSAM.
- The SATS LLM operational technologies under development would provide safety benefits to the aviation community even if the SATS program is not able to lure many passengers to switch to the air mode. The safety implications of LLM technologies should be studied using piloted and computer simulation studies.

External Cost Analysis

This study presents a first assessment of the potential noise impacts of SATS operations at various airports in the U.S. The following recommendations could be pursued to improve the analysis presented in this dissertation.

- The five airport studies represent a “pessimistic” scenario in that all SATS operations are assumed to be additional operations above the current GA activities at each airport. This is the result of a new paradigm on-demand services possible with the introduction of VLJ and advanced SE and ME piston aircraft. SATS aircraft including VLJs will probably have a replacement effect on existing aircraft technologies used today. Thus some of the older and noisier aircraft operating at these airports could probably be replaced by more environmentally friendly SATS aircraft. The authors are currently studying this replacement effect.
- The information on reliable local and itinerant operations at some rural airports is scarce. This is an issue that needs to be improved across NAS. The FAA Terminal Area Forecast (TAF) is unreliable and thus local information is preferred for airport impact studies. General survey and statistical data sources are preferred for nation-wide impact analyses. The analysis conducted in our study mixes local and general airport operational data as a best effort to model credible operations at each airport. Future efforts could use other data sources. For example, real-time or historical radar

track data will be helpful in building flight paths and improve the estimation of runway utilization.

- The noise signature of very light jets and new technologies should be continuously validated along with the flight test progress of the new SATS aircraft. At least two new generation VLJs will be certified in 2006 (the Eclipse 500 and the Cessna Mustang). Future noise impact studies will benefit from actual aircraft noise certification data.
- A more comprehensive nation-wide analysis is desired to address, at the national level, the impacts of SATS aircraft deployment. A method with virtual airports and virtual runways is being developed at Virginia Tech to address this concern. The method considers typical runway configurations and parametric SATS demands as two explanatory variables. Local adjustments will be made to account for specific local effects at all the airports or at other sets of airports studied.

DataComm Study

This analysis evaluates DataComm benefits at New York area. It is recommended to consider the following methods to improve future study.

- New York terminal area is chosen to evaluate DataComm benefits. This area represents one of the most congested airspace in the U.S. The capacity of this area is mainly constrained by the final approach separation and departure delays. For future analysis, other less congested area should be studied.
- DataComm technology is projected to reduce uncertainty in the terminal operations. Currently this capability is modeled by reducing inter-arrival delivery errors at arrival fixes (arrival post nodes) using SIMMOD. Three intrail separation values, 9nm, 8nm and 7nm, are assessed. In the next stage of this analysis, distribution with reduced variation ought to be applied instead of fixed values to model its impact more precisely.
- The simulation fails to run for scenarios beyond Year 2018 due to infrastructure capacity constraints. Measures should be taken to alleviate delays if the demand grows according to the FAA forecast. Then these measures will be integrated into the simulation to re-examine the benefits of DataComm.

References

Airnav website. http://www.airnav.com

Air Transport Association (ATA), *Air Travel Survey*, 1993.

Algers,. S., Dillen, J. L. and Widlert, S., *The National Swedish Value of Time Study*, presented at PTRC International Conference on the Value of Time, 1996.

American Association of State Highway and Transportation Officials (AASHTO), *a Manual on User Benefit Analysis of Highway and Bus-Transit Improvements*, 1977.

ASCE, *Guide to Transportation Benefit-Cost Analysis*, Committee on Urban Transportation Economics and Policy (CUTEP) of the Urban Transportation Division, http://ceenve.calpoly.edu/sullivan/cutep/cutep_bc_outline_main.htm

Aeronautical Survey Program Home Page, *Universal Data Delivery Format*, National Geodetic Survey (NGS) Aeronautical Survey Program, January 28, 2004, http://www.ngs.noaa.gov/AERO/aero.html, Accessed March, 2004.

AOPA, Aircraft Owners and Pilots Association, AOPA Welcomes Improved WAAS Minima – Urges FAA to Speed Approach Approvals and Members to Equip, http://www.aopa.org/whatsnew/newsitems/2006/060307waas.html, Mar 2006.

Arizona Department of Transportation – Aeronautics Division, *Five-Year Airport Development Program*, FY 2008-2012, 2006.

ATAC Corporation, *Heliport Noise Modeling Report Model Review and Program Plan*, Prepared for the FAA Office of Environment and Energy, April 28, 2000.

Baughcum, S. L., et al. *Scheduled Civil Aircraft Emissions Inventories for 1992: Database Development and Analysis, Appendix D: Boeing Method 2 Fuel Flow Methodology Description*. Report NASA CR 4700, The Boeing Company, April 1996.

Bradford, S. and Colligan, W., *National Airspace Resource Investment Model: Needs Analysis and Operational Concept Report*, Draft Report, Federal Aviation Administration, April 1996.

Brown, S. and Watkins W., *Measuring Elasticities of Air Travel form New Cross-sectional Data*, Civil Aeronautics Board Staff Paper, Aug 1971.

BUCHair (U.S.A) Inc., *BUCHair DATABASE* - JP Airline Fleets International, New York.

Bureau of Transportation Statistics, *Airline On-Time Performance Data*, CY 2007.

Carlier, S., and Smith J., *Advanced Emission Model (AEM3) v1.5, validation exercise #2*, Eurocontrol Experimental Center, 2004.

City of Belfast, *Airport Layout Plan*, http://www.cityofbelfast.org/muniairport.shtm, accessed 2008.

City of Morgantown, Regular Meeting Aug 15 2006 Minutes, http://www.morgantown.com/minutes/8-15-06.htm, accessed 2008.

Coffman Associates / Gilbertson Associates, *Wickenburg Municipal Airport Runway Extension and Parallel Taxiway Relocation Cost and Funding Summary*, Feb 2006.

CRA International, *The Economic Impact of Very Light Jets*, prepared for Eclipse Aviation, May 2006.

Crane, H.L. et. al, *A Tool for Estimating Airport Access Benefits from GPS-Based Instrument Approaches*. American Institute of Aeronautics and Astronautics (AIAA) 2001-4236, MITRE Corporation.

Danielson Airport & CHA, *Danielson Airport Master Plan*, Study Advisory Committee Meeting # 2, http://www.danielsonairportplan.com/meetings/5B3-AC2-Presentation.pdf, Nov 2007.

Daniel J.I., *Benfit-Cost Analysis of Airport Infrastructure: The Case of Taxiways*, Journal of Air Transport Management 8, P149-164, 2002.

Dar Al. Handasah Consultants—Shair and Partners, *Road user charging study*, Ministry of Public Works, Republic of Lebanon. Beirut, Lebanon, 1997.

Dare County Airport Authority, *Dare County Regional Airport (MQI) Airport Master Plan Update*, Prepared by Talbert & Bright Inc., 2005.

De Vany, *The Revealed Value of Time in Air Travel*, The Review of Economics and Statistics, Vol 56, No. 1, Feb 1974.

Delaware River & Bay Authority, *New Castle Airport Projects*, http://www.drba.net/capital/ncapt-projects.pdf, accessed 2008.

Delta Airport Consultants, Inc., *Instrument Approach Procedure Study*, Prepared for Department of Aeronautics, State of Nebraska, May 1999.

Department of Transportation, Office of the Secretary of Transportation, *Departmental Guidance for the Valuation of Travel Time in Economic Analysis*, Apr. 1997.

Dou L., Lee, D., Johnson, J., and Kostiuk, P., *A Small Aircraft Transportation System (SATS) Demand Model,* NASA/CR-2001-210874, Logistics Management Institute, McLean, Virginia, June 2001.

Emily Brandler, *Airport to Start $72 Million in Work in Three-Year Span: Extended Runway, Hangars, Taxiways among Projects Set in 2006 through 2008*, The Journal of Business, Special Reports, May 2006

Eurocontrol, *Approach to Access the Benefits and Costs of ATM investments*, June 2003.

Eurocontrol Experiment Center, *the Toolset for Emission Analysis*, Sep 2003.

Eurocontrol, *Guidelines for the Economic Appraisal of EATMP Projects – the Effective Use of Cost-Benefit Studies*, Sep 2000.

Federal Aviation Administration, *Air Carrier Activity Information System (ACAIS), Primary and Non-primary Commercial Service Airports (by Rank Order),* CY 2006.

Federal Aviation Administration (FAA), *Economic Values for FAA Investment and Regulatory Decisions, a Guide*, Dec 2004.

Federal Aviation Administration (FAA), *GPS/WAAS Approaches*, FAA, Department of Transportation, http://gps.faa.gov/CapHill/approachGPS-WAAS-text.htm

Federal Aviation Administration (FAA), *Landing Facility Database*, FAA, U.S. Department of Transportation, March 2004.

Federal Aviation Administration (FAA), GPS Product Team, *Wide Area Augmentation System*, FAA, Department of Transportation, 2003 FRP User Conference, May 19, 2003

Federal Aviation Administration (FAA), Office of Aviation Policy and Plans, *Treatment of Values of Travel Time in Economic Analysis*, APO Bulletin APO-03-1, Mar 2003.

Federal Aviation Administration (FAA), Order 5200.9, *Financial Feasibility and Equivlancy of Runway Safety Area Improvement and Engineered Material Arresting Systems*, FAA, U.S. Department of Transportation, April 8, 1999

Federal Aviation Administration (FAA), Order 8260.48, *Area Navigation (RNAV) Approach Construction Criteria*, FAA, U.S. Department of Transportation, April 8, 1999.

Federal Aviation Administration (FAA), Order 8260.38A, *Civil Utilization of Global Positioning System*, FAA, U.S. Department of Transportation, April 5, 1995

Federal Aviation Administration (FAA), National Aeronautical Charting Office, *United States Government Flight Information Publication U.S. Terminal Procedures*, FAA, U.S. Department of Transportation, July 10, 2003.

Federal Aviation Administration (FAA), Navigation Services, *Lighting System Group – MALSR*, ,
http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/lsg/malsr/, accesed 2008.

Federal Aviation Administration (FAA), Navigation Services, *Lighting System Group – PAPI*,
http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/lsg/papi/, accessed 2008.

Federal Aviation Administration (FAA), *FAA Wide Area Augmentation System – Factsheet*, FAA, Department of Transportation, http://gps.faa.gov/Library/waas-f.htm

Federal Aviation Administration (FAA), GPS Product Team, *Wide Area Augmentation System*, FAA, Department of Transportation, 2003 FRP User Conference, May 19, 2003.

Federal Aviation Administration (FAA), *Terminal Area Forecast Summery Fiscal Years 2005-2025*, FAA-APO-06-1, Washington DC, March 2006.

Federal Aviation Administration (FAA), Operations & Performance Data, *Terminal Area Forecast*, http://www.apo.data.faa.gov/main/taf.asp

Federal Aviation Administration (FAA), Office of Aviation Policy and Plans Statistics and Forecast Branch (APO-110) and GRA Incorporated, *Forecasting Aviation Activity by Airport*, Washington DC, July 2001.

Federal Aviation Administration (FAA), Office of Aviation Policy and Plans, *FAA Airport Benefit-Cost Analysis Guidance*, Dec 1999.

Federal Aviation Administration (FAA), *General Aviation and Air Taxi Activity Survey*, Washington DC, 2002.

Federal Aviation Administration (FAA) National Aeronautical Charting Office, *U.S. Government Flight Information Publication*, U.S. Terminal Procedure, 2003.

Federal Aviation Administration, *Aviation System Performance Metrics (ASPM)*, http://www.apo.data.faa.gov/faamatsall.HTM
FAA, *National Airspace Resource Investment Model (NARIM)*, http://www.faa.gov/ASD/ia-or/NARIM.htm, Jan 2004, accessed May 2006.

Federal Aviation Administration Simulation and Analysis Group, *National Airspace Resource Investment Model (NARIM)*, http://www.tc.faa.gov/acb300/330_narim.asp

Federal Aviation Administration Office of Environment and Energy, ATAC Corporation, Volpe National Transportation Systems Center Acoustics Facility, *The Integrated Noise Model 6.0 User's Guide*, September 1999.

Federal Aviation Administration Office of Environment and Energy, ATAC Corporation, Volpe National Transportation Systems Center Acoustics Facility, *The Integrated Noise Model 6.0 Technical Manual*, January 2002.

Federal Highway Administration (FHWA), *Surface Transportation Efficiency Analysis Model (STEAM)*, March 8th, 2005, http://www.fhwa.dot.gov/steam/index.htm

Feitelson, E.I., Hurd, R.E. and Mudge, R.R., *The Impact of Airport Noise on Willingness to Pay for Residences*, Transportation Research, 1D P1-14, 1996.

Fowkes, A.S., *The UK Department of Transport Value of Time Project*, International Journal of Transport Economics, 13(2), P. 197-207.

FAA Advisory Circular 150/5340-1H, *Standards for Airport Markings*, FAA, August, 1999.

FAA Advisory Circular 150/5300-13, *Airport Design*, FAA, September 1989.

Gellman Research Associates, Inc., *The Value of Time in Benefit-Cost Analysis of FAA Investment Decisions*, Volume I, Report to FAA, undated.

Gellman Research Associates, Inc., *A Report on the Value of Air Traveler's Time*, March 1988).

Ginger Shepherd, *Tulsa International Airport Runway Closed for Taxiway Improvements*, Journal Record, The (Oklahoma City), Oct 23, 2006.

GRA, Incorporated, Aviation Specialists Group Inc., Data Base Products. Prepared or FAA Office of Aviation Policy and Plans, *Economic Value for FAA Investment and Regulatory Decisions, A Guide*, Contract No DTFA 01-02-C00200, Draft Final Report, Dec 2004.

Gronau R., *The Value of Time in Passenger Transportation: The Demand for Air Travel*, Pp. xiv, 74, New York, National Bureau of Economic Research, 1970.

Gunn, H. and Rohr, C., Cheung, Y.H.F. and Kleijn, H.J., *Value of Dutch Travel Time Savings*, 1997 Proceedings of the 8th World Conference on Transport Research, pp.513-526. Volume 3 Transport Modeling /Assessment. Edited by Meersman, H., Van de Voorde, E. and Winkelmans, W., Pergamon, Amsterdam, 1999.

Hector International Airport, http://www.fargoairport.com/, accessed 2008.

Hobeika, A., *Transportation and Land Use*, class notes, Virginia Tech, Fall 2003.

Holland, J.H., *Hidden Order: How Adaptation Builds Complexity*, Addison-Wisely, Reading, MA, 1995.

Hunter, G., et, al. *Evaluation of Future National Airspace System Architecture*, AIAA Modeling and Simulation Technologies Conference and Exhibit, AIAA 2005-6492, Sensis Corporation and George Mason University, San Francisco, CA, Aug 2005.

Johnson, J., and Santmire, T., Logistics Management Institute, *Aviation Systems Analysis Capacity Air Carrier Investment Model – Cargo*, NASA/CR-1999-208984, Mclean VA, Jan, 1999.

Kostiuk, P. F., Lee, D., and Long, D., Logistics Management Institute, *Closed Loop Forecasting of Air Traffic Demand and Delay*, 3rd USA/Europe Air Traffic Management R&D Seminar, Napoli, 13-16 June 2000.

Lancaster, J.A., Saleem, J.J., Kleiner, B.M., Robinson, G.S. and Casali, J.G., *Preliminary study on the effects of approach angle and lower landing minimum level on pilot performance in a low-fidelity static aircraft simulator*, the 12th Biennial International Symposium on Aviation Psychology, Dayton, Ohio, 2002.

Lee, D. A., et al, National Aeronautics and Space Administration Langley Research Center and Logistics Management Institute, NASA Contractor Report 201682, *Estimating the Effects of the Terminal Area Productivity Program*, NASA Contract NAS2-14361, Hampton, Virginia, April 1997.

Lee S., et al., *Hybrid Agent-Based Simulation for Analyzing the National Airspace System*, Proceedings of the 2001 Winter Simulation Conference, 2001.

Lewe, J.H., *An Integrated Decision-Making Framework for Transportation Architecture: Application to Aviation Systems Design*, Dissertation, Georgia Tech, Apr 2005.

Lewe, J.H., et al., *An Agent-Based Framework for Evaluating Future Transportation Architectures*, the American Institute of Aeronautics and Astronautics (AIAA), AIAA-2003-6769, 2003.

Long, D., et al, *Integrating LMINET with TAAM and SIMMOD* – A Feasibility Study, NASA / CR-2001-210875, June 2001.

Long D., et al, *Modeling Air Traffic Management Technologies with a Queuing Network Model of the National Airspace System*, NASA-CR-1999-208988, Logistics Management Institute, Mclean, VA, Jan 1999.

Lori Lynah, *Savannah Airport Commission Notices for Public Comment*, http://www.savannahairport.com/news.php?key=188, Accessed 2007.

Mackie, P.J., *The Value of Travel Time Savings in Evaluation*, Transportation Research Part A., Vol 37, Issue 18, Page 91, 2001.

Metron Aviation, *Noise Integrated Routing System*, http://www.metronaviation.com/nirs.php, Accessed June 2005.

Miller, T. R., *The Value of Time and the Benefit of Time Savings*, National Public Services Research Institute, May 1996.

MIT AATT Model Project, ACIM: *The ASAC Air Carrier Investment Model*, http://web.mit.edu/aeroastro/www/labs/AATT/reviews/lmiacim.html, Oct 1996, accessed May 2006.

Minnesota Department of Transportation, Office of Aeronautics, *Estimated Airport Improvement Costs*, http://www.dot.state.mn.us/aero/avoffice/pdf/costupdate.pdf

National Oceanic and Atmospheric Administration, *National Virtual Data System (NVDA) Local Climatological Data Publication*, http://nndc.noaa.gov/?http://ols.nndc.noaa.gov/plolstore/plsql/olstore.publist?prodnum=C00128-PUB-S0001&subset=005, Accessed May 2005.

Nebraska Department of Aeronautics, *Instrument Approach Procedure Study*, 1999.

Nelson, J. P., *Meta-Analysis of Airport Noise and Hedonic Property Values: Problems and Prospects*, Journal of Transport Economics and Policy, July 2003.

Nelson, J. P., *Airports and Property Values: A Survey of Recent Evidence*, Journal of Transport Economics and Policy, 14, P37-52, 1980.

Niedringhaus, W.P., The MITRE Corporation, *The Jet:Wise Model of National Airspace System Evolution*, Simulation, Vol 80, Issue 1, p45-58, Jan 2004.

Office of Management and Budget, Circular No. A-94 Revised, *Guidelines and Discount Rates for Benefit-Cost Analysis of Federal Programs*, Oct 1992.

Odoni, A. R. et al, *Modeling Research under NASA/AATT Final Report, Existing and Required Modeling Capability for Evaluating ATM Systems and Concepts*, MIT, Boston, March 1997.

Philip Greenspun, *Garmin GNS 530/430 GPS Nav/Com*, http://philip.greenspun.com/flying/garmin-gps, accessed 2008.

Pittsfield Municipal Airport, *Municipal Airport Project*, Jul. 2002.

Pursula, M. and Kurri, J., *Value of Time Research in Finland*, presented at PTRC International Conference on the Value of Time.

Ramjerdi, F., Rand, L. and Salensminde, K., *the Norwegian Value of Time Study: Some Preliminary Results,* Institute of Transport Economics, Oslo, Norway, 1997.

Redding Municipal Airport, *Bid Tabulation for Taxiways A & B Reconstruction*, Aug 2007.

Ruley, J.D., *Understanding WAAS*, Plane and Pilot, Vol 40. No. 8, August 2004.

Sarasota Avionics International, *Garmin GNS-530 TWAS*, http://www.sarasotaavionics.com/product.aspx?id=523, accessed 2008.

Schaefer, L.A., et al., *Decision Support for Advanced Aviation Concepts*, Proceedings of the 2002 Winter Simulation Conference, 2002.

Schultz, T.J., Synthesis of Social Surveys on Noise Annoyance, *Journal of the Acoustical Society of America* 64(2) pp377-405, 1978.

Society of Automotive Engineers, Committee A-21, Aircraft Noise, *Procedure for the Computation of Airplane Noise in the Vicinity of Airports*, Aerospace Information Report No. 1845, Warrendale, PA: Society of Automotive Engineers Inc., March 1986

Society of Automotive Engineers, Committee A-21, Aircraft Noise, *Prediction Method for Lateral Attenuation of Airplane Noise During Takeoff and Landing*, Aerospace Information Report No. 1751, Warrendale, PA: Society of Automotive Engineers Inc., March 1986.

Society of Automotive Engineers, *Standard Values at Atmospheric Absorption as a Function of Temperature and Humidity for Use in Evaluating Aircraft Flyover Noise*, Aerospace Recommended Practice No 866A, March 15, 1975.

Texas Transportation Institute, Texas A&M University System, Office of Highway Information Management of FHWA, *Travel Time Data Collection Handbook*, Report FHWA-PL-98-035March 1998, http://www.fhwa.dot.gov/ohim/start.pdf

The Broward County Aviation Department, *North Perry Airport Public Meeting*, Jun 2006.

The Port Authourity of NY&NJ, *Airport Traffic Report 2006*, 2006.

Trani, A.A., H. Baik, H. Swingle, S. Ashiabor, N. Hinze, A. Seshadri, K. Murthy and Y. Xu, *SATS Transportation System Analysis Model*, Blacksburg, Virginia, May 2005.

Trani, A.A., *National Transportation Systems Analysis for the SATS Program*, presented at the NCAM LLM/SSP Meeting, November 2, 2004.

Trani, A.A., *Analysis of Air Transportation Systems, class notes*, Virginia Polytechnic Institute and State University, June 2003 .

Trani, A. A., Baik, H., Swingle, H., and C. Ackley, *Small Aircraft Transportation Noise Impact Study*, Report to NASA Langley Center, October 2000.

Trani, A.A. *Strategies to Improve Availability of Airports using LLM Technologies*, presented at NCAM Strategic Meeting, February 18, 2004.

U.S. Census Bureau, *Census 2000 population file*, http://www2.census.gov/census_2000/datasets/redistricting_file--pl_94-171/

U.S. Census Bureau, *Census TIGER 2000/Line file*, http://www.esri.com/data/download/census2000_tigerline/

Ventana Systems Inc, *FAA NAS Strategy Simulator*, unpublished paper, FAA NAS Strategy Simulator version 0.152.

Wang, M., Santini, D., Monetary Values of Air Pollution Emissions in Various US Areas, Transportation Research Board Paper no. 951046, Washington DC, 1995.

Wardman, M., *Inter-Temporal Variation in the Value of Time*, ITS Leeds, Working Paper 566, December. 2001.

Wikipedia, *External Cost*, http://en.wikipedia.org/wiki/External_cost, accessed May 2006.

Wikipedia, Wide Area Augmentation System, http://en.wikipedia.org/wiki/Wide_Area_Augmentation_System#_note-AOPA_welcomes_WAAS, accessed Feb 2008.

Williams, H., On *the Formation of Travel Demand Models and Economic Evaluation Measures of User Benefit*, University of Leeds, Institute of Transportation Studies, 1976.

Wingrove, E. R. III et al, *The ASAC Air Carrier Investment Model (Second Generation)*, NASA Contractor Report 201678, NASA Langley Research Center, Hampton, VA, April 1997.

Wingrove, E. R. III et al, *The ASAC Air Carrier Investment Model (Third Generation)*, NASA Contractor Report NASA-CR-1998-207656, NASA Langley Research Center, Hampton, VA, April 1998.

Wit, R.C.N., et. al, *Meeting External Costs in the Aviation Industry*, Report to the Commission of Integrated Transport (CfIT), Aug 2003.

Woodbridge, M., Jennings, N.R., *Intelligent agents: Theory and Practice*, The Knowledge Engineering Review, Vol 10(2), P115-152, 1995.

Y. Xu, A. A. Trani, Baik, H.J., *Preliminary Assessment of the Lower Landing Minima Capabilities in the SATS Program*, Transportation Research Record, Issue 1915, 2005.

Y. Xu, A. A. Trani, *A Preliminary Evaluation of Potential Cargo Demand for Very Light Jets*, submitted to the 3rd International Conference on Research in Air Transportation – ICRAT 2008.

Y. Xu, Baik, H.J., A. A. Trani, *A Preliminary Assessment of Airport Noise and Emission Impacts Induced by Small Aircraft Transportation System Operations,* 6th AIAA Aviation Technology, Integration, and Operation Conference (ATIO), Wichita, KS, 2006.

Zelinski, S., *Validating the Airspace Concept Evaluating System Using Real World Data*, AIAA Modeling and Simulation Technology Conference, San Francisco, CA, Aug 2005.

Appendix

Appendix A: DataComm Simulation Outputs

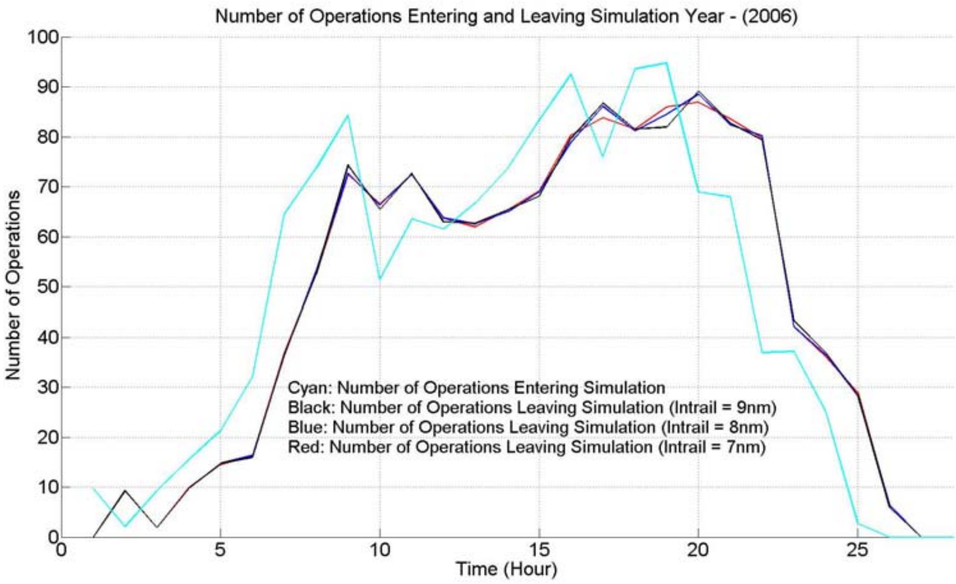

Figure 9-1: Number of Operations Entering and Leaving Simulation – EWR 2006.

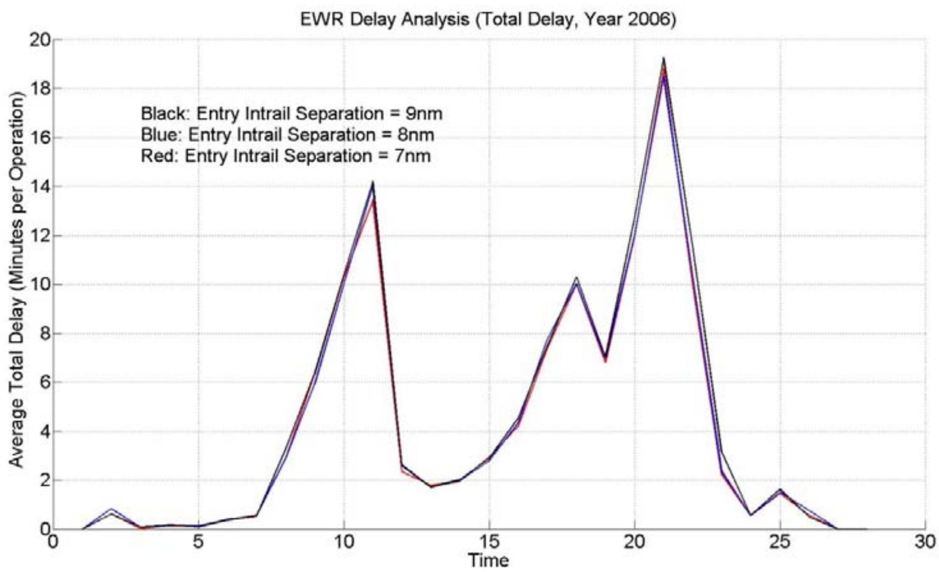

Figure 9-2: Average Total Delay – EWR, Year 2006.

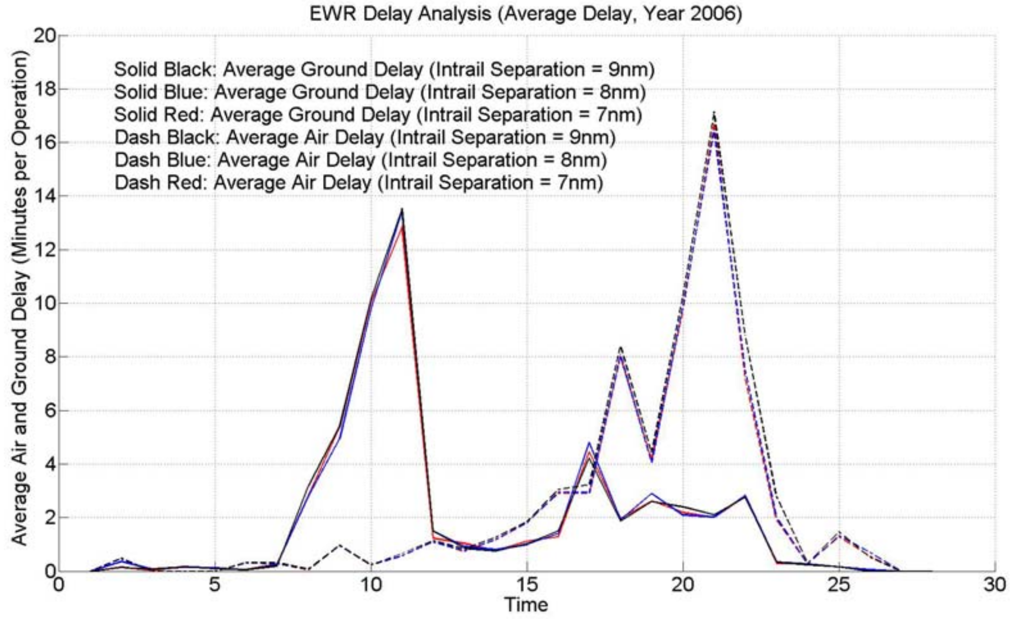

Figure 9-3: Average Air and Ground Delay – EWR, Year 2006.

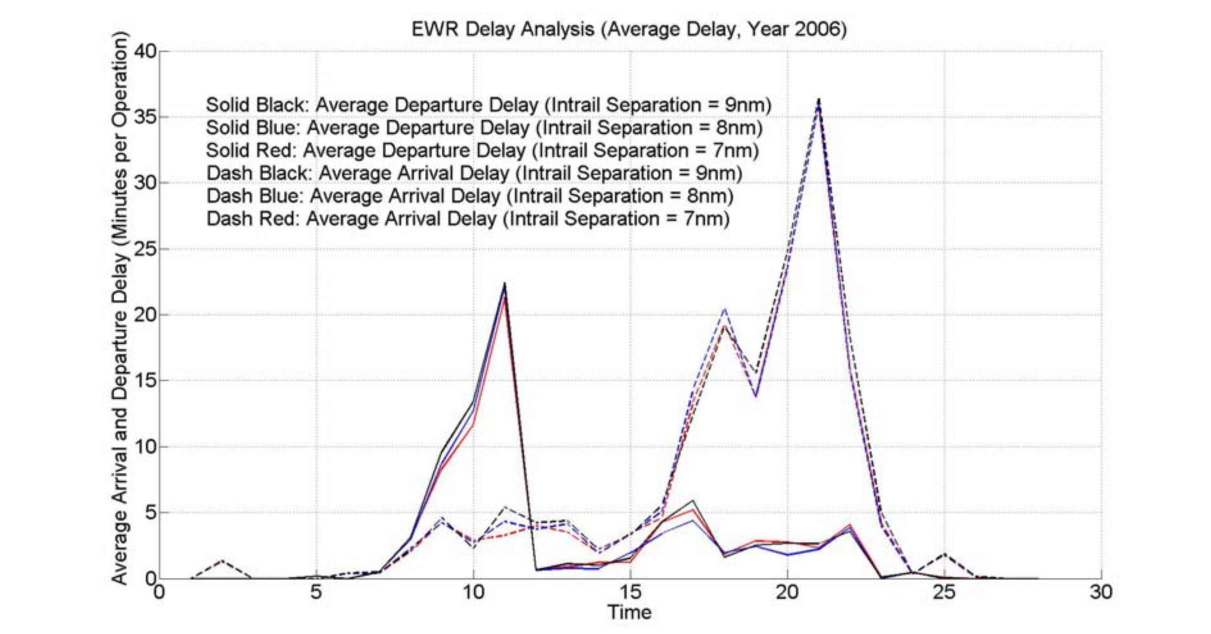

Figure 9-4: Average Arrival and Departure Delay – EWR, Year 2006.

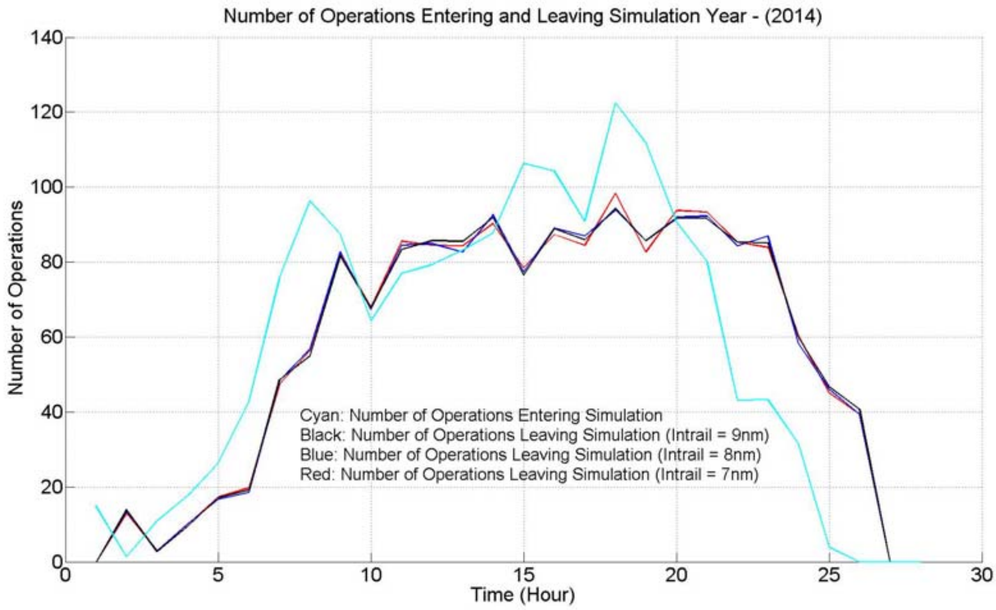

Figure 9-5: Number of Operations Entering and Leaving Simulation – EWR 2014.

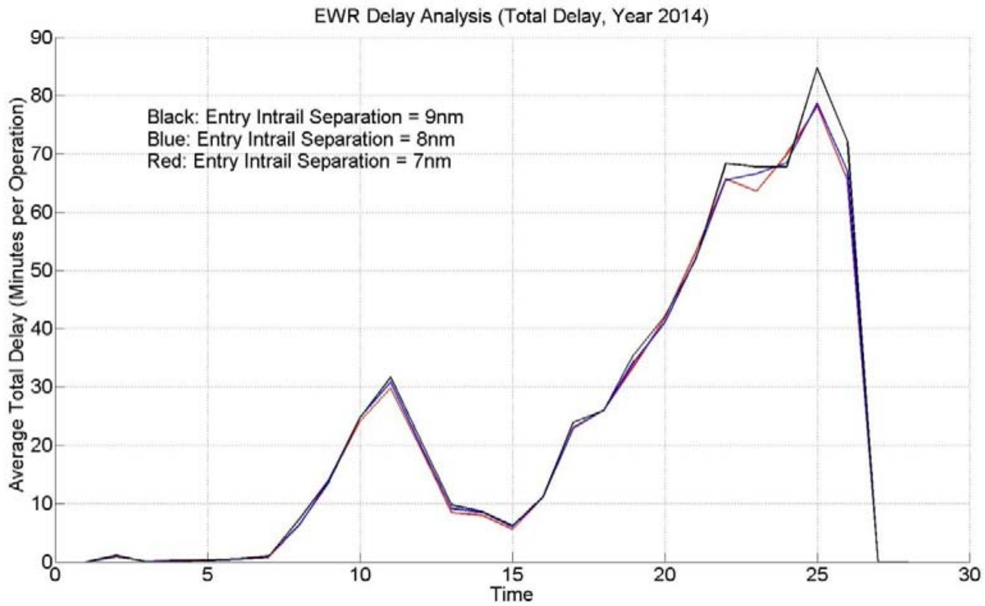

Figure 9-6: Average Total Delay – EWR, Year 2014.

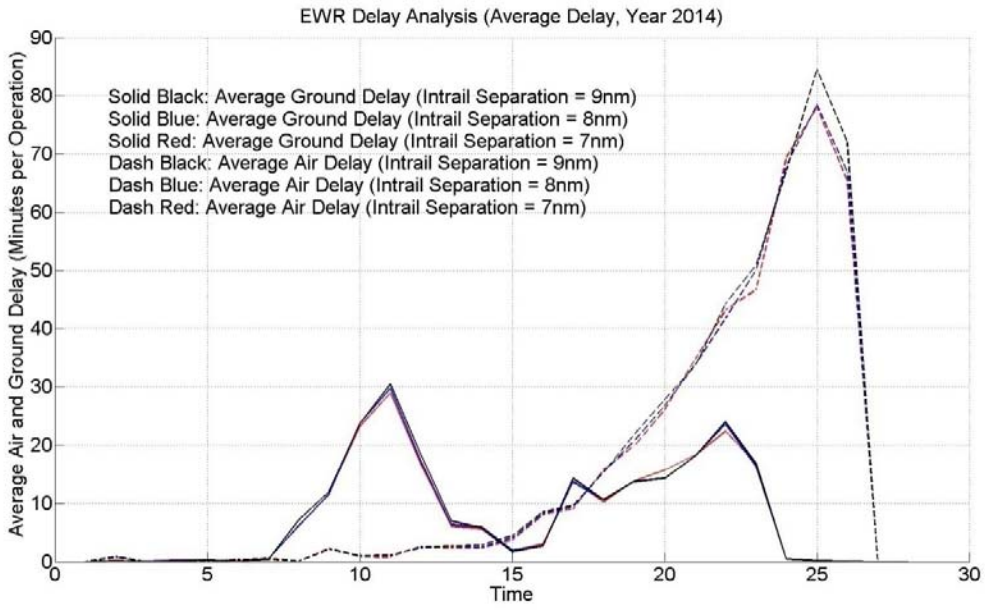

Figure 9-7: Average Air and Ground Delay – EWR, Year 2014.

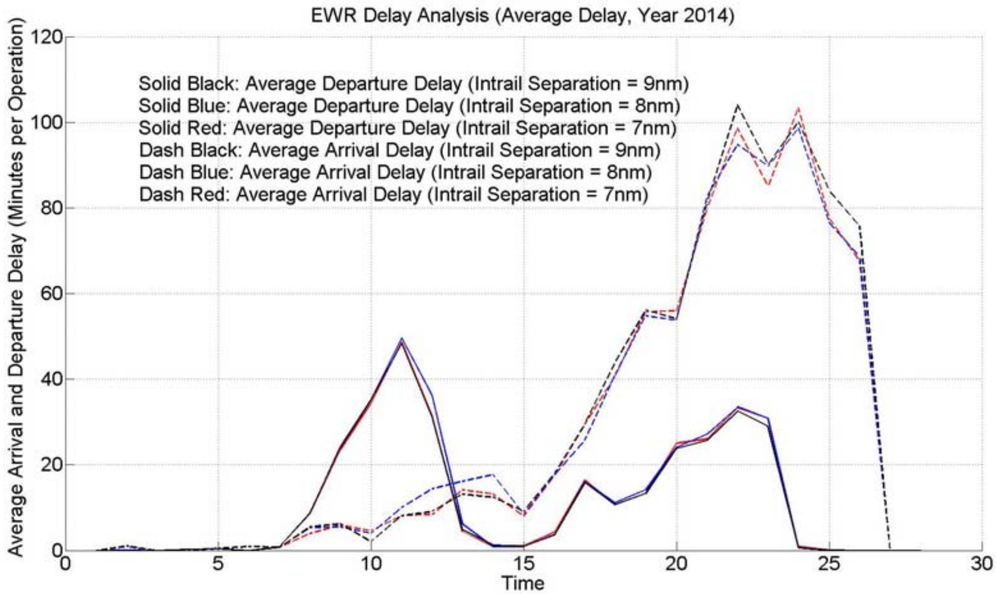

Figure 9-8: Average Arrival and Departure Delay – EWR, Year 2014.

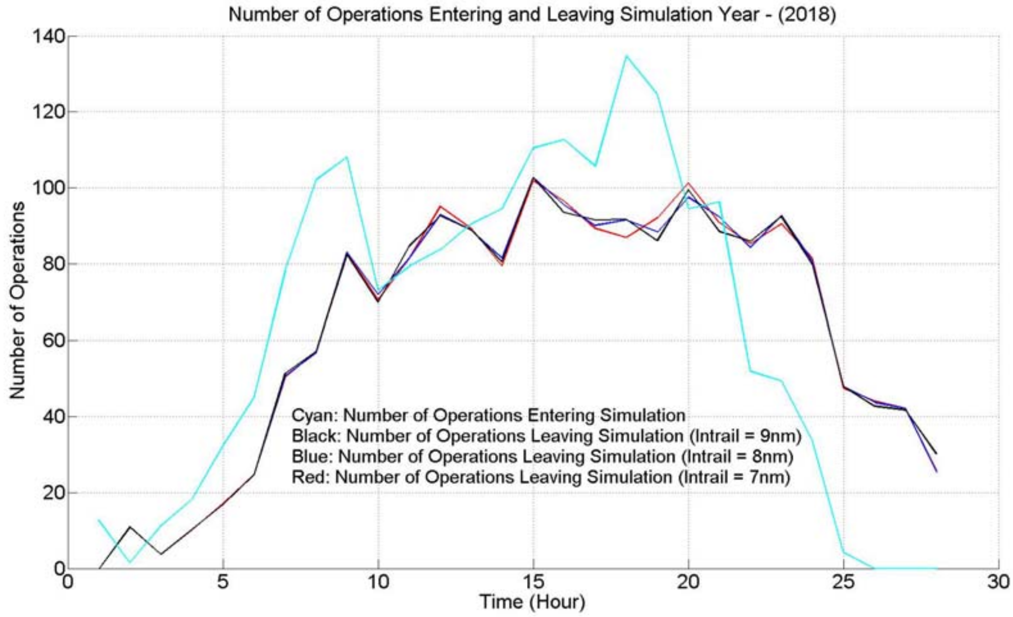

Figure 9-9: Number of Operations Entering and Leaving Simulation – EWR 2018.

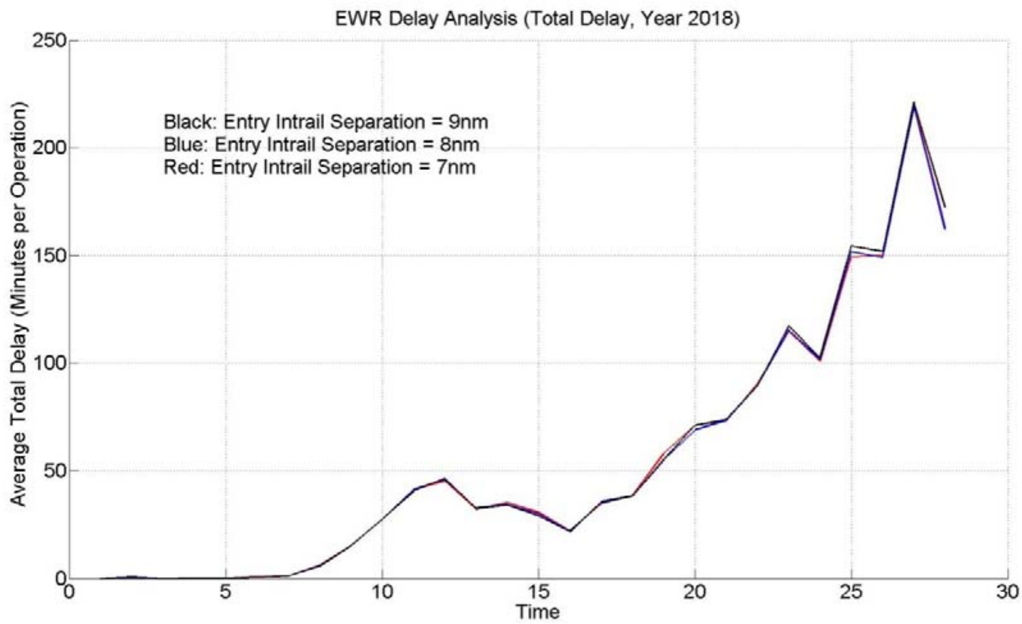

Figure 9-10: Average Total Delay – EWR, Year 2018.

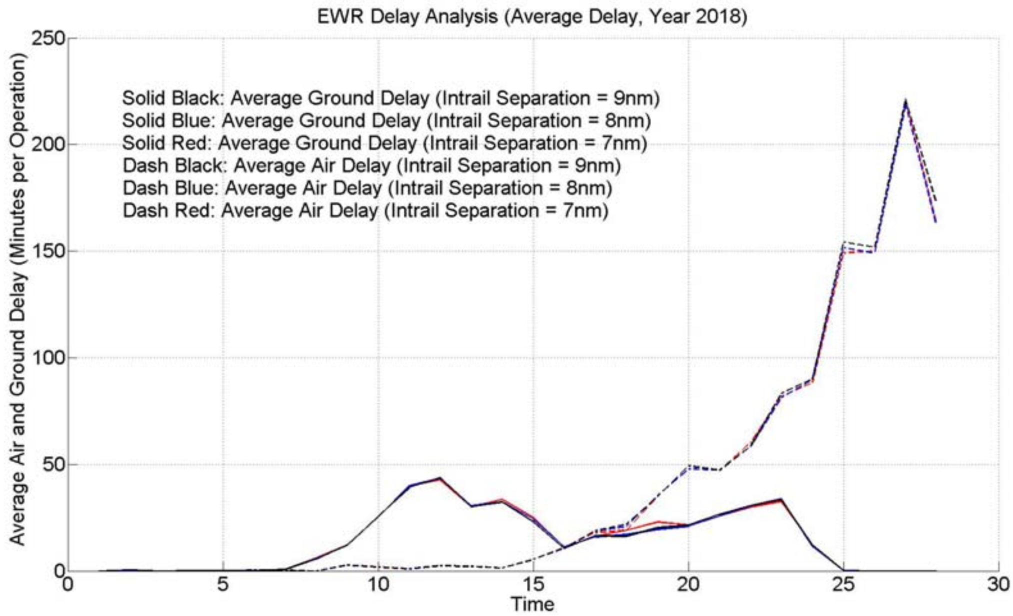

Figure 9-11: Average Air and Ground Delay – EWR, Year 2018.

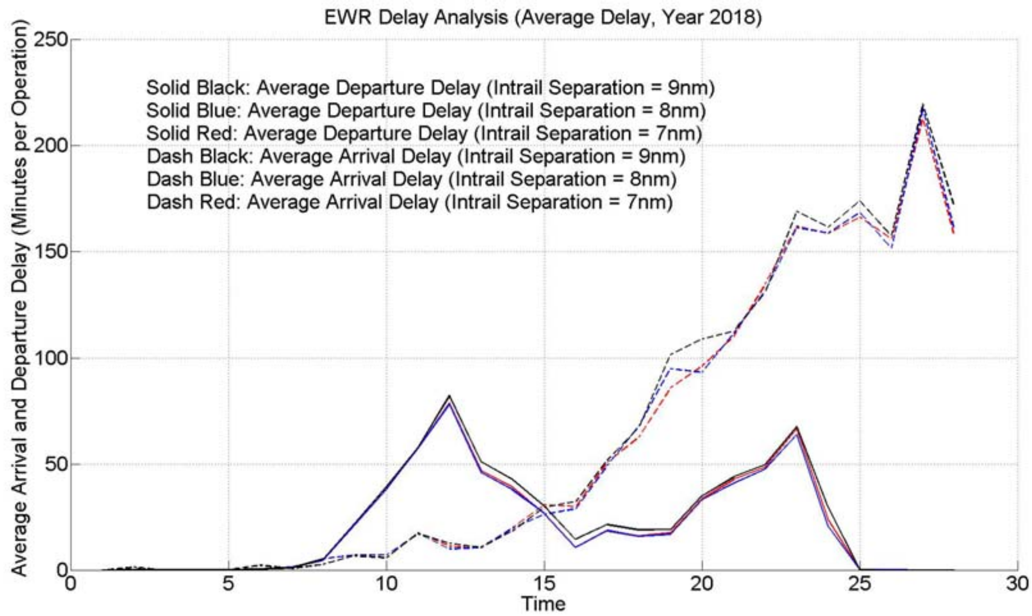

Figure 9-12: Average Arrival and Departure Delay – EWR, Year 2018.

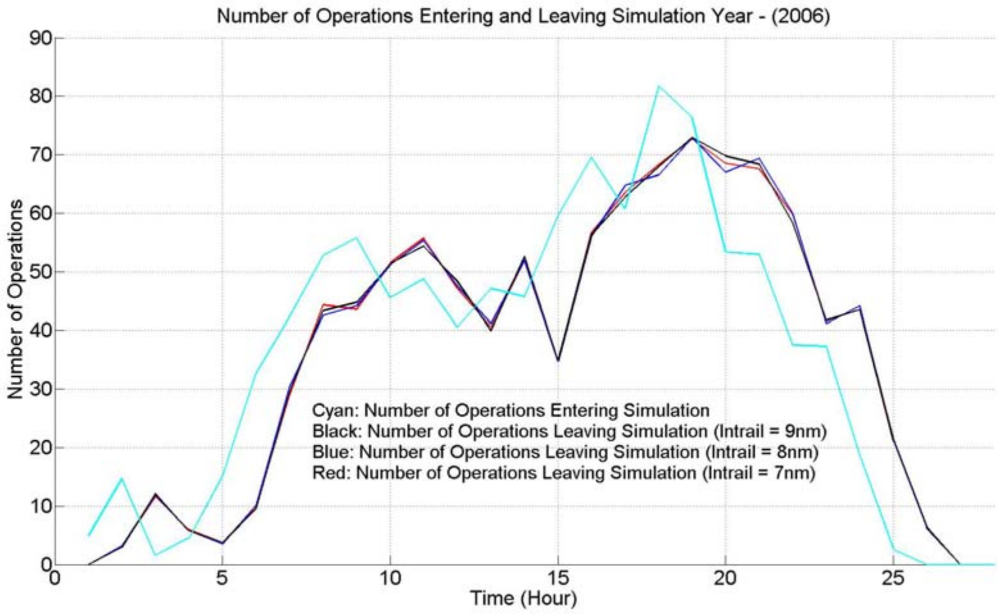

Figure 9-13: Number of Operations Entering and Leaving Simulation – JFK 2006.

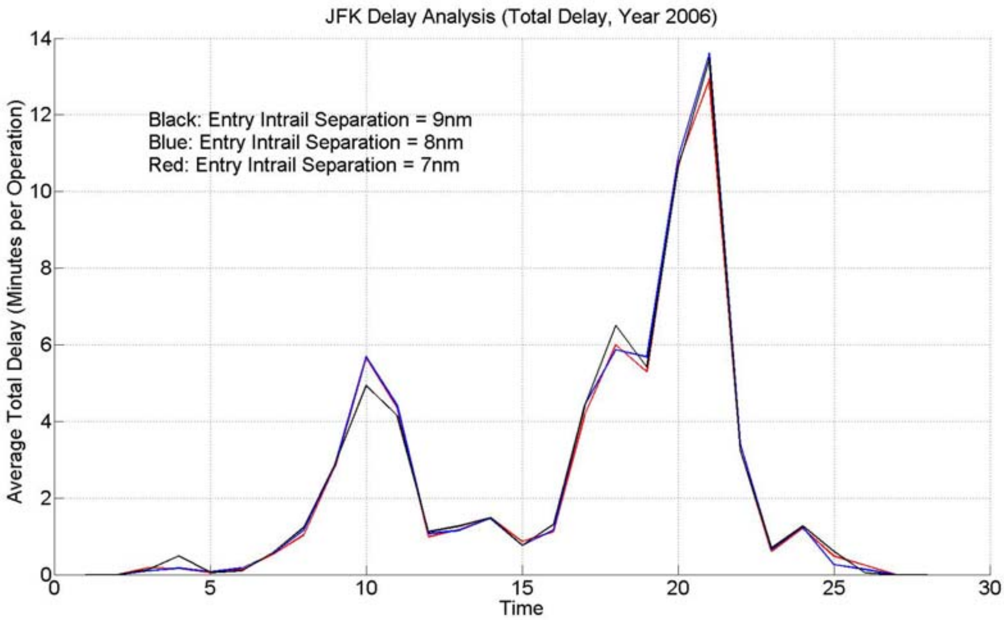

Figure 9-14: Average Total Delay – JFK, Year 2006.

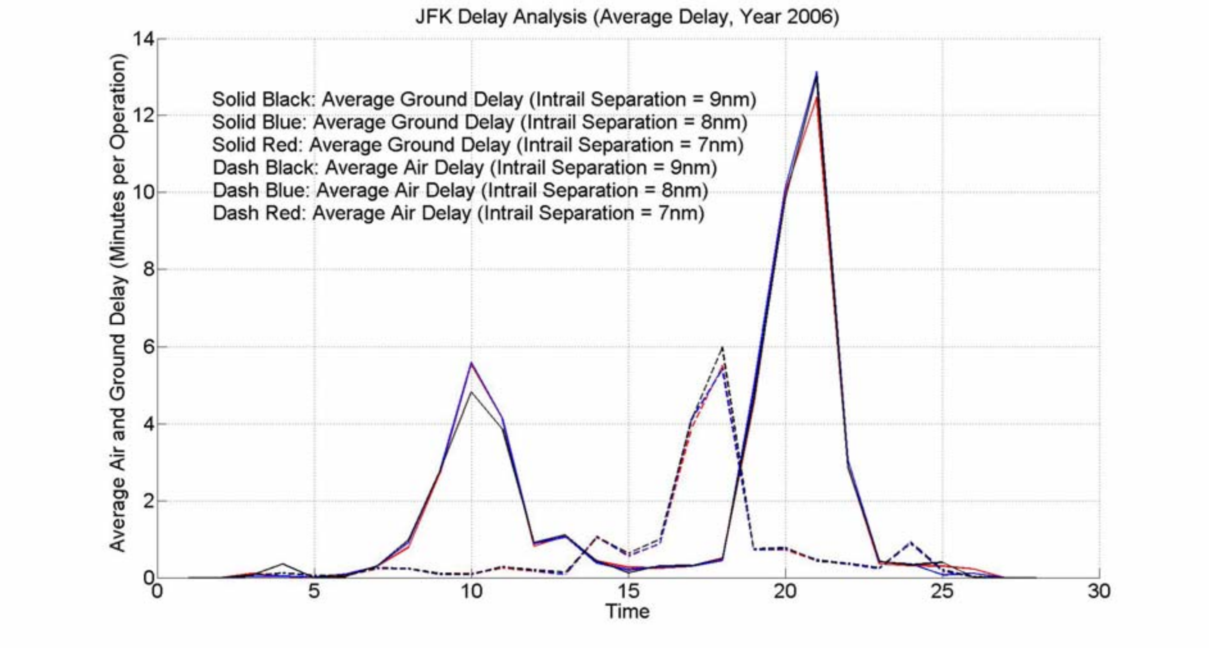

Figure 9-15: Average Air and Ground Delay – JFK, Year 2006.

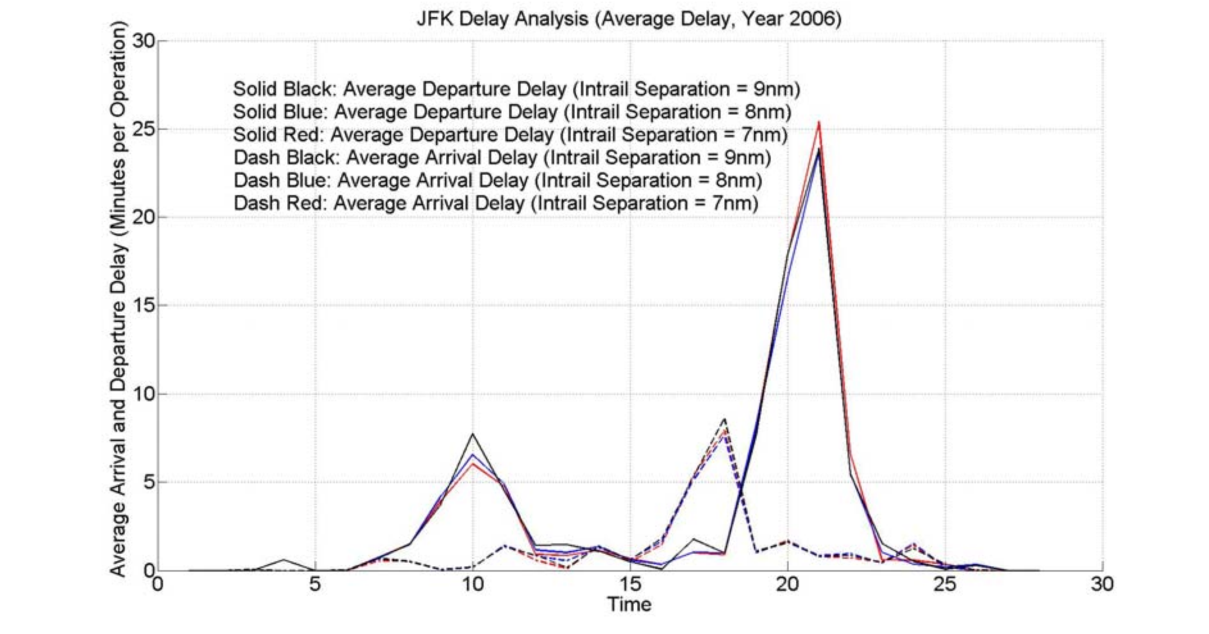

Figure 9-16: Average Arrival and Departure Delay – JFK, Year 2006.

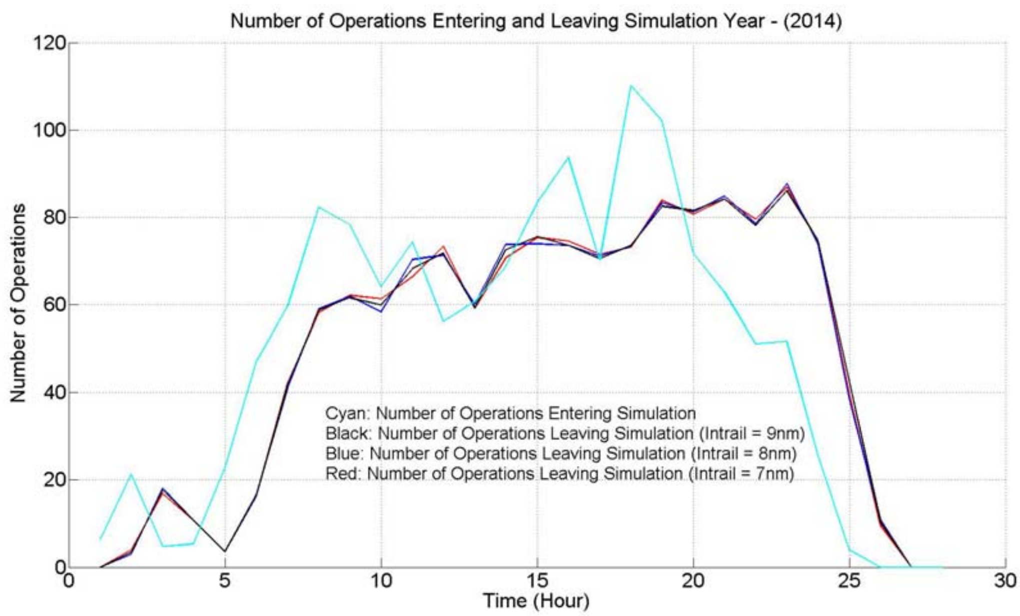

Figure 9-17: Number of Operations Entering and Leaving Simulation – JFK 2014.

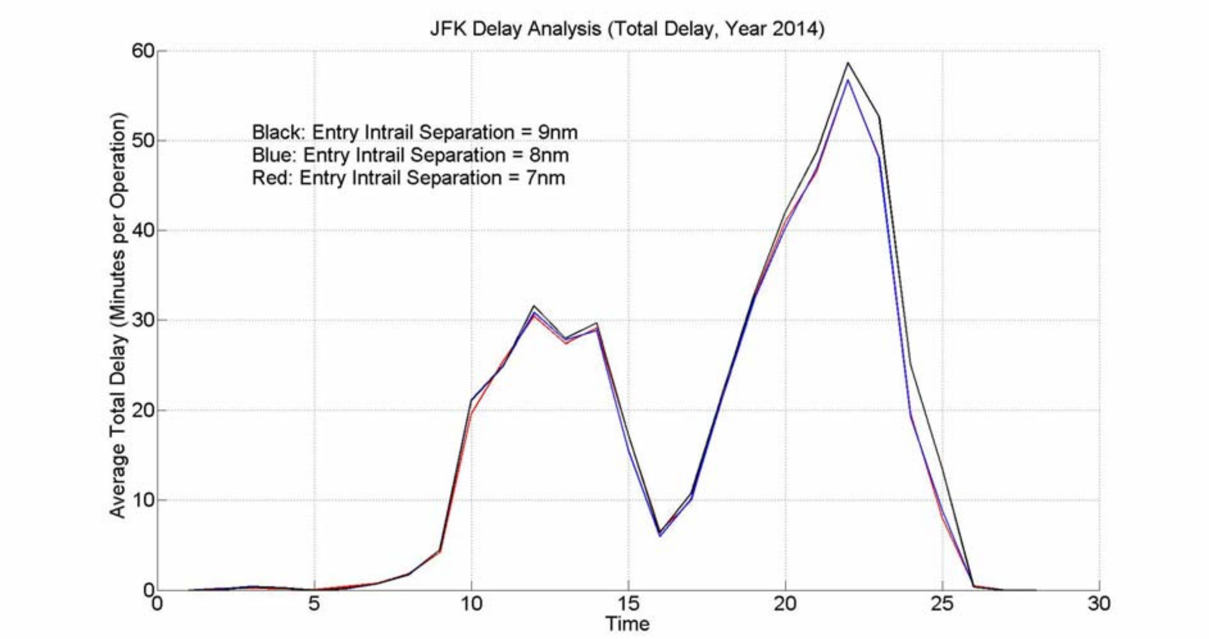

Figure 9-18: Average Total Delay – JFK, Year 2014.

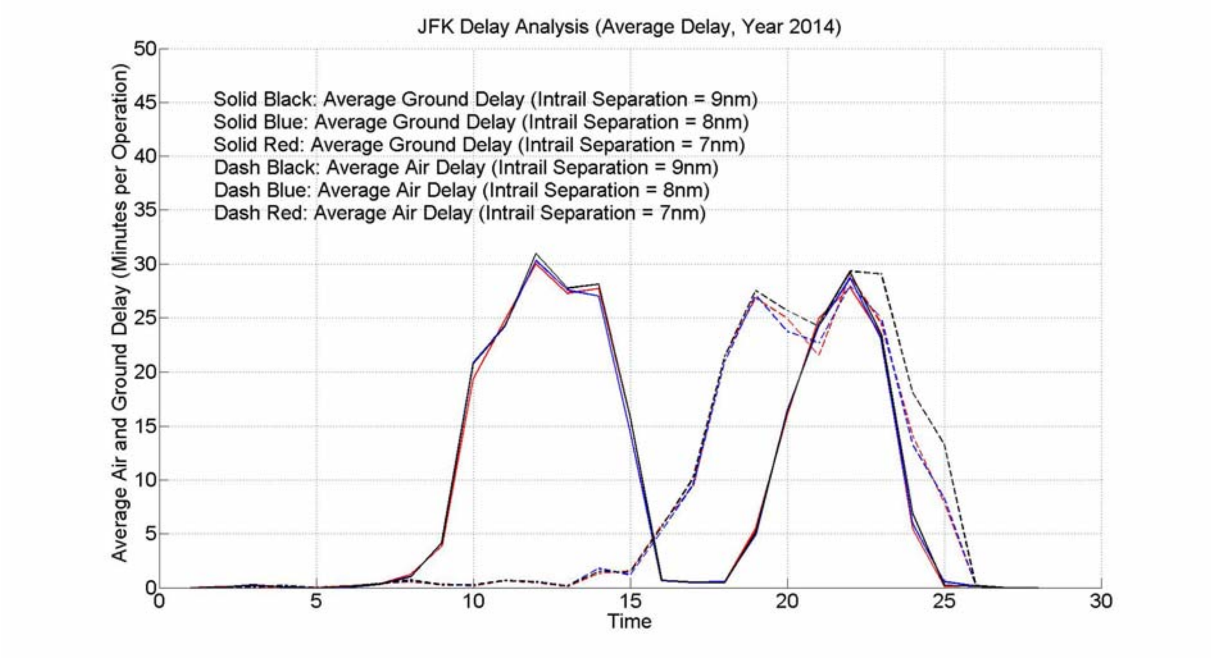

Figure 9-19: Average Air and Ground Delay – JFK, Year 2014.

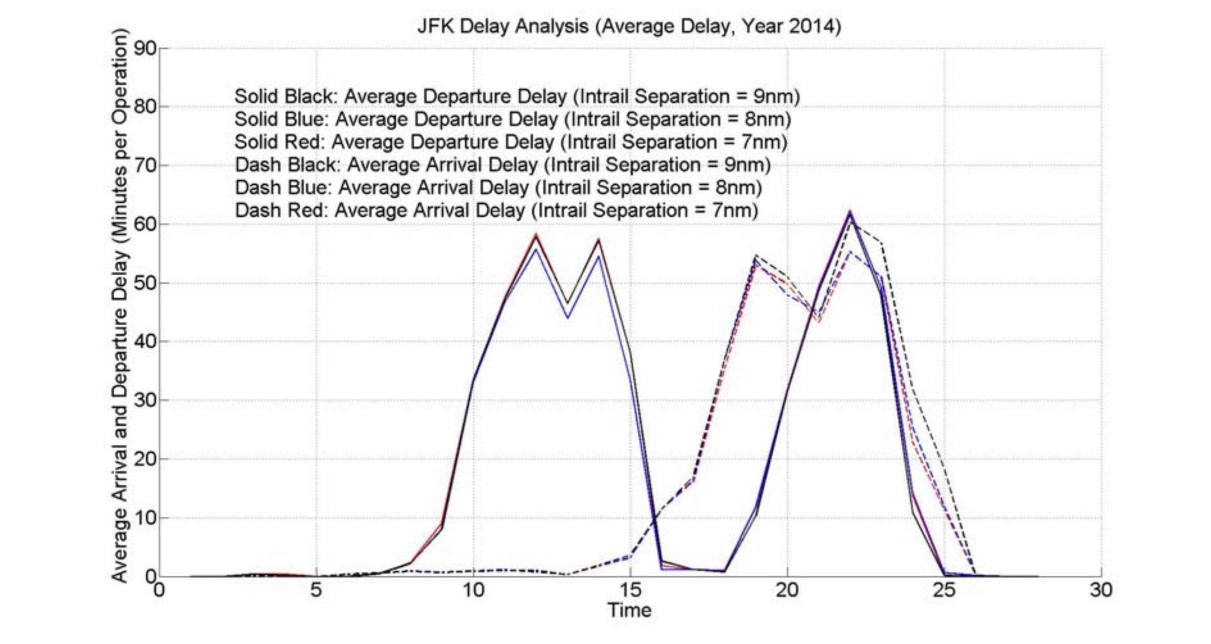

Figure 9-20: Average Arrival and Departure Delay – JFK, Year 2014.

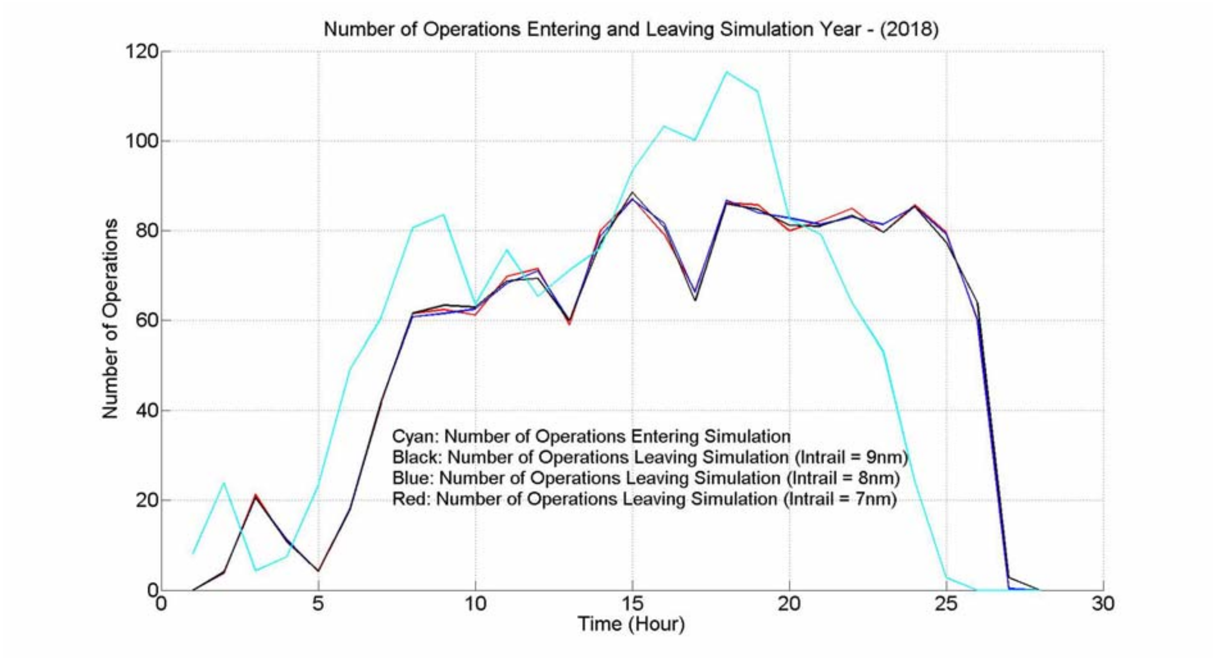

Figure 9-21: Number of Operations Entering and Leaving Simulation – JFK 2018.

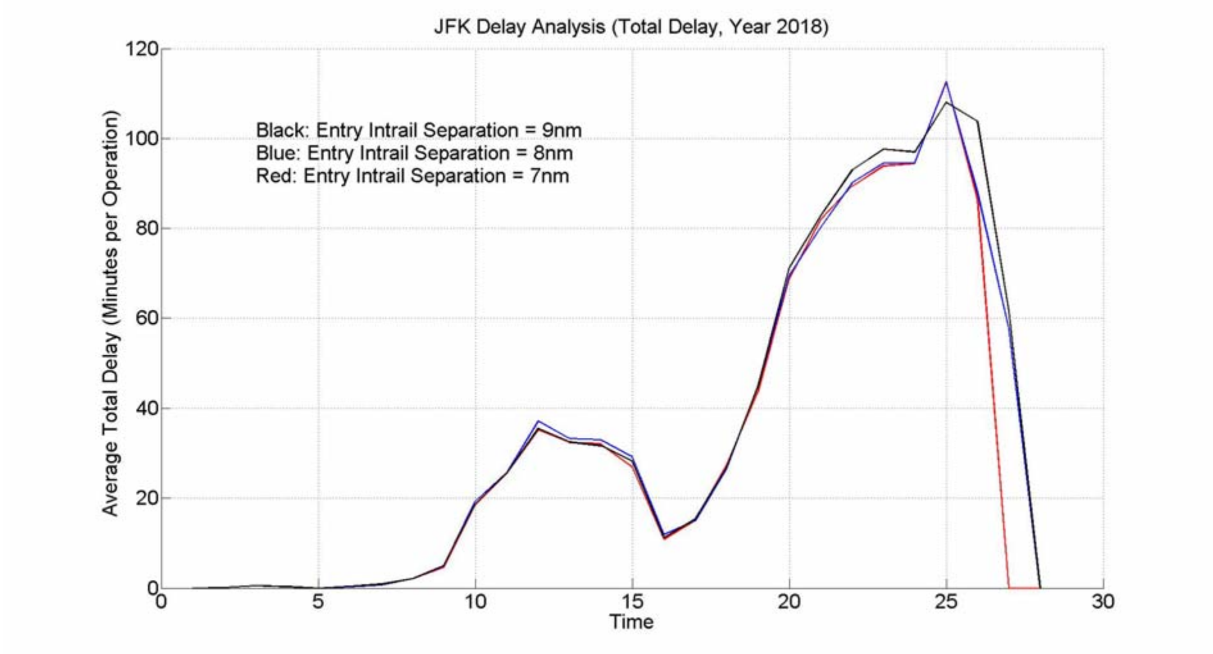

Figure 9-22: Average Total Delay – JFK, Year 2018.

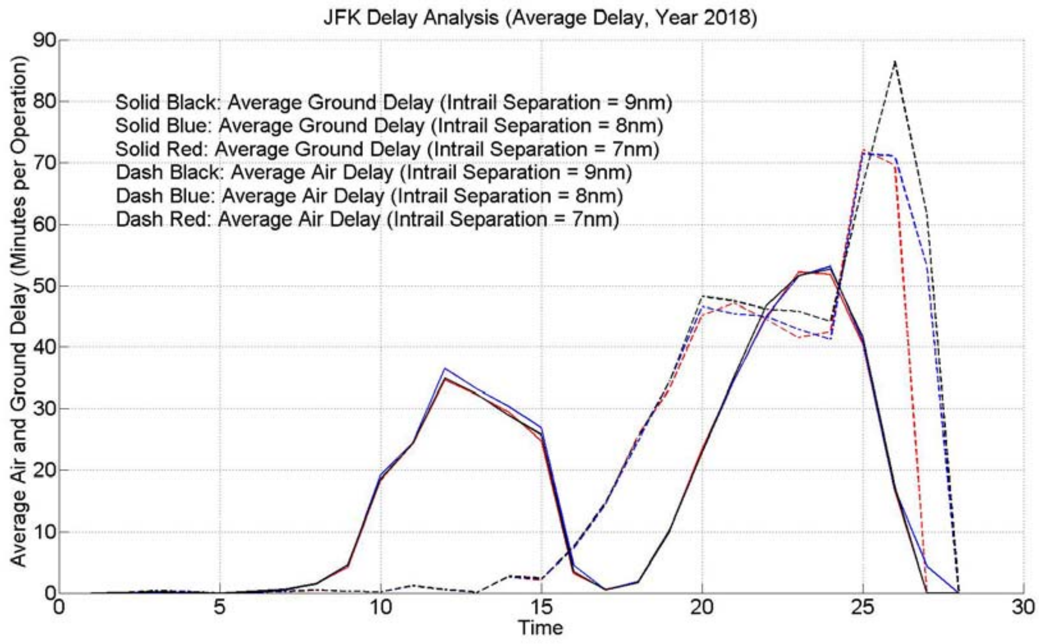

Figure 9-23: Average Air and Ground Delay – JFK, Year 2018.

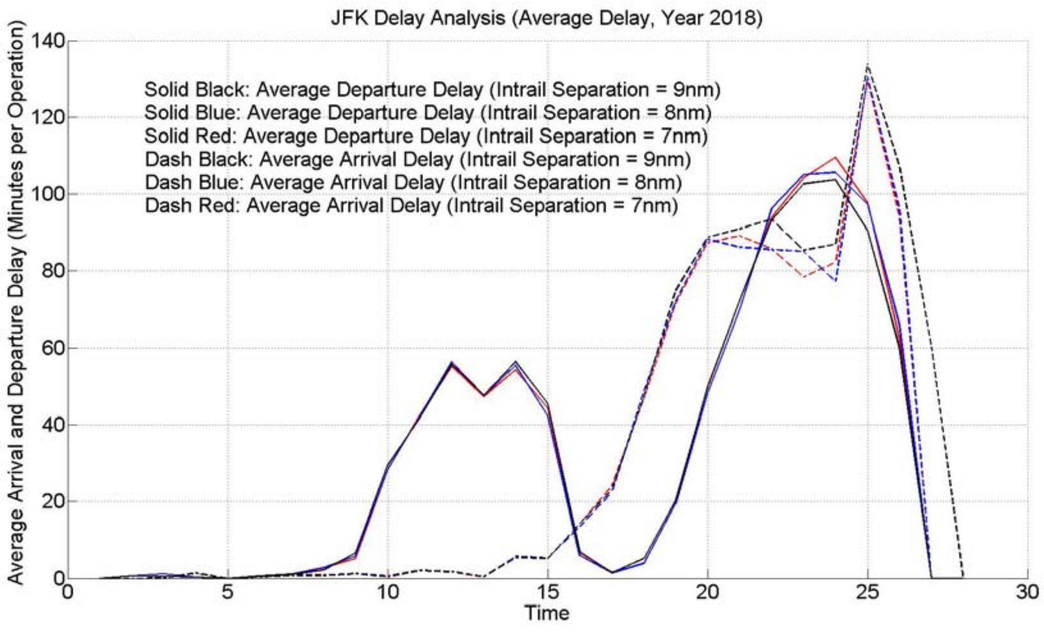

Figure 9-24: Average Arrival and Departure Delay – JFK, Year 2018.

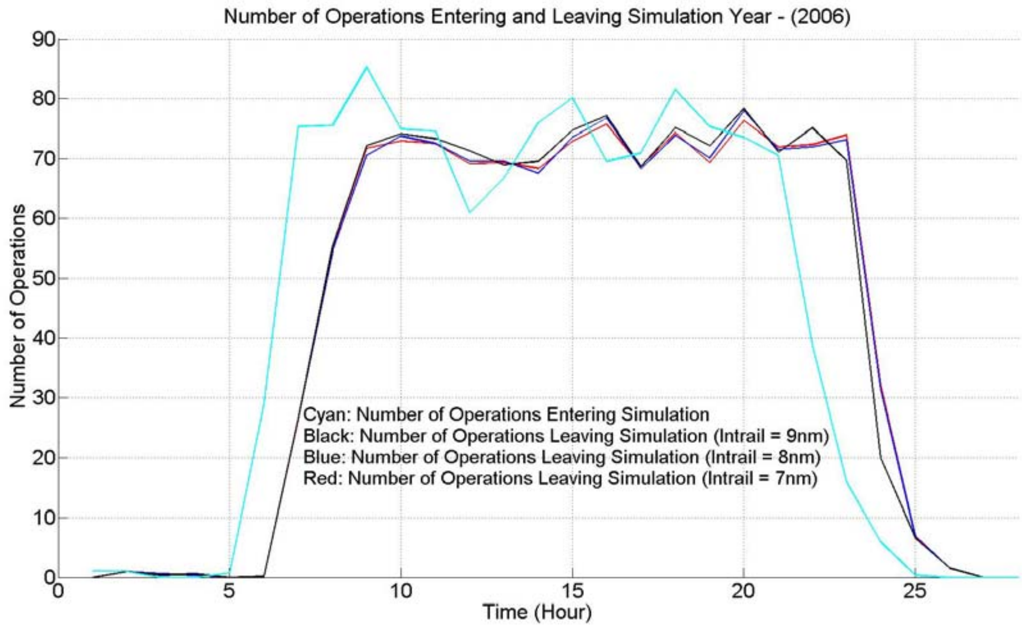

Figure 9-25: Number of Operations Entering and Leaving Simulation – LGA 2006.

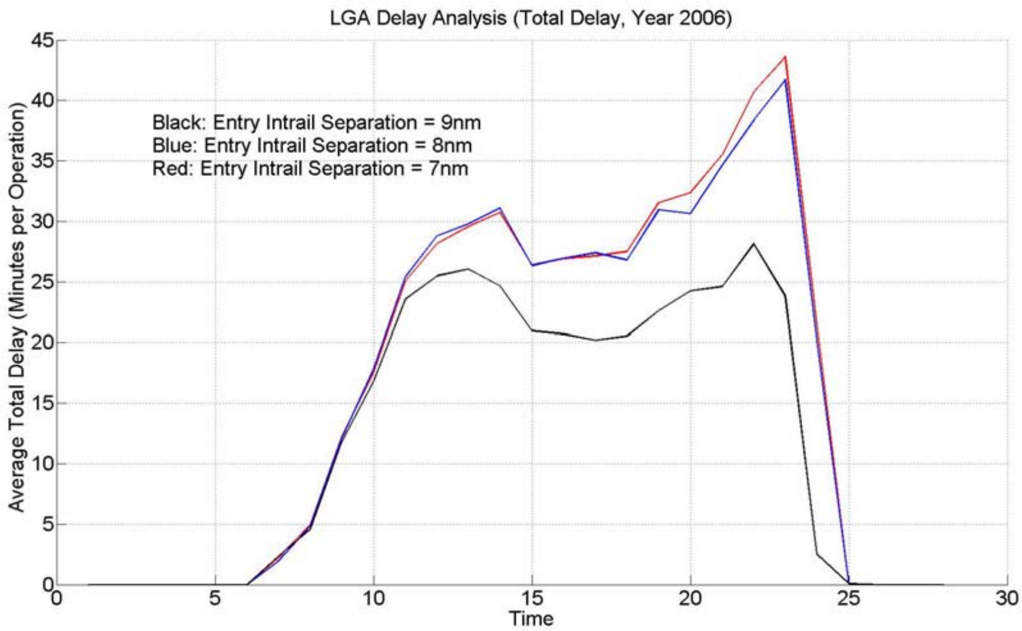

Figure 9-26: Average Total Delay – LGA, Year 2006.

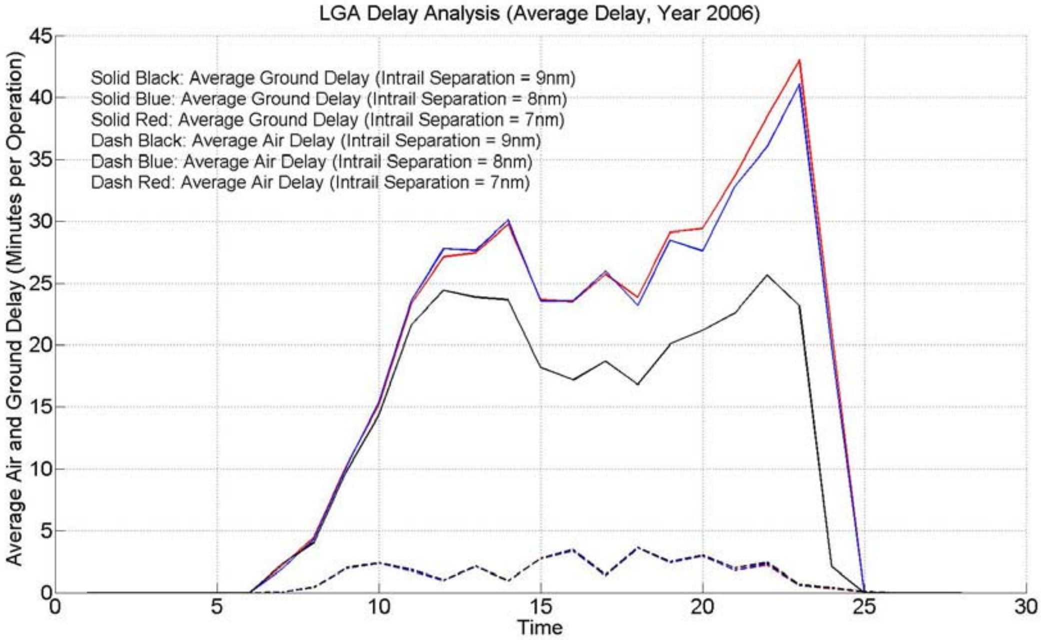

Figure 9-27: Average Air and Ground Delay – LGA, Year 2006.

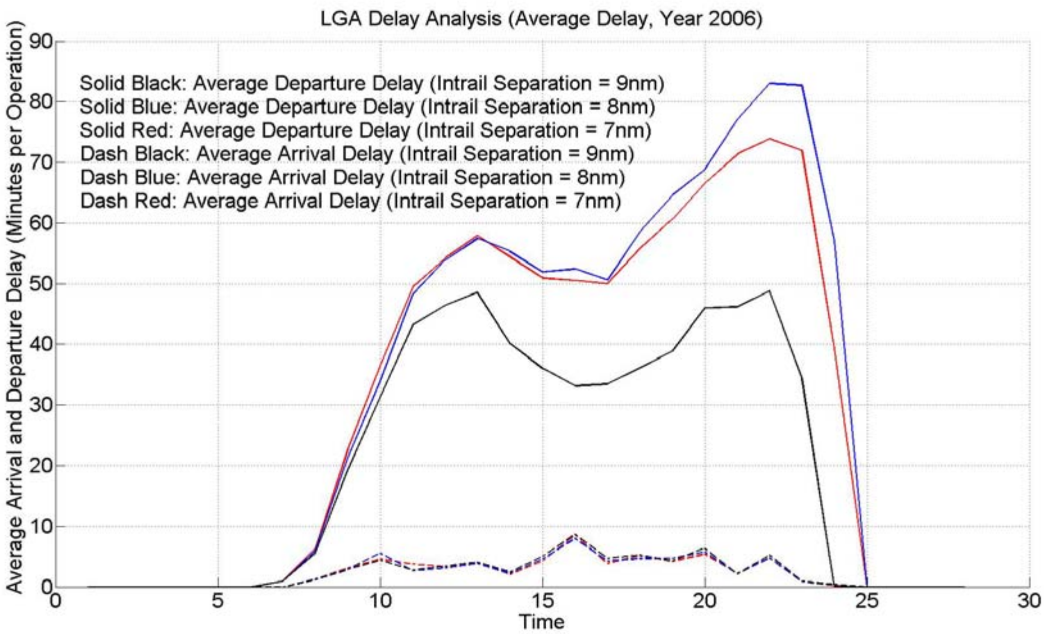

Figure 9-28: Average Arrival and Departure Delay – LGA, Year 2006.

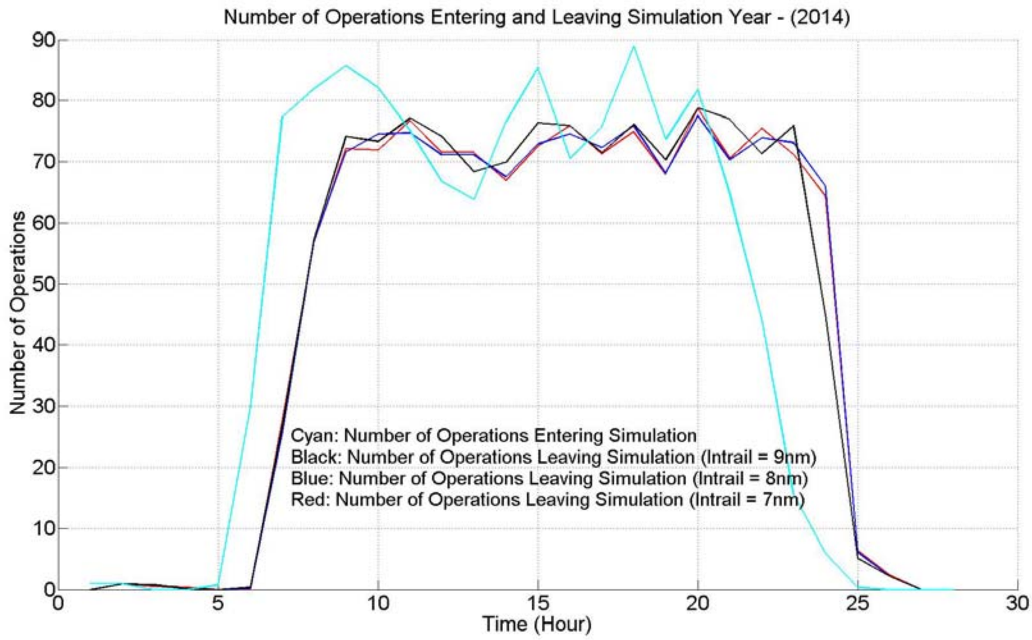

Figure 9-29: Number of Operations Entering and Leaving Simulation – LGA 2014.

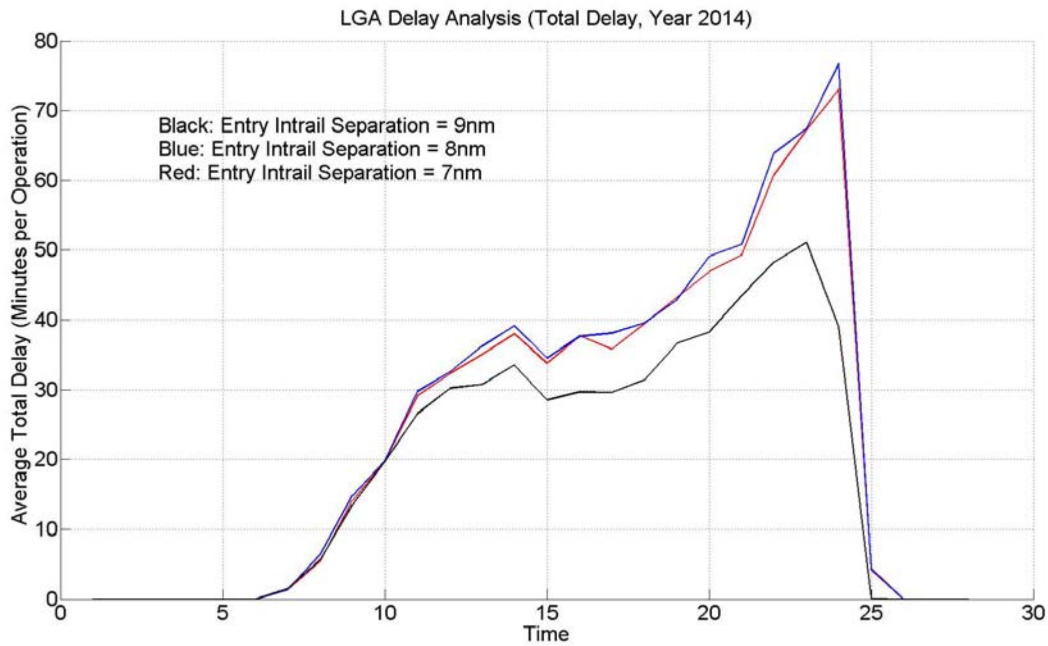

Figure 9-30: Average Total Delay – LGA, Year 2014.

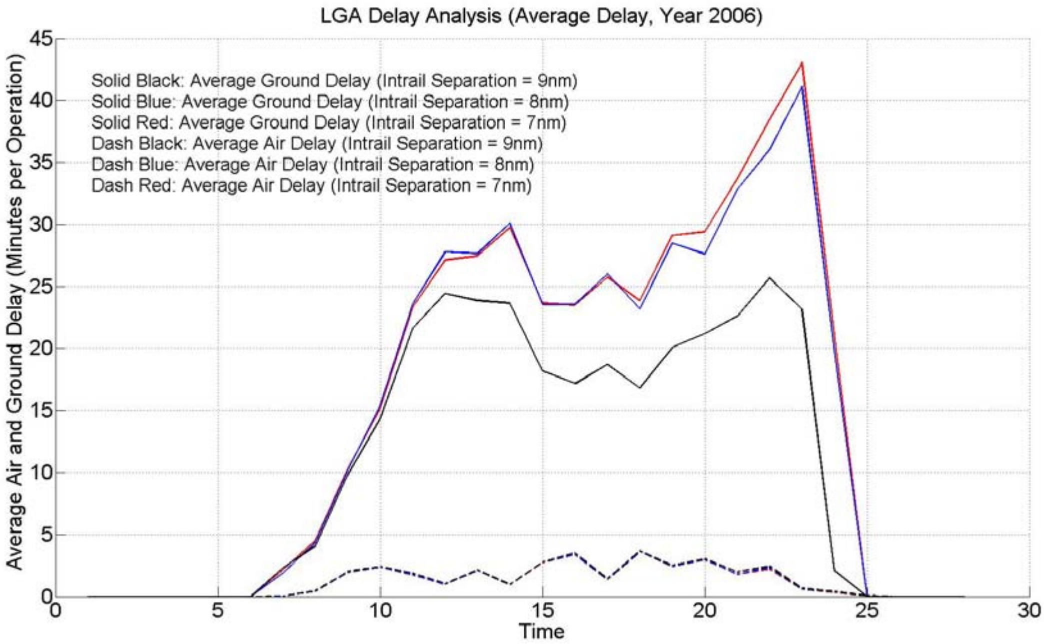

Figure 9-31: Average Air and Ground Delay – LGA, Year 2014.

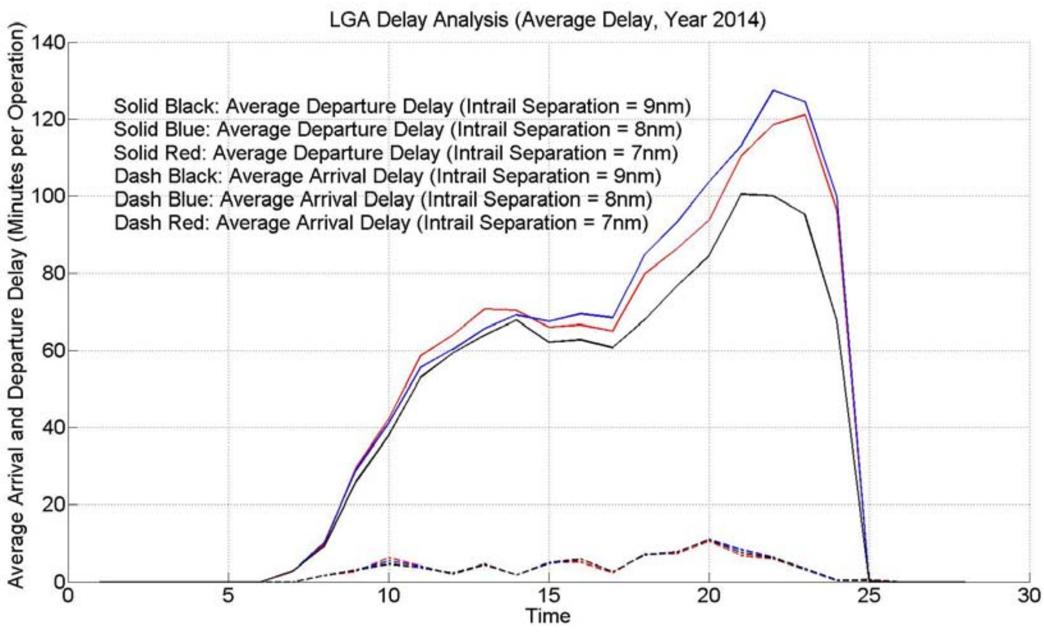

Figure 9-32: Average Arrival and Departure Delay – LGA, Year 2014.

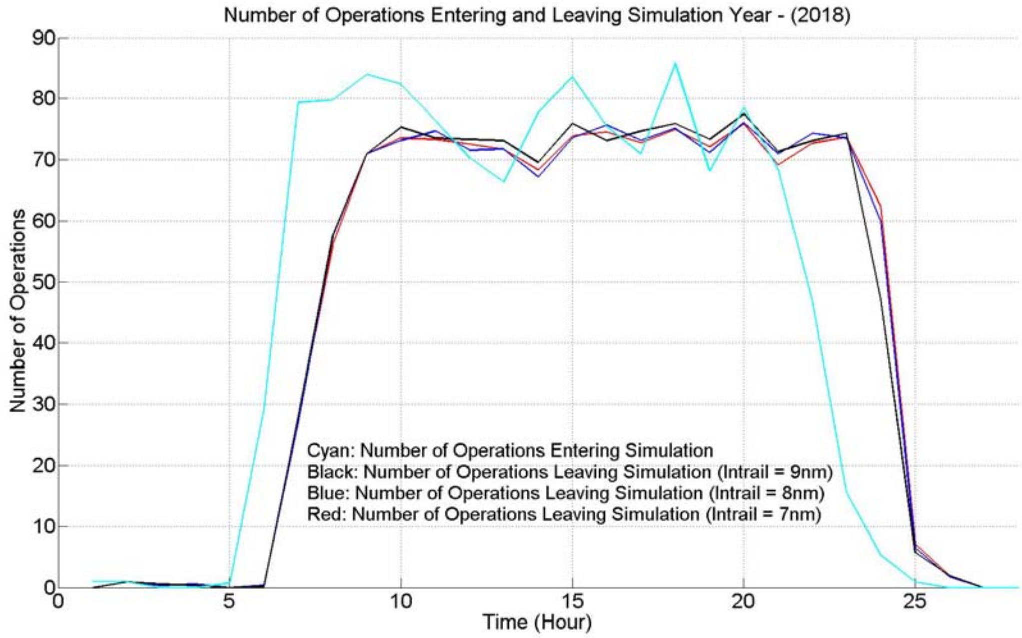

Figure 9-33: Number of Operations Entering and Leaving Simulation – LGA 2018.

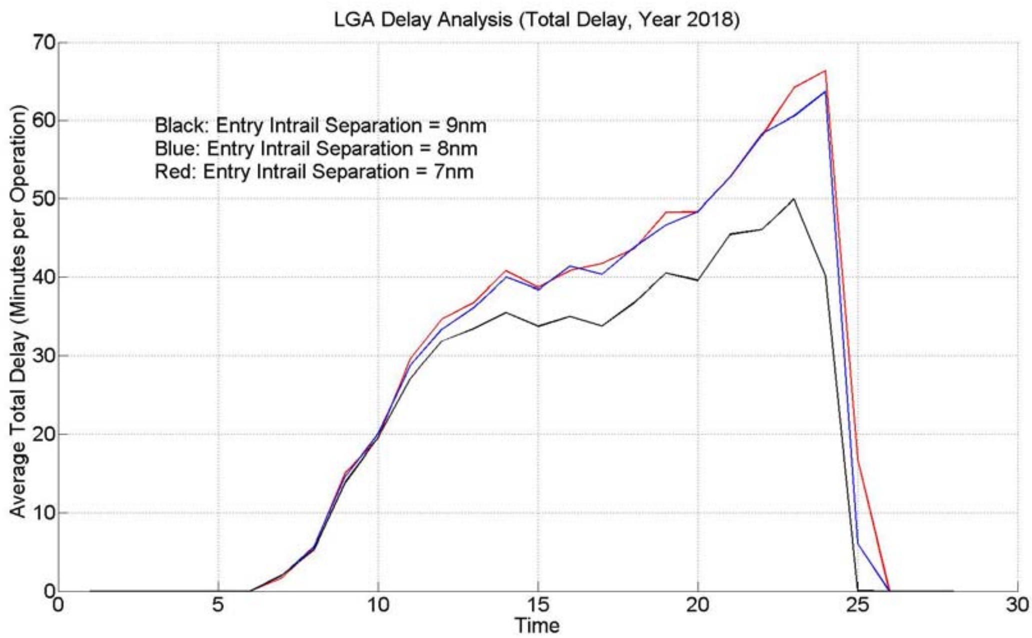

Figure 9-34: Average Total Delay – LGA, Year 2018.

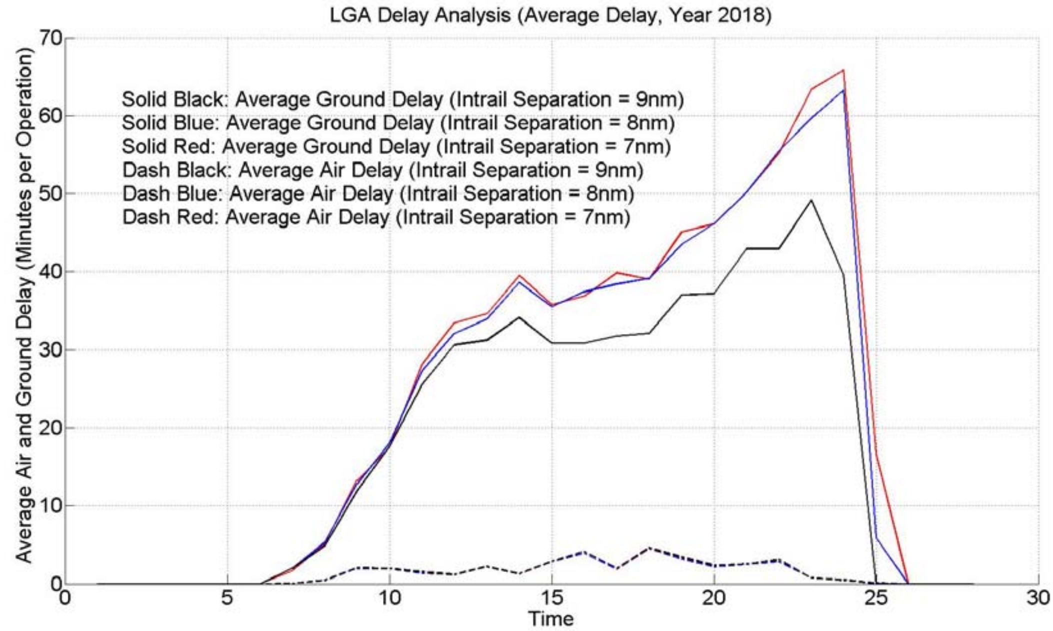

Figure 9-35: Average Air and Ground Delay – LGA, Year 2018.

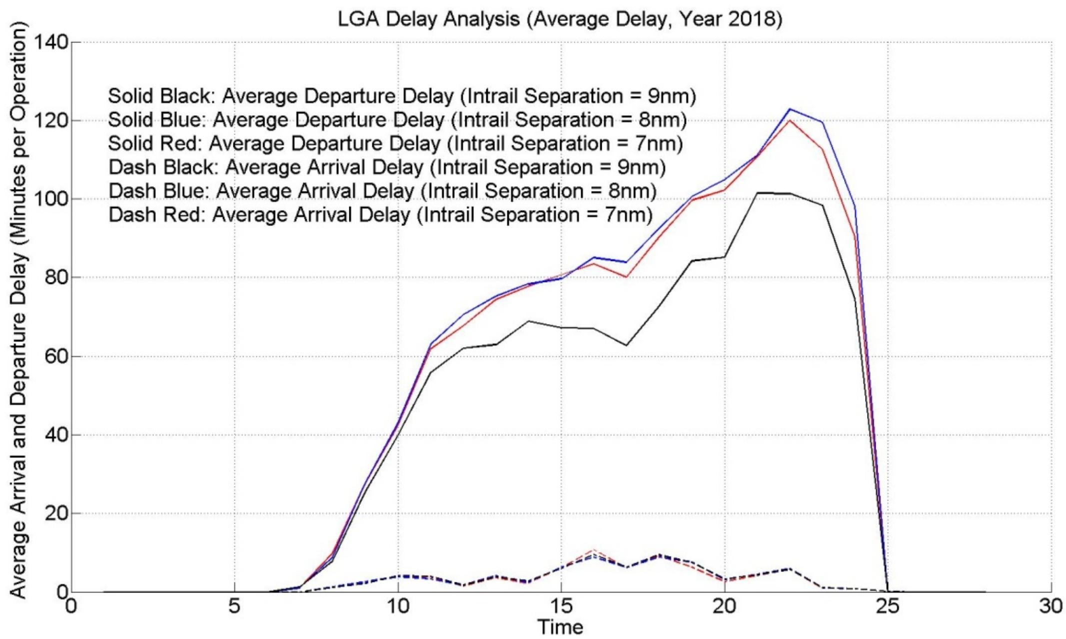

Figure 9-36: Average Arrival and Departure Delay – LGA, Year 2018.

Appendix B: Sample Matlab Code

LLM Analysis

Displaced Threshold

```
% ----------- Displaced Threshold Program --------------- %
% This program is to evaluate the possiblity to clear obstacles by
% displacing threshold
% Programmed by Yue Xu, Virginia Tech, ATSL
% Revised by Yue Xu, 7/7/2004, Version 1.1
% OCS surface is added in this revisement

clear all;
load object_database_3000_b;

% Base End Analysis
[DataNumber_B,b] = size(Object_Value_3000_B);

GPA = 5.0;
DisplacedThreshold = 0;
Betta = tan(GPA * pi/180 * 2/3);
DataNumber = 0;

for jj = 1:DataNumber_B
    DisplacedThreshold = 0;
   if Object_Index_3000_B(jj,1) ~= ' '
       DataNumber = DataNumber + 1;
       while DisplacedThreshold <= 1000      % Maximum Displaced
Threshold 1000 feet
           if (str2num(Object_RunwayLength_3000_B(jj,1:5))-
DisplacedThreshold)>=3000 % The remaining runway length greater than
3000 feet
             if (Object_Value_3000_B(jj,1) >= (200-DisplacedThreshold))
& (Object_Value_3000_B(jj,1) <= (50200-DisplacedThreshold))
               if abs(Object_Value_3000_B(jj,2)) <=
(9*Object_Value_3000_B(jj,1)/250+1964/5+9/250*DisplacedThreshold)
                   if (Object_Value_3000_B(jj,3) >
((Object_Value_3000_B(jj,1) + DisplacedThreshold)*Betta))
                       DisplacedThreshold = DisplacedThreshold + 100; %
If intrude, increase displaced threshold
                   elseif DisplacedThreshold == 0; % Under Q Surface
but not intrusive
                       break;
                   else
                       break;
                   end
               elseif abs(Object_Value_3000_B(jj,2)) <=
(5376/50000*Object_Value_3000_B(jj,1) + 678.496 + 5376/50000 *
DisplacedThreshold)
```

```
                    if (Object_Value_3000_B(jj,3) > (Betta *
(Object_Value_3000_B(jj,1)+DisplacedThreshold) + ...
                            1/4 * (abs(Object_Value_3000_B(jj,2))-
9/250*(Object_Value_3000_B(jj,1)+DisplacedThreshold)-1964/5)))
                        DisplacedThreshold = DisplacedThreshold +
100;  % If intrude, increase displaced threshold
                    elseif DisplacedThreshold == 0; % Under X Surface
but not intrusive
                       break;
                   else
                       break;
                   end
               else DisplacedThreshold = 0;
               break;
               end
           else DisplacedThreshold = 0;
               break;
           end
       else DisplacedThreshold = 9999;
           break;
       end
       if DisplacedThreshold == 1100
           DisplacedThreshold = 9999;
       end

   end
       DT_Airport_Index_3000_B(DataNumber,:)        =
Object_Index_3000_B(jj,1:4);
       DT_Airport_RunwayID_3000_B(DataNumber,:)     =
Object_RunwayID_3000_B(jj,1:3);
       DT_Airport_RunwayLength_3000_B(DataNumber,:) =
Object_RunwayLength_3000_B(jj,1:5);
       DT_Airport_Value_3000_B(DataNumber,:)        =
DisplacedThreshold;
end
end

save DT_WQS_5Degrees_B DT_Airport_Index_3000_B
DT_Airport_RunwayID_3000_B DT_Airport_RunwayLength_3000_B
DT_Airport_Value_3000_B;

% This part is to get the Cumulative Distribution Function

DT_100 = 0;
DT_200 = 0;
DT_300 = 0;
DT_400 = 0;
DT_500 = 0;
DT_600 = 0;
DT_700 = 0;
DT_800 = 0;
DT_900 = 0;
DT_1000= 0;
DT_0= 0;

for jj = 1:DataNumber
```

```
    if DT_Airport_Value_3000_B(jj) == 100
        DT_100 = DT_100 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 200
        DT_200 = DT_200 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 300
        DT_300 = DT_300 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 400
        DT_400 = DT_400 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 500
        DT_500 = DT_500 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 600
        DT_600 = DT_600 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 700
        DT_700 = DT_700 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 800
        DT_800 = DT_800 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 900
        DT_900 = DT_900 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 1000
        DT_1000 = DT_1000 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 0
        DT_0 = DT_0 + 1;
    end
end

DTs = [DT_0 DT_100 DT_200 DT_300 DT_400 DT_500 DT_600 DT_700 DT_800
DT_900 DT_1000];

DT_CDF(1) = 0;
DT_CDF(2) = DTs(1);

for pp = 1:10
    DT_CDF(pp+2)=DT_CDF(pp+1) + DTs(pp+1);
end

% subplot(2,1,1);
plot(DT_CDF(2:12)/DataNumber_B*100,'linewidth',2,'color','r');
set(gca,'XTicklabel',[' 0  ';' 100';' 200';' 300';' 400';' 500';'
600';' 700';' 800';' 900';'1000']);

xlabel('Displaced Threshold (feet)');ylabel('Percentage');
title('Percentage of Runways Clearing WQS and OCS (Base End)');
grid;

hold on;

% Reciprocal End Analysis

clear all;

load object_database_3000_r;

[DataNumber_R,r] = size(Object_Value_3000_R);

GPA = 5.0;
```

```
DisplacedThreshold = 0;
Betta = tan(GPA * pi/180 * 2/3);
DataNumber = 0;

for jj = 1:DataNumber_R
    DisplacedThreshold = 0;
   if Object_Index_3000_R(jj,1) ~= ' '
       DataNumber = DataNumber + 1;
       while DisplacedThreshold <= 1000      % Maximum Displaced
Threshold 1000 feet
           if (str2num(Object_RunwayLength_3000_R(jj,1:5))-
DisplacedThreshold)>=3000 % The rest runway length greater than 3000
feet
             if (Object_Value_3000_R(jj,1) >= (200-DisplacedThreshold))
& (Object_Value_3000_R(jj,1) <= (50200-DisplacedThreshold))
               if abs(Object_Value_3000_R(jj,2)) <=
(9*Object_Value_3000_R(jj,1)/250+1964/5+9/250*DisplacedThreshold)
                   if (Object_Value_3000_R(jj,3) >
((Object_Value_3000_R(jj,1) + DisplacedThreshold)*Betta))
                       DisplacedThreshold = DisplacedThreshold + 100; %
If intrude, increase displaced threshold
                   elseif DisplacedThreshold == 0; % Under Q Surface
but not intrusive
                       break;
                   else
                       break;
                   end
               elseif abs(Object_Value_3000_R(jj,2)) <=
(5376/50000*Object_Value_3000_R(jj,1) + 678.496 + 5376/50000 *
DisplacedThreshold)
                    if (Object_Value_3000_R(jj,3) > (Betta *
(Object_Value_3000_R(jj,1)+DisplacedThreshold) +...
                            1/4 * (abs(Object_Value_3000_R(jj,2))-
9/250*(Object_Value_3000_R(jj,1)+DisplacedThreshold)-1964/5)))
                        DisplacedThreshold = DisplacedThreshold +
100;  % If intrude, increase displaced threshold
                    elseif DisplacedThreshold == 0; % Under X Surface
but not intrusive
                       break;
                   else
                       break;
                   end
               else DisplacedThreshold = 0;
               break;
               end
           else DisplacedThreshold = 0;
               break;
           end
       else DisplacedThreshold = 9999;
           break;
       end
       if DisplacedThreshold == 1100
           DisplacedThreshold = 9999;
       end

   end
```

```
   DT_Airport_Index_3000_R(DataNumber,1:4) =
Object_Index_3000_R(jj,1:4);
   DT_Airport_RunwayID_3000_R(DataNumber,1:3)     =
Object_RunwayID_3000_R(jj,1:3);
   DT_Airport_RunwayLength_3000_R(DataNumber,1:5) =
Object_RunwayLength_3000_R(jj,1:5);
   DT_Airport_Value_3000_R(DataNumber,:) = DisplacedThreshold;
end
end

save DT_WQS_5Degrees_R DT_Airport_Index_3000_R
DT_Airport_RunwayID_3000_R DT_Airport_RunwayLength_3000_R
DT_Airport_Value_3000_R;

% This part is to get the Cumulative Distribution Function

DT_100 = 0;
DT_200 = 0;
DT_300 = 0;
DT_400 = 0;
DT_500 = 0;
DT_600 = 0;
DT_700 = 0;
DT_800 = 0;
DT_900 = 0;
DT_1000= 0;
DT_0 = 0;

for jj = 1:DataNumber
    if DT_Airport_Value_3000_R(jj) == 100
        DT_100 = DT_100 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 200
        DT_200 = DT_200 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 300
        DT_300 = DT_300 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 400
        DT_400 = DT_400 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 500
        DT_500 = DT_500 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 600
        DT_600 = DT_600 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 700
        DT_700 = DT_700 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 800
        DT_800 = DT_800 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 900
        DT_900 = DT_900 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 1000
        DT_1000 = DT_1000 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 0
        DT_0 = DT_0 + 1;
    end
end

DTs = [DT_0 DT_100 DT_200 DT_300 DT_400 DT_500 DT_600 DT_700 DT_800
DT_900 DT_1000];
```

```
DT_CDF(1) = 0;
DT_CDF(2) = DTs(1);
for pp = 1:10
    DT_CDF(pp+2)=DT_CDF(pp+1) + DTs(pp+1);
end

% subplot(2,1,2);
% plot(DT_CDF(2:12)/DataNumber_R*100,'linewidth',2,'color','m');
% set(gca,'XTicklabel',[' 0  ';' 100';' 200';' 300';' 400';' 500';'
600';' 700';' 800';' 900';'1000']);
%
% xlabel('Displaced Threshold (feet)');ylabel('Percentage');
title('Percentage of Runways Clearing WQS and OCS (Reciprocal End)');
% grid;
%
% hold on;

% This is to evaluate Obstacle Clearance Surface (OCS) by raising GPA

clear all;
load object_database_3000_b;

% Base End Analysis
[DataNumber_B,b] = size(Object_Value_3000_B);

GPA = 5.0;
DisplacedThreshold = 0;
Betta = GPA/102;
DataNumber = 0;

for jj = 1:DataNumber_B
    DisplacedThreshold = 0;
   if Object_Index_3000_B(jj,1) ~= ' '
       DataNumber = DataNumber + 1;
       while DisplacedThreshold <= 1000      % Maximum Displaced
Threshold 1000 feet
           if (str2num(Object_RunwayLength_3000_B(jj,1:5))-
DisplacedThreshold)>=3000 % The remaining runway length greater than
3000 feet
             if (Object_Value_3000_B(jj,1) >= (200-DisplacedThreshold))
& (Object_Value_3000_B(jj,1) <= (50200-DisplacedThreshold))
               if abs(Object_Value_3000_B(jj,2)) <=
(9*Object_Value_3000_B(jj,1)/250+1964/5+9/250*DisplacedThreshold)
                   if (Object_Value_3000_B(jj,3) >
((Object_Value_3000_B(jj,1)-200 + DisplacedThreshold)*Betta))
                       DisplacedThreshold = DisplacedThreshold + 100; %
If intrude, increase displaced threshold
                   elseif DisplacedThreshold == 0; % Under Q Surface
but not intrusive
                       break;
                   else
                       break;
                   end
```

```
                elseif abs(Object_Value_3000_B(jj,2)) <=
(5376/50000*Object_Value_3000_B(jj,1) + 678.496 + 5376/50000 *
DisplacedThreshold)
                    if (Object_Value_3000_B(jj,3) > (Betta *
(Object_Value_3000_B(jj,1) -200 + DisplacedThreshold)...
                            +1/4 * (abs(Object_Value_3000_B(jj,2))-
9/250*(Object_Value_3000_B(jj,1)+DisplacedThreshold)-1964/5)))
                        DisplacedThreshold = DisplacedThreshold +
100;  % If intrude, increase displaced threshold
                    elseif DisplacedThreshold == 0; % Under X Surface
but not intrusive
                       break;
                    else
                       break;
                    end
               elseif abs(Object_Value_3000_B(jj,2)) <=
(7576/50000*(Object_Value_3000_B(jj,1)+DisplacedThreshold)...
                      + 696.96) % Y surface
                if (Object_Value_3000_B(jj,3) > (Betta *
(Object_Value_3000_B(jj,1)-200 + DisplacedThreshold) + 1/7 * ...
                       (abs(Object_Value_3000_B(jj,2))-
1.0752*(Object_Value_3000_B(jj,1)+ DisplacedThreshold)-484.96)))
                        DisplacedThreshold = DisplacedThreshold +
100;  % If intrude, increase displaced threshold
                    elseif DisplacedThreshold == 0; % Under Y Surface
but not intrusive
                       break;
                    else
                       break;
                    end
                else DisplacedThreshold = 0;
                break;
                end
            else DisplacedThreshold = 0;
                break;
            end
        else DisplacedThreshold = 9999;
            break;
        end
        if DisplacedThreshold == 1100
            DisplacedThreshold = 9999;
        end

   end
        DT_Airport_Index_3000_B(DataNumber,:)        =
Object_Index_3000_B(jj,1:4);
        DT_Airport_RunwayID_3000_B(DataNumber,:)     =
Object_RunwayID_3000_B(jj,1:3);
        DT_Airport_RunwayLength_3000_B(DataNumber,:) =
Object_RunwayLength_3000_B(jj,1:5);
        DT_Airport_Value_3000_B(DataNumber,:)        =
DisplacedThreshold;
end
end
```

```
save DT_OCS_5Degrees_B DT_Airport_Index_3000_B
DT_Airport_RunwayID_3000_B DT_Airport_RunwayLength_3000_B
DT_Airport_Value_3000_B;

% This part is to get the Cumulative Distribution Function

DT_100 = 0;
DT_200 = 0;
DT_300 = 0;
DT_400 = 0;
DT_500 = 0;
DT_600 = 0;
DT_700 = 0;
DT_800 = 0;
DT_900 = 0;
DT_1000= 0;
DT_0= 0;

for jj = 1:DataNumber
    if DT_Airport_Value_3000_B(jj) == 100
        DT_100 = DT_100 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 200
        DT_200 = DT_200 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 300
        DT_300 = DT_300 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 400
        DT_400 = DT_400 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 500
        DT_500 = DT_500 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 600
        DT_600 = DT_600 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 700
        DT_700 = DT_700 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 800
        DT_800 = DT_800 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 900
        DT_900 = DT_900 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 1000
        DT_1000 = DT_1000 + 1;
    elseif DT_Airport_Value_3000_B(jj) == 0
        DT_0 = DT_0 + 1;
    end
end

DTs = [DT_0 DT_100 DT_200 DT_300 DT_400 DT_500 DT_600 DT_700 DT_800
DT_900 DT_1000];

DT_CDF(1) = 0;
DT_CDF(2) = DTs(1);

for pp = 1:10
    DT_CDF(pp+2)=DT_CDF(pp+1) + DTs(pp+1);
end

% subplot(2,1,1);
```

```
plot(DT_CDF(2:12)/DataNumber_B*100,'linewidth',2,'color','b');
set(gca,'XTicklabel',[' 0  ';' 100';' 200';' 300';' 400';' 500';'
600';' 700';' 800';' 900';'1000']);

xlabel('Displaced Threshold (feet)');ylabel('Percentage');
title('Percentage of Runways Clearing WQS and OCS (Base End)');
grid;

% Reciprocal End Analysis

clear all;

load object_database_3000_r;

[DataNumber_R,r] = size(Object_Value_3000_R);

GPA = 5.0;
DisplacedThreshold = 0;
Betta = GPA/102;
DataNumber = 0;

for jj = 1:DataNumber_R
    DisplacedThreshold = 0;
   if Object_Index_3000_R(jj,1) ~= ' '
       DataNumber = DataNumber + 1;
       while DisplacedThreshold <= 1000      % Maximum Displaced
Threshold 1000 feet
           if (str2num(Object_RunwayLength_3000_R(jj,1:5))-
DisplacedThreshold)>=3000 % The remaining runway length greater than
3000 feet
             if (Object_Value_3000_R(jj,1) >= (200-DisplacedThreshold))
& (Object_Value_3000_R(jj,1) <= (50200-DisplacedThreshold))
               if abs(Object_Value_3000_R(jj,2)) <=
(9*Object_Value_3000_R(jj,1)/250+1964/5+9/250*DisplacedThreshold)
                   if (Object_Value_3000_R(jj,3) >
((Object_Value_3000_R(jj,1)-200 + DisplacedThreshold)*Betta))
                       DisplacedThreshold = DisplacedThreshold + 100; %
If intrude, increase displaced threshold
                   elseif DisplacedThreshold == 0; % Under Q Surface
but not intrusive
                       break;
                   else
                       break;
                   end
               elseif abs(Object_Value_3000_R(jj,2)) <=
(5376/50000*Object_Value_3000_R(jj,1) + 678.496 + 5376/50000 *
DisplacedThreshold)
                    if (Object_Value_3000_R(jj,3) > (Betta *
((Object_Value_3000_R(jj,1)-200) + DisplacedThreshold)...
                            +1/4 * (abs(Object_Value_3000_R(jj,2))-
9/250*(Object_Value_3000_R(jj,1)+DisplacedThreshold)-1964/5)))
                        DisplacedThreshold = DisplacedThreshold +
100;  % If intrude, increase displaced threshold
                    elseif DisplacedThreshold == 0; % Under X Surface
but not intrusive
```

```
                       break;
                   else
                       break;
                    end
              elseif abs(Object_Value_3000_R(jj,2)) <=
(7576/50000*(Object_Value_3000_R(jj,1)+DisplacedThreshold)...
                      + 696.96) % Y surface
              if (Object_Value_3000_R(jj,3) > (Betta *
(Object_Value_3000_R(jj,1)-200 + DisplacedThreshold) + 1/7 * ...
                       (abs(Object_Value_3000_R(jj,2))-
1.0752*(Object_Value_3000_R(jj,1)+ DisplacedThreshold)-484.96)))
                        DisplacedThreshold = DisplacedThreshold +
100;  % If intrude, increase displaced threshold
                    elseif DisplacedThreshold == 0; % Under Y Surface
but not intrusive
                       break;
                    else
                       break;
                    end
              else DisplacedThreshold = 0;
              break;
              end
          else DisplacedThreshold = 0;
              break;
          end
      else DisplacedThreshold = 9999;
          break;
      end
      if DisplacedThreshold == 1100
          DisplacedThreshold = 9999;
      end

   end
   DT_Airport_Index_3000_R(DataNumber,1:4) =
Object_Index_3000_R(jj,1:4);
   DT_Airport_RunwayID_3000_R(DataNumber,1:3)     =
Object_RunwayID_3000_R(jj,1:3);
   DT_Airport_RunwayLength_3000_R(DataNumber,1:5) =
Object_RunwayLength_3000_R(jj,1:5);
   DT_Airport_Value_3000_R(DataNumber,:) = DisplacedThreshold;
end
end

save DT_OCS_5Degrees_R DT_Airport_Index_3000_R
DT_Airport_RunwayID_3000_R DT_Airport_RunwayLength_3000_R
DT_Airport_Value_3000_R;

% This part is to get the Cumulative Distribution Function

DT_100 = 0;
DT_200 = 0;
DT_300 = 0;
DT_400 = 0;
DT_500 = 0;
DT_600 = 0;
DT_700 = 0;
```

```
DT_800 = 0;
DT_900 = 0;
DT_1000= 0;
DT_0 = 0;

for jj = 1:DataNumber
    if DT_Airport_Value_3000_R(jj) == 100
        DT_100 = DT_100 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 200
        DT_200 = DT_200 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 300
        DT_300 = DT_300 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 400
        DT_400 = DT_400 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 500
        DT_500 = DT_500 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 600
        DT_600 = DT_600 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 700
        DT_700 = DT_700 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 800
        DT_800 = DT_800 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 900
        DT_900 = DT_900 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 1000
        DT_1000 = DT_1000 + 1;
    elseif DT_Airport_Value_3000_R(jj) == 0
        DT_0 = DT_0 + 1;
    end
end

DTs = [DT_0 DT_100 DT_200 DT_300 DT_400 DT_500 DT_600 DT_700 DT_800
DT_900 DT_1000];

DT_CDF(1) = 0;
DT_CDF(2) = DTs(1);
for pp = 1:10
    DT_CDF(pp+2)=DT_CDF(pp+1) + DTs(pp+1);
end

hold off;
```

Glide Path Angle Analysis

```
% This is to evaluate WQS and OCS by raising GPA
% Programmed by Yue Xu, Virginia Tech, ATSL,
% Revised by Yue Xu, 07/07/2004, Version 1.1
% OCS analysis are added

clear all;
load object_database_3000_b;

[ObjectNumbers_3000_B,b] = size(Object_Value_3000_B);

DataNumber = 0;
GPA_Max = 9.0;

for jj = 1:ObjectNumbers_3000_B
    GPA_trial = 3.0;
    Threshold = 0;

    if Object_Index_3000_B(jj,1)~= ' '
        DataNumber = DataNumber + 1;
        % Raising GPA to clear the objects
    while GPA_trial <= GPA_Max
        Betta = tan(2/3*pi*GPA_trial/180);
      if (Object_Value_3000_B(jj,1) >= 200) & (Object_Value_3000_B(jj,1)
<= 50200)
          if abs(Object_Value_3000_B(jj,2)) <=
(9*Object_Value_3000_B(jj,1)/250+1964/5)
              if (Object_Value_3000_B(jj,3) >
(Object_Value_3000_B(jj,1)*Betta))
                  GPA_trial = GPA_trial + 0.1;
              elseif GPA_trial == 3.0 % When the first time the height
is already lower than W surface, it is not intruding W surface
                  break;
               else
                  break;
              end
           elseif abs(Object_Value_3000_B(jj,2)) <=
(5376/50000*Object_Value_3000_B(jj,1) + 678.496)
               if (Object_Value_3000_B(jj,3) > (Betta *
Object_Value_3000_B(jj,1) + 1/4 * ...
                       (abs(Object_Value_3000_B(jj,2))-
9/250*Object_Value_3000_B(jj,1) -1964/5)))
                   GPA_trial = GPA_trial + 0.1;
               elseif GPA_trial == 3.0 % When the first time the height
is already lower than W surface, it is not intruding W surface
                  break;
               else
                   break;
               end
           else  GPA_trial = 3.0;
               break;
          end
```

```
      else  GPA_trial = 3.0;
          break;
      end
  end
  GPA_Airport_Index_3000_B(DataNumber,1:4) =
Object_Index_3000_B(jj,1:4) ;
  GPA_Airport_Value_3000_B(DataNumber)     = GPA_trial;
end
end

GPA_Airport_Value_3000_B = (GPA_Airport_Value_3000_B)';
save GPA_WQS_B GPA_Airport_Index_3000_B GPA_Airport_Value_3000_B
Airport_Latitude_3000_B Airport_Longitude_3000_B;

% This part is to get cumulative curve

GPA_3 = 0;
GPA_4 = 0;
GPA_5 = 0;
GPA_6 = 0;
GPA_7 = 0;
GPA_8 = 0;
GPA_9 = 0;

for jj = 1:DataNumber
    if GPA_Airport_Value_3000_B(jj) == 3.0
        GPA_3 = GPA_3 + 1;
    elseif GPA_Airport_Value_3000_B(jj) <= 4.0
        GPA_4 = GPA_4 + 1;
    elseif GPA_Airport_Value_3000_B(jj) <= 5.0
        GPA_5 = GPA_5 + 1;
    elseif GPA_Airport_Value_3000_B(jj) <= 6.0
        GPA_6 = GPA_6 + 1;
    elseif GPA_Airport_Value_3000_B(jj) <= 7.0
        GPA_7 = GPA_7 + 1;
    elseif GPA_Airport_Value_3000_B(jj) <= 8.0
        GPA_8 = GPA_8 + 1;
    elseif GPA_Airport_Value_3000_B(jj) <= 9.0
        GPA_9 = GPA_9 + 1;
    end
end

GPAs = [GPA_3 GPA_4 GPA_5 GPA_6 GPA_7 GPA_8 GPA_9];

GPA_CDF(1) = GPAs(1);
for pp = 1:6
    GPA_CDF(pp+1)=GPA_CDF(pp) + GPAs(pp+1);
end

% subplot(2,1,1);
plot(GPA_CDF/ObjectNumbers_3000_B*100,'r','linewidth',2);
set(gca,'XTicklabel',['3.0';'4.0';'5.0';'6.0';'7.0';'8.0';'9.0';]);

xlabel('Glide Path Angle (Degrees)');ylabel('Percentage'); title('Base
End');
```

```
hold on;
grid;

% ---- for reciprocal end information ---- %

clear all;
load object_database_3000_r;


[ObjectNumbers_3000_R,r] = size(Object_Value_3000_R);

GPA_Max = 9.0;
DataNumber = 0;

for jj = 1:ObjectNumbers_3000_R
    GPA_trial = 3.0;
    Threshold = 0;

    if Object_Index_3000_R(jj,1)~=' '
        DataNumber = DataNumber + 1;
        % Raising GPA to clear the objects
    while GPA_trial <= GPA_Max
        Betta = tan(2/3*pi*GPA_trial/180);
      if (Object_Value_3000_R(jj,1) >= 200) & (Object_Value_3000_R(jj,1)
<= 50200)
          if abs(Object_Value_3000_R(jj,2)) <=
(9*Object_Value_3000_R(jj,1)/250+9640/25)
              if (Object_Value_3000_R(jj,3) >
(Object_Value_3000_R(jj,1)*Betta))
                  GPA_trial = GPA_trial + 0.1;
              elseif GPA_trial == 3.0 % When the first time the height
is already lower than W surface, it is not intruding W surface
                  break;
               else
                  break;
              end
           elseif abs(Object_Value_3000_R(jj,2)) <=
(5376/50000*Object_Value_3000_R(jj,1) + 678.496)
               if (Object_Value_3000_R(jj,3) > (Betta *
Object_Value_3000_R(jj,1) + 1/4 * ...
                       (abs(Object_Value_3000_R(jj,2))-
9/250*Object_Value_3000_R(jj,1) -9640/25)))
                   GPA_trial = GPA_trial + 0.1;
               elseif GPA_trial == 3.0 % When the first time the height
is already lower than W surface, it is not intruding W surface
                  break;
               else
                   break;
               end
           else  GPA_trial = 3.0;
               break;
          end
      else  GPA_trial = 3.0;
          break;
```

```
        end
    end
    GPA_Airport_Index_3000_R(DataNumber,1:4) =
Object_Index_3000_R(jj,1:4) ;
    GPA_Airport_Value_3000_R(DataNumber)    = GPA_trial;
end
end

GPA_Airport_Value_3000_R = (GPA_Airport_Value_3000_R)';
save GPA_WQS_R GPA_Airport_Index_3000_R GPA_Airport_Value_3000_R
Airport_Latitude_3000_R Airport_Longitude_3000_R;

% This part is to get cumulative curve

GPA_3 = 0;
GPA_4 = 0;
GPA_5 = 0;
GPA_6 = 0;
GPA_7 = 0;
GPA_8 = 0;
GPA_9 = 0;

for jj = 1:DataNumber
    if GPA_Airport_Value_3000_R(jj) == 3.0
        GPA_3 = GPA_3 + 1;
    elseif GPA_Airport_Value_3000_R(jj) <= 4.0
        GPA_4 = GPA_4 + 1;
    elseif GPA_Airport_Value_3000_R(jj) <= 5.0
        GPA_5 = GPA_5 + 1;
    elseif GPA_Airport_Value_3000_R(jj) <= 6.0
        GPA_6 = GPA_6 + 1;
    elseif GPA_Airport_Value_3000_R(jj) <= 7.0
        GPA_7 = GPA_7 + 1;
    elseif GPA_Airport_Value_3000_R(jj) <= 8.0
        GPA_8 = GPA_8 + 1;
    elseif GPA_Airport_Value_3000_R(jj) <= 9.0
        GPA_9 = GPA_9 + 1;
    end
end

GPAs = [GPA_3 GPA_4 GPA_5 GPA_6 GPA_7 GPA_8 GPA_9];

GPA_CDF(1) = GPAs(1);
for pp = 1:6
    GPA_CDF(pp+1)=GPA_CDF(pp) + GPAs(pp+1);
end


clear all;
load object_database_3000_b;

[ObjectNumbers_3000_B,b] = size(Object_Value_3000_B);

DataNumber = 0;
GPA_Max = 9.0;
```

```
for jj = 1:ObjectNumbers_3000_B
    GPA_trial = 3.0;
    Threshold = 0;

    if Object_Index_3000_B(jj,1)~= ' '
        DataNumber = DataNumber + 1;
        % Raising GPA to clear the objects
    while GPA_trial <= GPA_Max
        Betta = GPA_trial/102; % Slope of OCS is S = 102/GPA;
      if (Object_Value_3000_B(jj,1) >= 200) & (Object_Value_3000_B(jj,1)
<= 50200) % within the length of OCS
          if abs(Object_Value_3000_B(jj,2)) <=
(9*Object_Value_3000_B(jj,1)/250+1964/5) % within the width of OCS
              if (Object_Value_3000_B(jj,3) >
((Object_Value_3000_B(jj,1)-200) * Betta)) % W surface
                  GPA_trial = GPA_trial + 0.1;
              elseif GPA_trial == 3.0 % When the first time the height
is already lower than W surface, it is not intruding W surface
                  break;
               else
                  break;
              end
           elseif abs(Object_Value_3000_B(jj,2)) <=
(5376/50000*Object_Value_3000_B(jj,1) + 678.496) % X surface
               if (Object_Value_3000_B(jj,3) > (Betta *
(Object_Value_3000_B(jj,1)-200) + 1/4 * ...
                       (abs(Object_Value_3000_B(jj,2))-
9/250*Object_Value_3000_B(jj,1) -1964/5)))
                   GPA_trial = GPA_trial + 0.1;
               elseif GPA_trial == 3.0 % When the first time the height
is already lower than W surface, it is not intruding W surface
                  break;
               else
                   break;
               end
          elseif abs(Object_Value_3000_B(jj,2)) <=
(7576/50000*Object_Value_3000_B(jj,1) + 969.696) % Y surface
               if (Object_Value_3000_B(jj,3) > (Betta *
(Object_Value_3000_B(jj,1)-200) + 1/7 * ...
                       (abs(Object_Value_3000_B(jj,2))- 0.10752 *
Object_Value_3000_B(jj,1) - 678.496)))
                   GPA_trial = GPA_trial + 0.1;
               elseif GPA_trial == 3.0 % When the first time the height
is already lower than W surface, it is not intruding W surface
                  break;
               else
                   break;
               end
           else  GPA_trial = 3.0;
               break;
          end
      else  GPA_trial = 3.0;
          break;
      end
  end
```

```
  GPA_Airport_Index_3000_B(DataNumber,1:4) =
Object_Index_3000_B(jj,1:4) ;
  GPA_Airport_Value_3000_B(DataNumber)     = GPA_trial;
end
end

GPA_Airport_Value_3000_B = (GPA_Airport_Value_3000_B)';
save GPA_OCS_B GPA_Airport_Index_3000_B GPA_Airport_Value_3000_B
Airport_Latitude_3000_B Airport_Longitude_3000_B;

% This part is to get cumulative curve

GPA_3 = 0;
GPA_4 = 0;
GPA_5 = 0;
GPA_6 = 0;
GPA_7 = 0;
GPA_8 = 0;
GPA_9 = 0;

for jj = 1:DataNumber
    if GPA_Airport_Value_3000_B(jj) == 3.0
        GPA_3 = GPA_3 + 1;
    elseif GPA_Airport_Value_3000_B(jj) <= 4.0
        GPA_4 = GPA_4 + 1;
    elseif GPA_Airport_Value_3000_B(jj) <= 5.0
        GPA_5 = GPA_5 + 1;
    elseif GPA_Airport_Value_3000_B(jj) <= 6.0
        GPA_6 = GPA_6 + 1;
    elseif GPA_Airport_Value_3000_B(jj) <= 7.0
        GPA_7 = GPA_7 + 1;
    elseif GPA_Airport_Value_3000_B(jj) <= 8.0
        GPA_8 = GPA_8 + 1;
    elseif GPA_Airport_Value_3000_B(jj) <= 9.0
        GPA_9 = GPA_9 + 1;
    end
end

GPAs = [GPA_3 GPA_4 GPA_5 GPA_6 GPA_7 GPA_8 GPA_9];

GPA_CDF(1) = GPAs(1);
for pp = 1:6
    GPA_CDF(pp+1)=GPA_CDF(pp) + GPAs(pp+1);
end

% subplot(2,1,1);
plot(GPA_CDF/ObjectNumbers_3000_B*100,'b','linewidth',2);
set(gca,'XTicklabel',['3.0';'4.0';'5.0';'6.0';'7.0';'8.0';'9.0';]);

xlabel('Glide Path Angle (Degrees)');ylabel('Percentage of Runways');
title('Base End');

% ---- for reciprocal end information ---- %

clear all;
```

```
load object_database_3000_r;

[ObjectNumbers_3000_R,r] = size(Object_Value_3000_R);

GPA_Max = 9.0;
DataNumber = 0;

for jj = 1:ObjectNumbers_3000_R
    GPA_trial = 3.0;
    Threshold = 0;

    if Object_Index_3000_R(jj,1)~=' '
        DataNumber = DataNumber + 1;
        % Raising GPA to clear the objects
    while GPA_trial <= GPA_Max
        Betta = GPA_trial/102; % Slope of OCS is S = 102/GPA;
      if (Object_Value_3000_R(jj,1) >= 200) & (Object_Value_3000_R(jj,1)
<= 50200)
          if abs(Object_Value_3000_R(jj,2)) <=
(9*Object_Value_3000_R(jj,1)/250+9640/25) % W surface
              if (Object_Value_3000_R(jj,3) >
((Object_Value_3000_R(jj,1)-200)*Betta))
                  GPA_trial = GPA_trial + 0.1;
              elseif GPA_trial == 3.0 % When the first time the height
is already lower than W surface, it is not intruding W surface
                  break;
               else
                  break;
              end
           elseif abs(Object_Value_3000_R(jj,2)) <=
(5376/50000*Object_Value_3000_R(jj,1) + 678.496) % Y surface
               if (Object_Value_3000_R(jj,3) > (Betta *
(Object_Value_3000_R(jj,1)-200) + 1/4 * ...
                       (abs(Object_Value_3000_R(jj,2))-
9/250*Object_Value_3000_R(jj,1) -9640/25)))
                   GPA_trial = GPA_trial + 0.1;
               elseif GPA_trial == 3.0 % When the first time the height
is already lower than W surface, it is not intruding W surface
                  break;
               else
                   break;
               end
           elseif abs(Object_Value_3000_R(jj,2)) <=
(7576/50000*Object_Value_3000_R(jj,1) + 969.696) % Y surface
               if (Object_Value_3000_R(jj,3) > (Betta *
(Object_Value_3000_R(jj,1)-200) + 1/7 * ...
                       (abs(Object_Value_3000_R(jj,2))- 0.10752 *
Object_Value_3000_R(jj,1) - 678.496)))
                   GPA_trial = GPA_trial + 0.1;
               elseif GPA_trial == 3.0 % When the first time the height
is already lower than W surface, it is not intruding W surface
                  break;
               else
                   break;
               end
           else  GPA_trial = 3.0;
```

```
                break;
            end
        else  GPA_trial = 3.0;
            break;
        end
    end
    GPA_Airport_Index_3000_R(DataNumber,1:4) = 
Object_Index_3000_R(jj,1:4) ;
    GPA_Airport_Value_3000_R(DataNumber)   = GPA_trial;
end
end

GPA_Airport_Value_3000_R = (GPA_Airport_Value_3000_R)';
save GPA_OCS_R GPA_Airport_Index_3000_R GPA_Airport_Value_3000_R 
Airport_Latitude_3000_R Airport_Longitude_3000_R;

% This part is to get cumulative curve

GPA_3 = 0;
GPA_4 = 0;
GPA_5 = 0;
GPA_6 = 0;
GPA_7 = 0;
GPA_8 = 0;
GPA_9 = 0;

for jj = 1:DataNumber
    if GPA_Airport_Value_3000_R(jj) == 3.0
        GPA_3 = GPA_3 + 1;
    elseif GPA_Airport_Value_3000_R(jj) <= 4.0
        GPA_4 = GPA_4 + 1;
    elseif GPA_Airport_Value_3000_R(jj) <= 5.0
        GPA_5 = GPA_5 + 1;
    elseif GPA_Airport_Value_3000_R(jj) <= 6.0
        GPA_6 = GPA_6 + 1;
    elseif GPA_Airport_Value_3000_R(jj) <= 7.0
        GPA_7 = GPA_7 + 1;
    elseif GPA_Airport_Value_3000_R(jj) <= 8.0
        GPA_8 = GPA_8 + 1;
    elseif GPA_Airport_Value_3000_R(jj) <= 9.0
        GPA_9 = GPA_9 + 1;
    end
end

GPAs = [GPA_3 GPA_4 GPA_5 GPA_6 GPA_7 GPA_8 GPA_9];

GPA_CDF(1) = GPAs(1);
for pp = 1:6
    GPA_CDF(pp+1)=GPA_CDF(pp) + GPAs(pp+1);
end

hold off;
```

Offset Course Analysis

```
% This program is to evaluate effecitveness of Course Offest to clear
% obstacles
% Programmed by Yue Xu, Virginia Tech, 12/12/2003, Version 1.0
% Revised by Yue Xu, 07/09/2004, Version 2.0, OCS analysis is added


% This is for WQS analysis, Base End

clear all;

% Initial Values
DA = 300;
aifa_final = 5.0;
GPA = 5.0;

load object_database_3000_b;
[ObjectNumbers_3000_B,b] = size(Object_Value_3000_B);

DataNumber = 0;

for obstacle_number = 1:ObjectNumbers_3000_B

    aifa_trial_1 = 0.0;

    while aifa_trial_1 <= aifa_final

        Betta = tan(2/3*pi*GPA/180);

            X_0 = (DA / tan(GPA * pi / 180) - 1150) * sin(aifa_trial_1
* pi / 180);
            Y_0 = (DA / tan(GPA * pi / 180) - 1150) * (1 -
cos(aifa_trial_1 * pi / 180));
            dis_B_1(obstacle_number) =
Object_Value_3000_B(obstacle_number,1) - X_0;
            off_B_1(obstacle_number) =
Object_Value_3000_B(obstacle_number,2) - Y_0;


      if (dis_B_1(obstacle_number) >= 200) & (dis_B_1(obstacle_number)
<= 50200)
          if abs(off_B_1(obstacle_number)) <= (9 *
dis_B_1(obstacle_number)/250 + 1964/5)
              if (Object_Value_3000_B(obstacle_number,3) >
(dis_B_1(obstacle_number)*Betta))
                  aifa_trial_1 = aifa_trial_1 + 0.5;
              elseif aifa_trial_1 == 0.0
                  % When the first time the height is already lower
than W surface, it is not intruding W surface
```

```
                    break;
                else
                    break;
                end
            elseif abs(off_B_1(obstacle_number)) <=
(5376/50000*dis_B_1(obstacle_number) + 678.496)
                if (Object_Value_3000_B(obstacle_number,3) > (Betta *
dis_B_1(obstacle_number) + 1/4 * (abs(off_B_1(obstacle_number))-
9/250*dis_B_1(obstacle_number) -1964/5)))
                    aifa_trial_1 = aifa_trial_1 + 0.5;
                elseif aifa_trial_1 == 0.0
                    % When the first time the height is already lower
than X surface, it is not intruding W surface
                   break;
                else
                    break;
                end
            else  aifa_trial_1 = 0.0;
                break;
          end
      elseif  aifa_trial_1 == 0.0;
          aifa_trial_1 = 0.0;
          break;
      else
          break;
      end
  end
  Offset_Airport_Index_3000_B_1(obstacle_number,1:4) =
Object_Index_3000_B(obstacle_number,1:4) ;
  Offset_Airport_Value_3000_B_1(obstacle_number)     = aifa_trial_1;
end

% ---------------- Offset from the below (anti Clockwise) -------------
------ %

for obstacle_number = 1:ObjectNumbers_3000_B

    aifa_trial_2 = 0.0;

    % Raising Offset Angles to clear the objects
    while aifa_trial_2 <= aifa_final

            Betta = tan(2/3*pi*GPA/180);
            % Position Transformation of critical obstacles to new
            % coordinate system so that WQS equation need not change
            X_0 = (DA / tan(GPA * pi / 180) - 1150) * sin(aifa_trial_2
* pi / 180);
            Y_0 = - (DA / tan(GPA * pi / 180) - 1150) * (1 -
cos(aifa_trial_2 * pi / 180));
            dis_B_2(obstacle_number) =
Object_Value_3000_B(obstacle_number,1) - X_0;
            off_B_2(obstacle_number) =
Object_Value_3000_B(obstacle_number,2) - Y_0;
```

```
      if (dis_B_2(obstacle_number) >= 200) & (dis_B_2(obstacle_number)
<= 50200)
          if abs(off_B_2(obstacle_number)) <=
(9*dis_B_2(obstacle_number)/250+1964/5)
              if (Object_Value_3000_B(obstacle_number,3) >
(dis_B_2(obstacle_number)*Betta))
                  aifa_trial_2 = aifa_trial_2 + 0.5;
              elseif aifa_trial_2 == 0.0
                  % When the first time the height is already lower
than W surface, it is not intruding W surface
                  break;
              else
                  break;
              end
           elseif abs(off_B_2(obstacle_number))  <= (5376/50000 *
dis_B_2(obstacle_number) + 678.496)
               if (Object_Value_3000_B(obstacle_number,3) > (Betta *
dis_B_2(obstacle_number)+1/4 * (abs(off_B_2(obstacle_number))- 9/250 *
dis_B_2(obstacle_number) -1964/5)))
                   aifa_trial_2 = aifa_trial_2 + 0.5;
               elseif aifa_trial_2 == 0.0
                   % When the first time the height is already lower
than X surface, it is not intruding W surface
                  break;
               else
                   break;
               end
           else  aifa_trial_2 = 0.0;
               break;
          end
      elseif  aifa_trial_2 == 0.0;
          aifa_trial_2 = 0.0;
          break;
      else
          break;
      end
  end
  Offset_Airport_Index_3000_B_2(obstacle_number,1:4) =
Object_Index_3000_B(obstacle_number,1:4) ;
  Offset_Airport_Value_3000_B_2(obstacle_number)     = aifa_trial_2;
end
% the end

Offset_Airport_Value_3000_B_1 = (Offset_Airport_Value_3000_B_1)';
Offset_Airport_Value_3000_B_2 = (Offset_Airport_Value_3000_B_2)';

for obstacle_number = 1:ObjectNumbers_3000_B
    if Offset_Airport_Value_3000_B_2(obstacle_number) >=
Offset_Airport_Value_3000_B_1(obstacle_number)
        Offset_Airport_Value_3000_B(obstacle_number) =
Offset_Airport_Value_3000_B_1(obstacle_number);
    else Offset_Airport_Value_3000_B(obstacle_number) =
Offset_Airport_Value_3000_B_2(obstacle_number);
    end
end
```

```
Offset_Airport_Value_3000_B = (Offset_Airport_Value_3000_B)';

save OC_WQS_5Degrees_B Object_Index_3000_B Offset_Airport_Value_3000_B;

% This part is to get cumulative curve

offset_00 = 0;
offset_05 = 0;
offset_10 = 0;
offset_15 = 0;
offset_20 = 0;
offset_25 = 0;
offset_30 = 0;
offset_35 = 0;
offset_40 = 0;
offset_45 = 0;
offset_50 = 0;


for obstacle_number = 1:ObjectNumbers_3000_B
    if Offset_Airport_Value_3000_B(obstacle_number) == 0.0
        offset_00 = offset_00 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 0.5
        offset_05 = offset_05 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 1.0
        offset_10 = offset_10 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 1.5
        offset_15 = offset_15 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 2.0
        offset_20 = offset_20 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 2.5
        offset_25 = offset_25 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 3.0
        offset_30 = offset_30 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 3.5
        offset_35 = offset_35 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 4.0
        offset_40 = offset_40 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 4.5
        offset_45 = offset_45 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 5.0
        offset_50 = offset_50 + 1;
    end
end

Offsets = [offset_00 offset_05 offset_10 offset_15 offset_20 offset_25
offset_30 offset_35 offset_40 offset_45 offset_50 ];

Offset_CDF = cumsum(Offsets) / ObjectNumbers_3000_B *100;

hold on;

% subplot(2,1,1);
plot(Offset_CDF,'r','linewidth',2);
```

```
set(gca,'XTicklabel',['0.0';'0.5';'1.0';'1.5';'2.0';'2.5';'3.0';'3.5';'
4.0';'4.5';'5.0';]);

xlabel('Offset Course (Degrees).');ylabel('Percentage');
title('Percentage of Runways Clearing OCS and WQS (Base End)');
grid;


% -------------------- For the Reciprocal End--------------------

clear;

load object_database_3000_r;

% Initial Values
DA = 300;
aifa_final = 5.0;
GPA = 5.0;

DataNumber = 0;

[ObjectNumbers_3000_R,r] = size(Object_Value_3000_R);

for obstacle_number = 1:ObjectNumbers_3000_R

    aifa_trial_3 = 0.0;

    % Raising Offset Angles to clear the objects

    while aifa_trial_3 <= aifa_final

        Betta = tan(2/3*pi*GPA/180);

            X_0 = (DA / tan(GPA * pi / 180) - 1150) * sin(aifa_trial_3
* pi / 180);
            Y_0 = (DA / tan(GPA * pi / 180) - 1150) * (1 -
cos(aifa_trial_3 * pi / 180));
            dis_R_1(obstacle_number) =
Object_Value_3000_R(obstacle_number,1) - X_0;
            off_R_1(obstacle_number) =
Object_Value_3000_R(obstacle_number,2) - Y_0;


      if (dis_R_1(obstacle_number) >= 200) & (dis_R_1(obstacle_number)
<= 50200)
          if abs(off_R_1(obstacle_number)) <= (9 *
dis_R_1(obstacle_number)/250 + 1964/5)
              if (Object_Value_3000_R(obstacle_number,3) >
(dis_R_1(obstacle_number)*Betta))
                  aifa_trial_3 = aifa_trial_3 + 0.5;
              elseif aifa_trial_3 == 0.0
                  % When the first time the height is already lower
than W surface, it is not intruding W surface
                  aifa_trial_3 = 0.0;
```

```
                break;
            else
                break;
            end
         elseif abs(off_R_1(obstacle_number)) <=
(5376/50000*dis_R_1(obstacle_number) + 678.496)
             if (Object_Value_3000_R(obstacle_number,3) > (Betta *
dis_R_1(obstacle_number) + 1/4 * (abs(off_R_1(obstacle_number))-
9/250*dis_R_1(obstacle_number) -1964/5)))
                 aifa_trial_3 = aifa_trial_3 + 0.5;
             elseif aifa_trial_3 == 0.0
                 % When the first time the height is already lower
than W surface, it is not intruding W surface
                 aifa_trial_3 = 0.0;
                break;
             else
                 break;
             end
         else  aifa_trial_3 = 0.0;
             break;
        end
      elseif  aifa_trial_3 == 0.0;
          aifa_trial_3 = 0.0;
          break;
      else
          break;
      end
  end
  Offset_Airport_Index_3000_R_1(obstacle_number,1:4) =
Object_Index_3000_R(obstacle_number,1:4) ;
  Offset_Airport_Value_3000_R_1(obstacle_number)     = aifa_trial_3;
end


% ---------------- Offset from the below (anti Clockwise) -------------
------ %

for obstacle_number = 1:ObjectNumbers_3000_R

    aifa_trial_4 = 0.0;

    % Raising Offset Angles to clear the objects
    while aifa_trial_4 <= aifa_final

            Betta = tan(2/3*pi*GPA/180);

            X_0 = (DA / tan(GPA * pi / 180) - 1150) * sin(aifa_trial_4
* pi / 180);
            Y_0 = (DA / tan(GPA * pi / 180) - 1150) * (1 -
cos(aifa_trial_4 * pi / 180));
            dis_R_2(obstacle_number) =
Object_Value_3000_R(obstacle_number,1) - X_0;
            off_R_2(obstacle_number) =
Object_Value_3000_R(obstacle_number,2) - Y_0;
```

```
      if (dis_R_2(obstacle_number) >= 200) & (dis_R_2(obstacle_number)
<= 50200)
          if abs(off_R_2(obstacle_number)) <=
(9*dis_R_2(obstacle_number)/250+1964/5)
              if (Object_Value_3000_R(obstacle_number,3) >
(dis_R_2(obstacle_number)*Betta))
                  aifa_trial_4 = aifa_trial_4 + 0.5;
              elseif aifa_trial_4 == 0.0
                  % When the first time the height is already lower
than W surface, it is not intruding W surface
                  break;
              else
                  break;
              end
           elseif abs(off_R_2(obstacle_number)) <= (5376/50000 *
dis_R_2(obstacle_number) + 678.496)
               if (Object_Value_3000_R(obstacle_number,3) > (Betta *
dis_R_2(obstacle_number)+1/4 * (abs(off_R_2(obstacle_number))- 9/250 *
dis_R_2(obstacle_number) -1964/5)))
                   aifa_trial_4 = aifa_trial_4 + 0.5;
               elseif aifa_trial_4 == 0.0
                   % When the first time the height is already lower
than X surface, it is not intruding W surface
                  break;
               else
                   break;
               end
           else  aifa_trial_4 = 0.0;
               break;
          end
      elseif  aifa_trial_4 == 0.0;
          aifa_trial_4 = 0.0;
          break;
      else
          break;
      end
  end
  Offset_Airport_Index_3000_R_2(obstacle_number,1:4) =
Object_Index_3000_R(obstacle_number,1:4) ;
  Offset_Airport_Value_3000_R_2(obstacle_number)     = aifa_trial_4;
end
% the end

Offset_Airport_Value_3000_R_1 = (Offset_Airport_Value_3000_R_1)';
Offset_Airport_Value_3000_R_2 = (Offset_Airport_Value_3000_R_2)';



for obstacle_number = 1:ObjectNumbers_3000_R
    if Offset_Airport_Value_3000_R_2(obstacle_number) >=
Offset_Airport_Value_3000_R_1(obstacle_number)
        Offset_Airport_Value_3000_R(obstacle_number) =
Offset_Airport_Value_3000_R_1(obstacle_number);
    else Offset_Airport_Value_3000_R(obstacle_number) =
Offset_Airport_Value_3000_R_2(obstacle_number);
    end
```

```
end

Offset_Airport_Value_3000_R = (Offset_Airport_Value_3000_R)';

save OC_WQS_5Degrees_R Object_Index_3000_R Offset_Airport_Value_3000_R;

% This part is to get cumulative curve

offset_00 = 0;
offset_05 = 0;
offset_10 = 0;
offset_15 = 0;
offset_20 = 0;
offset_25 = 0;
offset_30 = 0;
offset_35 = 0;
offset_40 = 0;
offset_45 = 0;
offset_50 = 0;

for obstacle_number = 1:ObjectNumbers_3000_R
    if Offset_Airport_Value_3000_R(obstacle_number) == 0.0
        offset_00 = offset_00 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 0.5
        offset_05 = offset_05 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 1.0
        offset_10 = offset_10 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 1.5
        offset_15 = offset_15 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 2.0
        offset_20 = offset_20 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 2.5
        offset_25 = offset_25 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 3.0
        offset_30 = offset_30 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 3.5
        offset_35 = offset_35 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 4.0
        offset_40 = offset_40 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 4.5
        offset_45 = offset_45 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 5.0
        offset_50 = offset_50 + 1;
    end
end

Offsets = [offset_00 offset_05 offset_10 offset_15 offset_20 offset_25
offset_30 offset_35 offset_40 offset_45 offset_50 ];

Offset_CDF = cumsum(Offsets) / ObjectNumbers_3000_R *100;

% This program is to evaluate effecitveness of Course Offest to clear
% obstacles
% Programmed by Yue Xu, Virginia Tech, 12/12/2003, Version 1.0
% Revised by Yue Xu, 07/09/2004, Version 2.0, OCS analysis is added
```

```
% This is for OCS analysis, Base End

clear;

load object_database_3000_b;

[ObjectNumbers_3000_B,b] = size(Object_Value_3000_B);

DataNumber = 0;

% Initial Values

DA = 300;
aifa_final = 5.0;
GPA = 5.0;

for obstacle_number = 1:ObjectNumbers_3000_B

    aifa_trial_1 = 0.0;

    while aifa_trial_1 <= aifa_final

        Betta = GPA/102;

            X_0 = (DA / tan(GPA * pi / 180) - 1150) * sin(aifa_trial_1 
* pi / 180);
            Y_0 = (DA / tan(GPA * pi / 180) - 1150) * (1 - 
cos(aifa_trial_1 * pi / 180));
            dis_B_1(obstacle_number) = 
Object_Value_3000_B(obstacle_number,1) - X_0;
            off_B_1(obstacle_number) = 
Object_Value_3000_B(obstacle_number,2) - Y_0;


      if (dis_B_1(obstacle_number) >= 200) & (dis_B_1(obstacle_number) 
<= 50200)
          if abs(off_B_1(obstacle_number)) <= (9 * 
dis_B_1(obstacle_number)/250 + 1964/5)
              if (Object_Value_3000_B(obstacle_number,3) > 
((dis_B_1(obstacle_number)-200)*Betta))
                  aifa_trial_1 = aifa_trial_1 + 0.5;
              elseif aifa_trial_1 == 0.0
                  % When the first time the height is already lower 
than W surface, it is not intruding W surface
                  break;
              else
                  break;
              end
           elseif abs(off_B_1(obstacle_number)) <= 
(5376/50000*dis_B_1(obstacle_number) + 678.496)
               if (Object_Value_3000_B(obstacle_number,3) > (Betta * 
(dis_B_1(obstacle_number)-200) + 1/4 * (abs(off_B_1(obstacle_number))-
9/250*dis_B_1(obstacle_number) -1964/5)))
```

```
                    aifa_trial_1 = aifa_trial_1 + 0.5;
                elseif aifa_trial_1 == 0.0
                    % When the first time the height is already lower
than X surface, it is not intruding W surface
                   break;
                else
                    break;
                end
            elseif abs(off_B_1(obstacle_number)) <=
(7576/50000*dis_B_1(obstacle_number) + 969.696)
                if (Object_Value_3000_B(obstacle_number,3) > (Betta *
(dis_B_1(obstacle_number)-200) + 1/7 * (abs(off_B_1(obstacle_number))-
0.10752*dis_B_1(obstacle_number) -678.496)))
                    aifa_trial_1 = aifa_trial_1 + 0.5;
                elseif aifa_trial_1 == 0.0
                    % When the first time the height is already lower
than X surface, it is not intruding W surface
                   break;
                else
                    break;
                end
            else  aifa_trial_1 = 0.0;
                break;
           end
        elseif  aifa_trial_1 == 0.0;
           aifa_trial_1 = 0.0;
           break;
        else
           break;
        end
  end
  Offset_Airport_Index_3000_B_1(obstacle_number,1:4) =
Object_Index_3000_B(obstacle_number,1:4) ;
  Offset_Airport_Value_3000_B_1(obstacle_number)     = aifa_trial_1;
end

% ---------------- Offset from the below (anti Clockwise) -------------
------ %

for obstacle_number = 1:ObjectNumbers_3000_B

    aifa_trial_2 = 0.0;

    % Raising Offset Angles to clear the objects
    while aifa_trial_2 <= aifa_final

            Betta = GPA/102;
            % Position Transformation of critical obstacles to new
            % coordinate system so that WQS equation need not change
            X_0 = (DA / tan(GPA * pi / 180) - 1150) * sin(aifa_trial_2
* pi / 180);
            Y_0 = - (DA / tan(GPA * pi / 180) - 1150) * (1 -
cos(aifa_trial_2 * pi / 180));
            dis_B_2(obstacle_number) =
Object_Value_3000_B(obstacle_number,1) - X_0;
```

```
            off_B_2(obstacle_number) =
Object_Value_3000_B(obstacle_number,2) - Y_0;


      if (dis_B_2(obstacle_number) >= 200) & (dis_B_2(obstacle_number)
<= 50200)
          if abs(off_B_2(obstacle_number)) <=
(9*dis_B_2(obstacle_number)/250+1964/5)
              if (Object_Value_3000_B(obstacle_number,3) >
((dis_B_1(obstacle_number)-200)*Betta))
                  aifa_trial_2 = aifa_trial_2 + 0.5;
              elseif aifa_trial_2 == 0.0
                  % When the first time the height is already lower
than W surface, it is not intruding W surface
                  break;
              else
                  break;
              end
           elseif abs(off_B_2(obstacle_number)) <= (5376/50000 *
dis_B_2(obstacle_number) + 678.496)
               if (Object_Value_3000_B(obstacle_number,3) > (Betta *
(dis_B_2(obstacle_number)-200)+1/4 * (abs(off_B_2(obstacle_number))-
9/250 * dis_B_2(obstacle_number) -1964/5)))
                   aifa_trial_2 = aifa_trial_2 + 0.5;
               elseif aifa_trial_2 == 0.0
                   % When the first time the height is already lower
than X surface, it is not intruding W surface
                  break;
               else
                   break;
               end
           elseif abs(off_B_2(obstacle_number)) <=
(7576/50000*dis_B_2(obstacle_number) + 969.696)
               if (Object_Value_3000_B(obstacle_number,3) > (Betta *
(dis_B_2(obstacle_number)-200) + 1/7 * (abs(off_B_2(obstacle_number))-
0.10752*dis_B_2(obstacle_number) -678.496)))
                   aifa_trial_2 = aifa_trial_2 + 0.5;
               elseif aifa_trial_2 == 0.0
                   % When the first time the height is already lower
than X surface, it is not intruding W surface
                  break;
               else
                   break;
               end
           else  aifa_trial_2 = 0.0;
               break;
          end
      elseif  aifa_trial_2 == 0.0;
          aifa_trial_2 = 0.0;
          break;
      else
          break;
      end
  end
  Offset_Airport_Index_3000_B_2(obstacle_number,1:4) =
Object_Index_3000_B(obstacle_number,1:4) ;
```

```
  Offset_Airport_Value_3000_B_2(obstacle_number)     = aifa_trial_2;
end
% the end


Offset_Airport_Value_3000_B_1 = (Offset_Airport_Value_3000_B_1)';
Offset_Airport_Value_3000_B_2 = (Offset_Airport_Value_3000_B_2)';



for obstacle_number = 1:ObjectNumbers_3000_B
    if Offset_Airport_Value_3000_B_2(obstacle_number) >=
Offset_Airport_Value_3000_B_1(obstacle_number)
        Offset_Airport_Value_3000_B(obstacle_number) =
Offset_Airport_Value_3000_B_1(obstacle_number);
    else Offset_Airport_Value_3000_B(obstacle_number) =
Offset_Airport_Value_3000_B_2(obstacle_number);
    end
end

Offset_Airport_Value_3000_B = (Offset_Airport_Value_3000_B)';

save OC_OCS_5Degrees_B Object_Index_3000_B Offset_Airport_Value_3000_B;

% This part is to get cumulative curve

offset_00 = 0;
offset_05 = 0;
offset_10 = 0;
offset_15 = 0;
offset_20 = 0;
offset_25 = 0;
offset_30 = 0;
offset_35 = 0;
offset_40 = 0;
offset_45 = 0;
offset_50 = 0;


for obstacle_number = 1:ObjectNumbers_3000_B
    if Offset_Airport_Value_3000_B(obstacle_number) == 0.0
        offset_00 = offset_00 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 0.5
        offset_05 = offset_05 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 1.0
        offset_10 = offset_10 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 1.5
        offset_15 = offset_15 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 2.0
        offset_20 = offset_20 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 2.5
        offset_25 = offset_25 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 3.0
        offset_30 = offset_30 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 3.5
```

```
        offset_35 = offset_35 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 4.0
        offset_40 = offset_40 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 4.5
        offset_45 = offset_45 + 1;
    elseif Offset_Airport_Value_3000_B(obstacle_number) == 5.0
        offset_50 = offset_50 + 1;
    end
end

Offsets = [offset_00 offset_05 offset_10 offset_15 offset_20 offset_25
offset_30 offset_35 offset_40 offset_45 offset_50 ];

Offset_CDF = cumsum(Offsets) / ObjectNumbers_3000_B *100;

% subplot(2,1,1);
plot(Offset_CDF,'b','linewidth',2);
set(gca,'XTicklabel',['0.0';'0.5';'1.0';'1.5';'2.0';'2.5';'3.0';'3.5';'
4.0';'4.5';'5.0';]);

xlabel('Offset Course (Degrees).');ylabel('Percentage');
title('Percentage of Runways Clearing OCS and WQS (Base End)');
grid;


% --------------------- For the Reciprocal End------------------ %

clear all;
load object_database_3000_r;

DataNumber = 0;

% Initial Values

GPA = 5.0;
DA = 300;
aifa_final = 5.0;

[ObjectNumbers_3000_R,r] = size(Object_Value_3000_R);

for obstacle_number = 1:ObjectNumbers_3000_R

    aifa_trial_3 = 0.0;

    % Raising Offset Angles to clear the objects

    while aifa_trial_3 <= aifa_final

        Betta = GPA/102;

            X_0 = (DA / tan(GPA * pi / 180) - 1150) * sin(aifa_trial_3
* pi / 180);
```

```
            Y_0 = (DA / tan(GPA * pi / 180) - 1150) * (1 -
cos(aifa_trial_3 * pi / 180));
            dis_R_1(obstacle_number) =
Object_Value_3000_R(obstacle_number,1) - X_0;
            off_R_1(obstacle_number) =
Object_Value_3000_R(obstacle_number,2) - Y_0;


      if (dis_R_1(obstacle_number) >= 200) & (dis_R_1(obstacle_number)
<= 50200)
          if abs(off_R_1(obstacle_number)) <= (9 *
dis_R_1(obstacle_number)/250 + 1964/5)
              if (Object_Value_3000_R(obstacle_number,3) >
((dis_R_1(obstacle_number)-200)*Betta))
                  aifa_trial_3 = aifa_trial_3 + 0.5;
              elseif aifa_trial_3 == 0.0
                  % When the first time the height is already lower
than W surface, it is not intruding W surface
                  aifa_trial_3 = 0.0;
                  break;
              else
                  break;
              end
           elseif abs(off_R_1(obstacle_number)) <=
(5376/50000*dis_R_1(obstacle_number) + 678.496)
               if (Object_Value_3000_R(obstacle_number,3) > (Betta *
(dis_R_1(obstacle_number)-200) + 1/4 * (abs(off_R_1(obstacle_number))-
9/250*dis_R_1(obstacle_number) -1964/5)))
                   aifa_trial_3 = aifa_trial_3 + 0.5;
               elseif aifa_trial_3 == 0.0
                   % When the first time the height is already lower
than W surface, it is not intruding W surface
                   aifa_trial_3 = 0.0;
                  break;
               else
                   break;
               end
           elseif abs(off_R_1(obstacle_number)) <=
(7576/50000*dis_R_1(obstacle_number) + 969.696)
               if (Object_Value_3000_R(obstacle_number,3) > (Betta *
(dis_R_1(obstacle_number)-200) + 1/7 * (abs(off_R_1(obstacle_number))-
0.10752*dis_R_1(obstacle_number) -678.496)))
                   aifa_trial_3 = aifa_trial_3 + 0.5;
               elseif aifa_trial_3 == 0.0
                   % When the first time the height is already lower
than X surface, it is not intruding W surface
                  break;
               else
                   break;
               end
           else  aifa_trial_3 = 0.0;
               break;
          end
      elseif  aifa_trial_3 == 0.0;
          aifa_trial_3 = 0.0;
          break;
```

```
      else
          break;
      end
  end
  Offset_Airport_Index_3000_R_1(obstacle_number,1:4) =
Object_Index_3000_R(obstacle_number,1:4) ;
  Offset_Airport_Value_3000_R_1(obstacle_number)     = aifa_trial_3;
end


% ----------Offset from the below (anti Clockwise) -----------------%

for obstacle_number = 1:ObjectNumbers_3000_R

    aifa_trial_4 = 0.0;

    % Raising Offset Angles to clear the objects

      while aifa_trial_4 <= aifa_final

            Betta = GPA/102;

            X_0 = (DA / tan(GPA * pi / 180) - 1150) * sin(aifa_trial_4
* pi / 180);
            Y_0 = (DA / tan(GPA * pi / 180) - 1150) * (1 -
cos(aifa_trial_4 * pi / 180));
            dis_R_2(obstacle_number) =
Object_Value_3000_R(obstacle_number,1) - X_0;
            off_R_2(obstacle_number) =
Object_Value_3000_R(obstacle_number,2) - Y_0;

      if (dis_R_2(obstacle_number) >= 200) & (dis_R_2(obstacle_number)
<= 50200)
          if abs(off_R_2(obstacle_number)) <=
(9*dis_R_2(obstacle_number)/250+1964/5)
              if (Object_Value_3000_R(obstacle_number,3) >
((dis_R_1(obstacle_number)-200)*Betta))
                  aifa_trial_4 = aifa_trial_4 + 0.5;
              elseif aifa_trial_4 == 0.0
                  % When the first time the height is already lower
than W surface, it is not intruding W surface
                  break;
              else
                  break;
              end
           elseif abs(off_R_2(obstacle_number)) <= (5376/50000 *
dis_R_2(obstacle_number) + 678.496)
               if (Object_Value_3000_R(obstacle_number,3) > (Betta *
(dis_R_2(obstacle_number)-200)+1/4 * (abs(off_R_2(obstacle_number))-
9/250 * dis_R_2(obstacle_number) -1964/5)))
                   aifa_trial_4 = aifa_trial_4 + 0.5;
               elseif aifa_trial_4 == 0.0
                   % When the first time the height is already lower
than X surface, it is not intruding W surface
                  break;
```

```
                else
                    break;
                end
            elseif abs(off_R_2(obstacle_number)) <=
(7576/50000*dis_R_2(obstacle_number) + 969.696)
                if (Object_Value_3000_R(obstacle_number,3) > (Betta *
(dis_R_2(obstacle_number)-200) + 1/7 * (abs(off_R_2(obstacle_number))-
0.10752*dis_R_2(obstacle_number) -678.496)))
                    aifa_trial_4 = aifa_trial_4 + 0.5;
                elseif aifa_trial_4 == 0.0
                    % When the first time the height is already lower
than X surface, it is not intruding W surface
                   break;
                else
                    break;
                end
            else  aifa_trial_4 = 0.0;
                break;
           end
       elseif  aifa_trial_4 == 0.0;
           aifa_trial_4 = 0.0;
           break;
       else
           break;
       end
  end
  Offset_Airport_Index_3000_R_2(obstacle_number,1:4) =
Object_Index_3000_R(obstacle_number,1:4) ;
  Offset_Airport_Value_3000_R_2(obstacle_number)     = aifa_trial_4;
end
% the end

Offset_Airport_Value_3000_R_1 = (Offset_Airport_Value_3000_R_1)';
Offset_Airport_Value_3000_R_2 = (Offset_Airport_Value_3000_R_2)';



for obstacle_number = 1:ObjectNumbers_3000_R
    if Offset_Airport_Value_3000_R_2(obstacle_number) >=
Offset_Airport_Value_3000_R_1(obstacle_number)
        Offset_Airport_Value_3000_R(obstacle_number) =
Offset_Airport_Value_3000_R_1(obstacle_number);
    else Offset_Airport_Value_3000_R(obstacle_number) =
Offset_Airport_Value_3000_R_2(obstacle_number);
    end
end

Offset_Airport_Value_3000_R = (Offset_Airport_Value_3000_R)';

save OC_OCS_5Degrees_R Object_Index_3000_R Offset_Airport_Value_3000_R;

% This part is to get cumulative curve

offset_00 = 0;
offset_05 = 0;
```

```
offset_10 = 0;
offset_15 = 0;
offset_20 = 0;
offset_25 = 0;
offset_30 = 0;
offset_35 = 0;
offset_40 = 0;
offset_45 = 0;
offset_50 = 0;

for obstacle_number = 1:ObjectNumbers_3000_R
    if Offset_Airport_Value_3000_R(obstacle_number) == 0.0
        offset_00 = offset_00 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 0.5
        offset_05 = offset_05 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 1.0
        offset_10 = offset_10 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 1.5
        offset_15 = offset_15 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 2.0
        offset_20 = offset_20 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 2.5
        offset_25 = offset_25 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 3.0
        offset_30 = offset_30 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 3.5
        offset_35 = offset_35 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 4.0
        offset_40 = offset_40 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 4.5
        offset_45 = offset_45 + 1;
    elseif Offset_Airport_Value_3000_R(obstacle_number) == 5.0
        offset_50 = offset_50 + 1;
    end
end

Offsets = [offset_00 offset_05 offset_10 offset_15 offset_20 offset_25
offset_30 offset_35 offset_40 offset_45 offset_50 ];

Offset_CDF = cumsum(Offsets) / ObjectNumbers_3000_R *100;
```

Shrink Surface Analysis

```
% This program is to clear the objects by shrinking the width of the
% surface.
% Programmed by Yue Xu, Virginia Tech, ATSL, Version 1.0, 05/10/03
%                                            Version 2.0, 05/05/04
%                                            Version 3.0, 08/07/04
% Version 3.0: OCS Analysis is added


clear all;
load object_database_3000_b;


[ObjectNumbers_3000_B,b] = size(Object_Value_3000_B);  % Number of
objects for base end


DataNumber = 0;
GPA = 5.0;
DisplacedThreshold = 0;
Ratio = 0.1;
Betta = tan(2/3*pi*GPA/180);

for jj = 1:ObjectNumbers_3000_B

   Ratio_trial = 1.0;                      % From the original size
down to zero
   DataNumber = DataNumber + 1;

   while Ratio_trial >= Ratio


      if (Object_Value_3000_B(jj,1) >= 200) & (Object_Value_3000_B(jj,1)
<= 50200)  % if X axis range falls inside WQS Surface

          % ----------------- W surface Analysis ------------------ %
          if abs(Object_Value_3000_B(jj,2)) <=
(9*(Object_Value_3000_B(jj,1)-200) * Ratio_trial/250 + 400 *
Ratio_trial)

           % if Y axis range falls inside WQS Surface
              if (Object_Value_3000_B(jj,3) >
(Object_Value_3000_B(jj,1)*Betta))  % if Object cannot be cleared here
      Ratio_trial = Ratio_trial - 0.1;
% Decrease the width of the surface by 10%  %
              elseif Ratio_trial == 1.0 % When the first time the
height is already lower than W surface, it is not intruding W surface
                  break;
              else               % Get the final Ratio here and break
from the main loop
```

```
                    break;
                end

            % ------------------- X surface Analysis ------------------
 %
            elseif abs(Object_Value_3000_B(jj,2)) <= (5376/50000*
Ratio_trial *(Object_Value_3000_B(jj,1)-200) + 700 * Ratio_trial)
                if (Object_Value_3000_B(jj,3) > (Betta *
Object_Value_3000_B(jj,1)+1/4 * (abs(Object_Value_3000_B(jj,2))...
                        -9/250 * Ratio_trial *
(Object_Value_3000_B(jj,1)-200) - 400 * Ratio_trial)))
                    Ratio_trial = Ratio_trial -
0.1;                                % Decrease the width of the surface
by 10%  %
                elseif Ratio_trial == 1.0 % When the first time the
height is already lower than W surface, it is not intruding W surface
                    break;
                else
                     break;                 % Get the final Ratio here and
break from the main loop
                end
            else
                break;                      % Object falls out of WQS
surface
           end
      else  Ratio_trial = 1.0;              % Object falls out of WQS
surface
           break;
      end
  end
  Shrinking_Airport_Index_3000_B(DataNumber,1:4) =
Object_Index_3000_B(jj,1:4) ;
  Shrinking_Airport_Value_3000_B(DataNumber)     = Ratio_trial;
end


Shrinking_Airport_Value_3000_B = (Shrinking_Airport_Value_3000_B)';
save SS_WQS_5Degrees_B Shrinking_Airport_Index_3000_B
Shrinking_Airport_Value_3000_B;

% This part is to get the Cumulative Distribution Function

SF_0 = 0;
SF_1 = 0;
SF_2 = 0;
SF_3 = 0;
SF_4 = 0;
SF_5 = 0;
SF_6 = 0;
SF_7 = 0;
SF_8 = 0;
SF_9 = 0;
SF_10 = 0;

for jj = 1:DataNumber
    if (Shrinking_Airport_Value_3000_B(jj) > 0.05)
```

```
        if (Shrinking_Airport_Value_3000_B(jj) < 0.15)
            SF_1 = SF_1 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.25)
            SF_2 = SF_2 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.35)
            SF_3 = SF_3 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.45)
            SF_4 = SF_4 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.55)
            SF_5 = SF_5 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.65)
            SF_6 = SF_6 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.75)
            SF_7 = SF_7 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.85)
            SF_8 = SF_8 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.95)
            SF_9 = SF_9 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 1.1)
            SF_10 = SF_10 + 1;
        end
    else SF_0 = SF_0 + 1;
    end

end

SFs = [SF_10 SF_9 SF_8 SF_7 SF_6 SF_5 SF_4 SF_3 SF_2 SF_1 SF_0];


SF_CDF(1) = SFs(1);
SF_CDF(2) = SFs(2)+SFs(1);
for pp = 1:10
    SF_CDF(pp+2)=SF_CDF(pp+1) + SFs(pp+1);
end

% subplot(2,1,1);
plot(SF_CDF(1:11)/SF_CDF(12)*100,'linewidth',2,'color','r');
set(gca,'XTicklabel',['1.0';'0.9';'0.8';'0.7';'0.6';'0.5';'0.4';'0.3';'
0.2';'0.1';'0.0';]);

xlabel('Shrinking Ratio');ylabel('Percentage'); title('Percentage of
Runways Clearing OCS and WQS (Base End)');
grid;

hold on;


% ---- for reciprocal end information ---- %

clear all;

load object_database_3000_r;
```

```
[ObjectNumbers_3000_R,r] = size(Object_Value_3000_R);  % Number of
objects for base end

DataNumber = 0;
GPA = 5.0;
Ratio = 0.1;
Betta = tan(2/3*pi*GPA/180);

for jj = 1:ObjectNumbers_3000_R
   Ratio_trial = 1.0;                         % From the original size
down to zero
   DataNumber = DataNumber + 1;
   while Ratio_trial >= Ratio
      if (Object_Value_3000_R(jj,1) >= 200) & (Object_Value_3000_R(jj,1)
<= 50200)  % if X axis range falls inside WQS Surface

          % ----------------- W surface Analysis ------------------ %
          if abs(Object_Value_3000_R(jj,2)) <=
(9*(Object_Value_3000_R(jj,1)-200) * Ratio_trial/250 + 400 *
Ratio_trial)

           % if Y axis range falls inside WQS Surface
              if (Object_Value_3000_R(jj,3) >
(Object_Value_3000_R(jj,1)*Betta))  % if Object cannot be cleared here
                  Ratio_trial = Ratio_trial -
0.1;                               % Decrease the width of the surface
by 10%  %
              elseif Ratio_trial == 1.0 % When the first time the
height is already lower than W surface, it is not intruding W surface
                  break;
              else                 % Get the final Ratio here and break
from the main loop
                  break;
              end

           % ------------------- X surface Analysis ------------------
 %
           elseif abs(Object_Value_3000_R(jj,2)) <= (5376/50000*
Ratio_trial *(Object_Value_3000_R(jj,1)-200) + 700 * Ratio_trial)
               if (Object_Value_3000_R(jj,3) > (Betta *
Object_Value_3000_R(jj,1)+1/4 * (abs(Object_Value_3000_R(jj,2))...
                       -9/250 * Ratio_trial *
(Object_Value_3000_R(jj,1)-200) - 400 * Ratio_trial)))
                   Ratio_trial = Ratio_trial -
0.1;                              % Decrease the width of the surface
by 10%  %
               elseif Ratio_trial == 1.0 % When the first time the
height is already lower than W surface, it is not intruding W surface
                  break;
               else
                   break;                 % Get the final Ratio here and
break from the main loop
               end
           else
```

```
               break;                    % Object falls out of WQS
surface
          end
      else  Ratio_trial = 1.0;           % Object falls out of WQS
surface
          break;
      end
  end
  Shrinking_Airport_Index_3000_R(DataNumber,1:4) =
Object_Index_3000_R(jj,1:4) ;
  Shrinking_Airport_Value_3000_R(DataNumber)     = Ratio_trial;
end


Shrinking_Airport_Value_3000_R = (Shrinking_Airport_Value_3000_R)';
save SS_WQS_5Degrees_R Shrinking_Airport_Index_3000_R
Shrinking_Airport_Value_3000_R;

% This part is to get the Cumulative Distribution Function
SF_0 = 0;
SF_1 = 0;
SF_2 = 0;
SF_3 = 0;
SF_4 = 0;
SF_5 = 0;
SF_6 = 0;
SF_7 = 0;
SF_8 = 0;
SF_9 = 0;
SF_10 = 0;


for jj = 1:DataNumber

    if (Shrinking_Airport_Value_3000_R(jj) > 0.05)

        if (Shrinking_Airport_Value_3000_R(jj) < 0.15)
            SF_1 = SF_1 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.25)
            SF_2 = SF_2 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.35)
            SF_3 = SF_3 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.45)
            SF_4 = SF_4 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.55)
            SF_5 = SF_5 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.65)
            SF_6 = SF_6 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.75)
            SF_7 = SF_7 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.85)
            SF_8 = SF_8 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.95)
            SF_9 = SF_9 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 1.1)
```

```
            SF_10 = SF_10 + 1;
        end
    else SF_0 = SF_0 + 1;
    end

end

SFs = [SF_10 SF_9 SF_8 SF_7 SF_6 SF_5 SF_4 SF_3 SF_2 SF_1 SF_0];


SF_CDF(1) = SFs(1);
SF_CDF(2) = SFs(2)+SFs(1);
for pp = 1:10
    SF_CDF(pp+2)=SF_CDF(pp+1) + SFs(pp+1);
end


% This program is to clear the objects of OCS Surface by shrinking the
width of the surface.
% Programmed by Yue Xu, Virginia Tech, ATSL, Version 1.0, 05/10/03
%                                            Version 2.0, 05/05/04
%                                            Version 3.0, 08/07/04
% Version 3.0: OCS Analysis is added

clear all;
load object_database_3000_b;

[ObjectNumbers_3000_B,b] = size(Object_Value_3000_B);  % Number of
objects for base end

DataNumber = 0;
GPA = 5.0;
DisplacedThreshold = 0;
Ratio = 0.1;
Betta = GPA/102;

for jj = 1:ObjectNumbers_3000_B

   Ratio_trial = 1.0;                         % From the original size
down to zero
   DataNumber = DataNumber + 1;

   while Ratio_trial >= Ratio

       if (Object_Value_3000_B(jj,1) >= 200) &
(Object_Value_3000_B(jj,1) <= 50200)  % if X axis range falls inside
WQS Surface

          % ----------------- W surface Analysis ------------------ %
           if abs(Object_Value_3000_B(jj,2)) <= (9/250 *
(Object_Value_3000_B(jj,1)-200) * Ratio_trial + 400 * Ratio_trial)

           % if Y axis range falls inside WQS Surface
```

```
              if (Object_Value_3000_B(jj,3) >
((Object_Value_3000_B(jj,1)-200)*Betta))  % if Object cannot be cleared
here
                  Ratio_trial = Ratio_trial -
0.1;                                % Decrease the width of the surface
by 10%  %
              elseif Ratio_trial == 1.0 % When the first time the
height is already lower than W surface, it is not intruding W surface
                  break;
              else               % Get the final Ratio here and break
from the main loop
                  break;
              end

           % ------------------- X surface Analysis ------------------
 %

            elseif abs(Object_Value_3000_B(jj,2)) <= (5376/50000*
      Ratio_trial *(Object_Value_3000_B(jj,1)-200) + 700 * Ratio_trial)

      if (Object_Value_3000_B(jj,3) > (Betta *
(Object_Value_3000_B(jj,1)-200) + 1/4 *
(abs(Object_Value_3000_B(jj,2))-9/250 * Ratio_trial *
(Object_Value_3000_B(jj,1)-200) - 400 * Ratio_trial)))

Ratio_trial = Ratio_trial - 0.1;

% Decrease the width of the surface by 10%  %

      elseif Ratio_trial == 1.0 % When the first time the height is
already lower than W surface, it is not intruding W surface

                  break;
               else
                   break;                   % Get the final Ratio here and
break from the main loop
               end
           % ------------------- Y surface Analysis ------------------
 %

           elseif abs(Object_Value_3000_B(jj,2)) <= (7576/50000*
Ratio_trial *(Object_Value_3000_B(jj,1)-200) + 1000 * Ratio_trial)
               if (Object_Value_3000_B(jj,3) > (Betta *
(Object_Value_3000_B(jj,1)-200) + 1/7 *
(abs(Object_Value_3000_B(jj,2))...
                       -5376/50000 * Ratio_trial *
(Object_Value_3000_B(jj,1)-200) - 700 * Ratio_trial)))
                   Ratio_trial = Ratio_trial -
0.1;                               % Decrease the width of the surface
by 10%  %
               elseif Ratio_trial == 1.0 % When the first time the
height is already lower than W surface, it is not intruding W surface
                  break;
               else
                   break;                   % Get the final Ratio here and
break from the main loop
```

```
                end
            else
                break;                     % Object falls out of WQS
surface
            end
       else  Ratio_trial = 1.0;            % Object falls out of WQS
surface
           break;
       end
  end
  Shrinking_Airport_Index_3000_B(DataNumber,1:4) =
Object_Index_3000_B(jj,1:4) ;
  Shrinking_Airport_Value_3000_B(DataNumber)     = Ratio_trial;
end

Shrinking_Airport_Value_3000_B = (Shrinking_Airport_Value_3000_B)';
save SS_OCS_5Degrees_B Shrinking_Airport_Index_3000_B
Shrinking_Airport_Value_3000_B;

% This part is to get the Cumulative Distribution Function

SF_0 = 0;
SF_1 = 0;
SF_2 = 0;
SF_3 = 0;
SF_4 = 0;
SF_5 = 0;
SF_6 = 0;
SF_7 = 0;
SF_8 = 0;
SF_9 = 0;
SF_10 = 0;

for jj = 1:DataNumber
    if (Shrinking_Airport_Value_3000_B(jj) > 0.05)

        if (Shrinking_Airport_Value_3000_B(jj) < 0.15)
            SF_1 = SF_1 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.25)
            SF_2 = SF_2 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.35)
            SF_3 = SF_3 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.45)
            SF_4 = SF_4 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.55)
            SF_5 = SF_5 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.65)
            SF_6 = SF_6 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.75)
            SF_7 = SF_7 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.85)
            SF_8 = SF_8 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 0.95)
            SF_9 = SF_9 + 1;
        elseif (Shrinking_Airport_Value_3000_B(jj) < 1.1)
            SF_10 = SF_10 + 1;
```

```
        end
    else SF_0 = SF_0 + 1;
    end

end

SFs = [SF_10 SF_9 SF_8 SF_7 SF_6 SF_5 SF_4 SF_3 SF_2 SF_1 SF_0];


SF_CDF(1) = SFs(1);
SF_CDF(2) = SFs(2)+SFs(1);
for pp = 1:10
    SF_CDF(pp+2)=SF_CDF(pp+1) + SFs(pp+1);
end


plot(SF_CDF(1:11)/SF_CDF(12)*100,'linewidth',2,'color','b');
set(gca,'XTicklabel',['1.0';'0.9';'0.8';'0.7';'0.6';'0.5';'0.4';'0.3';'
0.2';'0.1';'0.0';]);

xlabel('Shrinking Ratio');ylabel('Percentage'); title('Percentage of
Runways Clearing OCS and WQS (Base End)');
grid;

hold on;


% ---- for reciprocal end information ---- %

clear all;

load object_database_3000_r;


[ObjectNumbers_3000_R,r] = size(Object_Value_3000_R);  % Number of
objects for base end

DataNumber = 0;
GPA = 5.0;
Ratio = 0.1;
Betta = GPA/102;

for jj = 1:ObjectNumbers_3000_R
   Ratio_trial = 1.0;                        % From the original size
down to zero
   DataNumber = DataNumber + 1;
   while Ratio_trial >= Ratio
      if (Object_Value_3000_R(jj,1) >= 200) & (Object_Value_3000_R(jj,1)
<= 50200)  % if X axis range falls inside WQS Surface

          % ----------------- W surface Analysis ------------------ %
```

```
            if abs(Object_Value_3000_R(jj,2)) <=
(9*(Object_Value_3000_R(jj,1)-200) * Ratio_trial/250 + 400 *
Ratio_trial)

            % if Y axis range falls inside WQS Surface
               if (Object_Value_3000_R(jj,3) >
((Object_Value_3000_R(jj,1)-200)*Betta))  % if Object cannot be cleared
here
                   Ratio_trial = Ratio_trial -
0.1;                                   % Decrease the width of the surface
by 10%  %
               elseif Ratio_trial == 1.0 % When the first time the
height is already lower than W surface, it is not intruding W surface
                   break;
                else                   % Get the final Ratio here and break
from the main loop
                   break;
               end

            % ------------------- X surface Analysis ------------------
 %
            elseif abs(Object_Value_3000_R(jj,2)) <= (5376/50000*
Ratio_trial *(Object_Value_3000_R(jj,1)-200) + 700 * Ratio_trial)
                if (Object_Value_3000_R(jj,3) > (Betta *
(Object_Value_3000_R(jj,1)-200)+1/4  *  (abs(Object_Value_3000_R(jj,2))...
                       -9/250 * Ratio_trial *
(Object_Value_3000_R(jj,1)-200) - 400 * Ratio_trial)))
                    Ratio_trial = Ratio_trial -
0.1;                                  % Decrease the width of the surface
by 10%  %
                elseif Ratio_trial == 1.0 % When the first time the
height is already lower than W surface, it is not intruding W surface
                   break;
                else
                    break;                 % Get the final Ratio here and
break from the main loop
                end
            % ------------------- Y surface Analysis ------------------
 %
            elseif abs(Object_Value_3000_R(jj,2)) <= (7576/50000*
Ratio_trial *(Object_Value_3000_R(jj,1)-200) + 1000 * Ratio_trial)
                if (Object_Value_3000_R(jj,3) > (Betta *
(Object_Value_3000_R(jj,1)-200)+1/7  *  (abs(Object_Value_3000_R(jj,2))...
                       -5376/50000 * Ratio_trial *
(Object_Value_3000_R(jj,1)-200) - 700 * Ratio_trial)))
                    Ratio_trial = Ratio_trial -
0.1;                                  % Decrease the width of the surface
by 10%  %
                elseif Ratio_trial == 1.0 % When the first time the
height is already lower than W surface, it is not intruding W surface
                   break;
                else
                    break;                 % Get the final Ratio here and
break from the main loop
                end
            else
```

```
               break;                    % Object falls out of WQS
surface
          end
      else  Ratio_trial = 1.0;           % Object falls out of WQS
surface
          break;
      end
  end
  Shrinking_Airport_Index_3000_R(DataNumber,1:4) =
Object_Index_3000_R(jj,1:4) ;
  Shrinking_Airport_Value_3000_R(DataNumber)     = Ratio_trial;
end


Shrinking_Airport_Value_3000_R = (Shrinking_Airport_Value_3000_R)';
save SS_OCS_5Degrees_R Shrinking_Airport_Index_3000_R
Shrinking_Airport_Value_3000_R;

% This part is to get the Cumulative Distribution Function
SF_0 = 0;
SF_1 = 0;
SF_2 = 0;
SF_3 = 0;
SF_4 = 0;
SF_5 = 0;
SF_6 = 0;
SF_7 = 0;
SF_8 = 0;
SF_9 = 0;
SF_10 = 0;


for jj = 1:DataNumber

    if (Shrinking_Airport_Value_3000_R(jj) > 0.05)

        if (Shrinking_Airport_Value_3000_R(jj) < 0.15)
            SF_1 = SF_1 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.25)
            SF_2 = SF_2 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.35)
            SF_3 = SF_3 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.45)
            SF_4 = SF_4 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.55)
            SF_5 = SF_5 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.65)
            SF_6 = SF_6 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.75)
            SF_7 = SF_7 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.85)
            SF_8 = SF_8 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 0.95)
            SF_9 = SF_9 + 1;
        elseif (Shrinking_Airport_Value_3000_R(jj) < 1.1)
```

```
            SF_10 = SF_10 + 1;
        end
    else SF_0 = SF_0 + 1;
    end

end

SFs = [SF_10 SF_9 SF_8 SF_7 SF_6 SF_5 SF_4 SF_3 SF_2 SF_1 SF_0];


SF_CDF(1) = SFs(1);
SF_CDF(2) = SFs(2)+SFs(1);
for pp = 1:10
    SF_CDF(pp+2)=SF_CDF(pp+1) + SFs(pp+1);
end

subplot(2,1,2);
plot(SF_CDF(1:11)/SF_CDF(12)*100,'linewidth',2,'color','b');
set(gca,'XTicklabel',['1.0';'0.9';'0.8';'0.7';'0.6';'0.5';'0.4';'0.3';'
0.2';'0.1';'0.0';]);

xlabel('Shrinking Ratio');ylabel('Percentage of Runways'); title('Base
End');
grid;

hold on;

xlabel('Shrinking Ratio');ylabel('Percentage of Runways');
title('Reciprocal End');


hold on;
```

DataComm Analysis

Poisson Demand Generator

```
%  Scrip to demonstrate how to generate random numbers according to a
given
%  distribution

% Date: 11/5/07
% Trani

clc
clear

% ----------- Do it for a negative exponential distributed random
variate -------
% ------------Equivalent to Poisson Arrivals

% Generates a neg. exponential random variate
% define mean of distribution
% beta = mean of distribution
% f(t) = 1/beta * exp(-t / beta)

File_Location = 'D:\Research\NewYork Metroplex\Flight Generator\';

lambda = 34;        % operations /hr

arrivals = 24 * lambda;

% arrivals = 1000;             % No. of points used in the generation
process
interarrivalTime = 3600 / lambda;                            % mean
inter-arrival time (seconds)
u=rand(1,arrivals);                  % random number generator (0-1)
timeBetweenArrivals_negExpon = - interarrivalTime * log(u);         %
Neg. exponential random variate


% figure
% hist(timeBetweenArrivals_negExpon)
% grid
% xlabel('Time between Arrivals (seconds)')
% ylabel('Frequency')
%
% % calculate metrics
%
% meanIAT_NegExpon = mean(timeBetweenArrivals_negExpon);
% stdIAT_NegExpon = std(timeBetweenArrivals_negExpon);
% disp(['Expected Time Between Arrivals =  ', num2str(meanIAT_NegExpon),
'   seconds'])
```

```
% disp(['Std. Deviation of Time Between Arrivals =  ',
num2str(stdIAT_NegExpon), '   seconds'])

% create a vector with injection times

timeToInject(1) = 0;

for i=2:arrivals
    timeToInject(i) = timeToInject(i-1) +
timeBetweenArrivals_negExpon(i-1);
end

% ----------- Do it for a normal distributed random variate -------

interarrivalTime = 3600 / lambda;                            % mean
inter-arrival time (seconds)
interarrivalTimeStd = 20;                                    % std of
interarrival time (seconds)

 timeBetweenArrivals_Normal = interarrivalTime + randn(1,arrivals-1) *
interarrivalTimeStd;

%  figure
%  hist(timeBetweenArrivals_Normal)
%  grid
%  xlabel('Time between Arrivals (seconds)')
%  ylabel('Frequency')

 % calculate metrics

% meanIAT_Normal = mean(timeBetweenArrivals_Normal);
% stdIAT_Normal = std(timeBetweenArrivals_Normal);
% disp(['Expected Time Between Arrivals =  ', num2str(meanIAT_Normal),
'   seconds'])
% disp(['Std. Deviation of Time Between Arrivals =  ',
num2str(stdIAT_Normal), '   seconds'])

% -  ------------------- declare aircraft mix -----------

percentMix = [ 0 20 30 30];                        % percent

nl = length(percentMix);

% Probability of i aircraft following j

  for n=1:nl
      for m=1:nl
            Pij(n,m) = percentMix(n)/100 *percentMix(m)/100;
      end
  end


  % ---------- Assign aircraft generated to groups -------
 %LGA Aircraft mix
```

```
LGA_Model_Mix_RAMS    = ['A319';  'A320';  'B717';  'B737';  'B757';
'CL60';  'CRJ1';  'DH8A';  'E145';  'JS31'];
LGA_Model_Mix_SIMMOD  = ['A320  ';'A320  ';'DC930 ';'737D17';'757RR
';'CL600 ';'CL601 ';'DHC8  ';'CL601 ';'BAE300'];


LGA_Percentage_Mix = [16 5 4 12 8 1 12 8 30 2];

%JFK Aircraft Mix

JFK_Model_Mix_RAMS   = ['A300';  'A320';  'A330';  'A340';  'B717';
'B737';  'B747';  'B757';  'B767';  'B777';  'BE20';  'CRJ1';  'E145';
'MD83'];
JFK_Model_Mix_SIMMOD = ['A300  ';'A320  ';'A300  ';'A300  ';'DC930
';'737D17';'747200';'757RR ';'767JT9';'767JT9';'BAE300';'CL601 ';'CL601
';'MD82  '];

JFK_Percentage_Mix = [4 27 2 1 3 3 7 16 10 4 7 7 8 2];

%EWR Aircraft Mix

EWR_Model_Mix_RAMS   = ['A320';  'B717';  'B727';  'B737';  'B757';
'B767';  'B777';  'CRJ1';  'DC10';  'E145';  'MD83';  'PA34'];
EWR_Model_Mix_SIMMOD = ['A320  ';'DC930 ';'727D17';'737D17';'757RR
';'767JT9';'767JT9';'CL601 ';'DC1030';'CL601 ';'MD82  ';'BEC58P'];

EWR_Percentage_Mix = [6 2 1 29  12  4 2 2 2 34 5 2];

comma = ',';

% ------------------------ Aircraft mix input end ---------------------
----- %


randomNumber =
round(rand(1,arrivals)*100);                            % generate
random number for arrivals


% Generate outputs for LGA

fid = fopen([File_Location, 'SIMMOD_NYC_LGA_ARR_', num2str(lambda),
'.txt'], 'w');  % write output to a text flie

for i=1:1:arrivals

     LGA_TimetoInject(i) = timeToInject(i)/3600; % Time in decimal
     LGA_Flight_Sequence(i) = i;                 % flight sequence
number
     LGA_Airline(i,1:3)     = 'AAA';             % all airlines are
named AAA
     LGA_Approach_Path(i,1:12) = 'LGA_APP_N_22'; % Approach path name
```

```
    if randomNumber(i) >= sum(LGA_Percentage_Mix(1:9))
        LGA_aircraft(i,1:6) = LGA_Model_Mix_SIMMOD(10,1:6);

    elseif randomNumber(i) >= sum(LGA_Percentage_Mix(1:8))
        LGA_aircraft(i,1:6) = LGA_Model_Mix_SIMMOD(9,1:6);

     elseif randomNumber(i) >= sum(LGA_Percentage_Mix(1:7))
        LGA_aircraft(i,1:6) = LGA_Model_Mix_SIMMOD(8,1:6);

     elseif randomNumber(i) >= sum(LGA_Percentage_Mix(1:6))
        LGA_aircraft(i,1:6) = LGA_Model_Mix_SIMMOD(7,1:6);

     elseif randomNumber(i) >= sum(LGA_Percentage_Mix(1:5))
        LGA_aircraft(i,1:6) = LGA_Model_Mix_SIMMOD(6,1:6);

     elseif randomNumber(i) >= sum(LGA_Percentage_Mix(1:4))
        LGA_aircraft(i,1:6) = LGA_Model_Mix_SIMMOD(5,1:6);

     elseif randomNumber(i) >= sum(LGA_Percentage_Mix(1:3))
        LGA_aircraft(i,1:6) = LGA_Model_Mix_SIMMOD(4,1:6);

     elseif randomNumber(i) >= sum(LGA_Percentage_Mix(1:2))
        LGA_aircraft(i,1:6) = LGA_Model_Mix_SIMMOD(3,1:6);

    elseif randomNumber(i) > LGA_Percentage_Mix(1)
        LGA_aircraft(i,1:6) = LGA_Model_Mix_SIMMOD(2,1:6);

    else
        LGA_aircraft(i,1:6) = LGA_Model_Mix_SIMMOD(1,1:6);

    end

    fprintf(fid,
'%s%s%f%s%s%s%s%s%s%s%s%s%d%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s\n',
[LGA_Airline(i,1:3),num2str(i)],comma,LGA_TimetoInject(i),  comma,  comma,
comma, comma, 'A', comma, LGA_Airline(i,1:3),comma,...
        i,comma, LGA_aircraft(i,1:6), comma, LGA_Approach_Path(i,1:12),
comma, comma, comma, comma, 'LateNess_Flight', comma, comma, 'ADES',
comma, 'LGA', comma, comma, comma, comma);


end

fclose(fid);


% Generate outputs for LGA

fid = fopen([File_Location, 'SIMMOD_NYC_JFK_ARR_', num2str(lambda),
'.txt'], 'w'); % write output to a text flie
```

```
for i=1:1:arrivals

    JFK_TimetoInject(i) = timeToInject(i)/3600; % Time in decimal
    JFK_Flight_Sequence(i) = i;                 % flight sequence
number
    JFK_Airline(i,1:3)     = 'AAA';             % all airlines are
named AAA
    JFK_Approach_Path(i,1:14) = 'JFK_APP_NE_31R'; % Approach path name

    if randomNumber(i) >= sum(JFK_Percentage_Mix(1:13))
         JFK_aircraft(i,1:6) = JFK_Model_Mix_SIMMOD(14,1:6);

    elseif randomNumber(i) >= sum(JFK_Percentage_Mix(1:12))
         JFK_aircraft(i,1:6) = JFK_Model_Mix_SIMMOD(13,1:6);

    elseif randomNumber(i) >= sum(JFK_Percentage_Mix(1:11))
         JFK_aircraft(i,1:6) = JFK_Model_Mix_SIMMOD(12,1:6);

    elseif randomNumber(i) >= sum(JFK_Percentage_Mix(1:10))
         JFK_aircraft(i,1:6) = JFK_Model_Mix_SIMMOD(11,1:6);

    elseif randomNumber(i) >= sum(JFK_Percentage_Mix(1:9))
         JFK_aircraft(i,1:6) = JFK_Model_Mix_SIMMOD(10,1:6);

     elseif randomNumber(i) >= sum(JFK_Percentage_Mix(1:8))
         JFK_aircraft(i,1:6) = JFK_Model_Mix_SIMMOD(9,1:6);

      elseif randomNumber(i) >= sum(JFK_Percentage_Mix(1:7))
         JFK_aircraft(i,1:6) = JFK_Model_Mix_SIMMOD(8,1:6);

      elseif randomNumber(i) >= sum(JFK_Percentage_Mix(1:6))
         JFK_aircraft(i,1:6) = JFK_Model_Mix_SIMMOD(7,1:6);

      elseif randomNumber(i) >= sum(JFK_Percentage_Mix(1:5))
         JFK_aircraft(i,1:6) = JFK_Model_Mix_SIMMOD(6,1:6);

      elseif randomNumber(i) >= sum(JFK_Percentage_Mix(1:4))
         JFK_aircraft(i,1:6) = JFK_Model_Mix_SIMMOD(5,1:6);

      elseif randomNumber(i) >= sum(JFK_Percentage_Mix(1:3))
         JFK_aircraft(i,1:6) = JFK_Model_Mix_SIMMOD(4,1:6);

      elseif randomNumber(i) >= sum(JFK_Percentage_Mix(1:2))
         JFK_aircraft(i,1:6) = JFK_Model_Mix_SIMMOD(3,1:6);

     elseif randomNumber(i) > JFK_Percentage_Mix(1)
         JFK_aircraft(i,1:6) = JFK_Model_Mix_SIMMOD(2,1:6);

     else
         JFK_aircraft(i,1:6) = JFK_Model_Mix_SIMMOD(1,1:6);

    end
```

```
     fprintf(fid,
'%s%s%f%s%s%s%s%s%s%s%s%d%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s\n',
[JFK_Airline(i,1:3),num2str(i)],comma,JFK_TimetoInject(i),  comma,  comma,
comma, comma, 'A', comma, JFK_Airline(i,1:3),comma,...
         i,comma,  JFK_aircraft(i,1:6),  comma,  JFK_Approach_Path(i,1:14),
comma, comma, comma, comma, 'LateNess_Flight', comma, comma, 'ADES',
comma, 'JFK', comma, comma, comma, comma);

end

fclose(fid);


% Generate outputs for EWR

fid = fopen([File_Location, 'SIMMOD_NYC_EWR_ARR_', num2str(lambda),
'.txt'], 'w'); % write output to a text flie

for i=1:1:arrivals

    EWR_TimetoInject(i) = timeToInject(i)/3600;  % Time in decimal
    EWR_Flight_Sequence(i) = i;                     % flight sequence
number
    EWR_Airline(i,1:3)     = 'AAA';                 % all airlines are
named AAA
    EWR_Approach_Path(i,1:13) = 'EWR_APP_N_22L'; % Approach path name

    if randomNumber(i) >= sum(EWR_Percentage_Mix(1:11))
         EWR_aircraft(i,1:6) = EWR_Model_Mix_SIMMOD(12,1:6);

     elseif randomNumber(i) >= sum(EWR_Percentage_Mix(1:10))
         EWR_aircraft(i,1:6) = EWR_Model_Mix_SIMMOD(11,1:6);

    elseif randomNumber(i) >= sum(EWR_Percentage_Mix(1:9))
         EWR_aircraft(i,1:6) = EWR_Model_Mix_SIMMOD(10,1:6);

    elseif randomNumber(i) >= sum(EWR_Percentage_Mix(1:8))
         EWR_aircraft(i,1:6) = EWR_Model_Mix_SIMMOD(9,1:6);

     elseif randomNumber(i) >= sum(EWR_Percentage_Mix(1:7))
         EWR_aircraft(i,1:6) = EWR_Model_Mix_SIMMOD(8,1:6);

      elseif randomNumber(i) >= sum(EWR_Percentage_Mix(1:6))
         EWR_aircraft(i,1:6) = EWR_Model_Mix_SIMMOD(7,1:6);

      elseif randomNumber(i) >= sum(EWR_Percentage_Mix(1:5))
         EWR_aircraft(i,1:6) = EWR_Model_Mix_SIMMOD(6,1:6);

      elseif randomNumber(i) >= sum(EWR_Percentage_Mix(1:4))
         EWR_aircraft(i,1:6) = EWR_Model_Mix_SIMMOD(5,1:6);

      elseif randomNumber(i) >= sum(EWR_Percentage_Mix(1:3))
         EWR_aircraft(i,1:6) = EWR_Model_Mix_SIMMOD(4,1:6);
```

```
        elseif randomNumber(i) >= sum(EWR_Percentage_Mix(1:2))
            EWR_aircraft(i,1:6) = EWR_Model_Mix_SIMMOD(3,1:6);

       elseif randomNumber(i) > EWR_Percentage_Mix(1)
            EWR_aircraft(i,1:6) = EWR_Model_Mix_SIMMOD(2,1:6);

       else
            EWR_aircraft(i,1:6) = EWR_Model_Mix_SIMMOD(1,1:6);

      end

       fprintf(fid,
'%s%s%f%s%s%s%s%s%s%s%s%d%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s\n',
[EWR_Airline(i,1:3),num2str(i)],comma,EWR_TimetoInject(i),  comma,  comma,
comma, comma, 'A', comma, EWR_Airline(i,1:3),comma,...
         i,comma,  EWR_aircraft(i,1:6),  comma,  EWR_Approach_Path(i,1:13),
comma, comma, comma, comma, 'LateNess_Flight', comma, comma, 'ADES',
comma, 'EWR', comma, comma, comma, comma);


end

fclose(fid);
```

Output Processor

```
% Generate delay curves by time
% Generate demand files by time
% programed by Yue Xu
% Version 1.0, 12/17/07

clear;

clc

Year = [2006 2014 2018];

Directory =
'C:\Simmod_PLUS\applications\NewYork_Metroplex_121407_Q2_Peak_';

Entry_sep = [7 8 9];

Airport_ID = ['JFK';'EWR';'LGA'];

end_sim = 28;

Iteration = 5;


% load data, calculate delay and # of operations in each hour.

disp('process data step 1');

for year_number = 1:length(Year)

    disp(['Year ',num2str(Year(year_number))]);

    for Entry_number = 1:length(Entry_sep)

        for Iteration_number = 1:Iteration

           disp(['Entry ', num2str(Entry_sep(Entry_number)),'nm
','Iteration ',num2str(Iteration_number)]);

            [c1, c2, acid, start_time, c5, c6, c7, c8, c9, c10, c11,
c12, end_time]  =
textread([Directory,num2str(Year(year_number)),'_','Entry',...

num2str(Entry_sep(Entry_number)),'\Reporter','\simu26_standard_',num2st
r(Iteration_number),'.csv'],'%s%s%s%f%s%s%s%s%s%s%s%s%f','delimiter',',
');
            [d1, d2, acid_delay, d4, d5, d6, d7, d8, d9, d10, d11, d12,
delay]   = textread([Directory,num2str(Year(year_number)),'_','Entry',...

num2str(Entry_sep(Entry_number)),'\Reporter','\simu26_standard_delayed_
```

```
total_',num2str(Iteration_number),'.csv'],'%s%s%s%s%s%s%s%s%s%s%s%s%s%f',
'delimiter',',');
            [e1, e2, acid_ops, operation_type, e5, airport_id, e7, e8,
e9, e10, e11, e12, e13, e14]  =
textread([Directory,num2str(Year(year_number)),'_','Entry',...

num2str(Entry_sep(Entry_number)),'\Reporter','\simu26_standard_header_'
,num2str(Iteration_number),'.csv'],'%s%s%s%s%s%s%s%s%s%s%s%s%s%s','deli
miter',',');


eval(['acid','_',num2str(Year(year_number)),'_Entry',num2str(Entry_sep(
Entry_number)),'nm','_Iteration',num2str(Iteration_number),'=
acid;']); % acid_****_Entry*nm = acid

eval(['start_time','_',num2str(Year(year_number)),'_Entry',num2str(Entr
y_sep(Entry_number)),'nm','_Iteration',num2str(Iteration_number),'=
start_time;']); % start_time_****_Entry*nm = start_time

eval(['end_time','_',num2str(Year(year_number)),'_Entry',num2str(Entry_
sep(Entry_number)),'nm','= end_time;']); % end_time_****_Entry*nm =
end_time

eval(['delay','_',num2str(Year(year_number)),'_Entry',num2str(Entry_sep
(Entry_number)),'nm','= delay;']); % % delay_****_Entry*nm = delay

eval(['operation_type','_',num2str(Year(year_number)),'_Entry',num2str(
Entry_sep(Entry_number)),'nm','= operation_type;']); % %
delay_****_Entry*nm = delay

            for hour = 1:end_sim

                for airport_number = 1:length(Airport_ID)

                    % Initialization

eval(['flights','_',num2str(Year(year_number)),'_Entry',num2str(Entry_s
ep(Entry_number)),'nm','_',num2str(hour),'_',Airport_ID(airport_number,
1:3),'=0;']);

eval(['demand','_',num2str(Year(year_number)),'_Entry',num2str(Entry_se
p(Entry_number)),'nm','_',num2str(hour),'_',...

Airport_ID(airport_number,1:3),'_Iteration',num2str(Iteration_number),'
=0;']);

eval(['delay_hours','_',num2str(Year(year_number)),'_Entry',num2str(Ent
ry_sep(Entry_number)),'nm','_',num2str(hour),'_',Airport_ID(airport_num
ber,1:3),'=0;']);

                end % for airport_number = 1:length(Airport_ID)

            end % for hour = 1:24
```

```
            for flight_number =
eval(['1:length(','delay','_',num2str(Year(year_number)),'_Entry',num2s
tr(Entry_sep(Entry_number)),'nm)'])
            % for flight_number = 1:length(delay_****_Entry*nm)

                airport_id = char(airport_id);

                aaa = isletter(airport_id(flight_number,:));
                [b c] = max(aaa); % find the starting point of
airport_id



eval(['airport_id','_',num2str(Year(year_number)),'_Entry',num2str(Entr
y_sep(Entry_number)),'nm','_Iteration',...
                    num2str(Iteration_number),'(flight_number,1:3)','=
airport_id(flight_number,c+1:c+3);']); % % airport_id_****_Entry*nm =
airport_id(i,end-2:end)

                for hours = 1:end_sim % find delays and number of
operations processed in SIMMOD

                    if
eval(['end_time','_',num2str(Year(year_number)),'_Entry',num2str(Entry_
sep(Entry_number)),'nm(flight_number)','<=',...
                        num2str(hours)])
                    % if end_time_****_Entry*nm(flight_number) <= hours

                        for airport_number = 1:length(Airport_ID) %
match airport ID

                            if
strcmp(Airport_ID(airport_number,1:3),eval(['airport_id','_',num2str(Ye
ar(year_number)),'_Entry',...

num2str(Entry_sep(Entry_number)),'nm','_Iteration',num2str(Iteration_nu
mber),'(flight_number,1:3)']))
                            %if
strcmp(Airport_ID,airport_id_****_Entry*nm_***


eval(['flights','_',num2str(Year(year_number)),'_Entry',num2str(Entry_s
ep(Entry_number)),'nm','_',num2str(hours),...

'_',Airport_ID(airport_number,1:3),'=','flights','_',num2str(Year(year_
number)),'_Entry',...

num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours),'_',Airport_ID
(airport_number,1:3),'+1;']);
                                % flights_****_Entry*nm_*_*** =
flights_****_Entry*nm_*_*** + 1;
```

```
                                if
eval(['flights','_',num2str(Year(year_number)),'_Entry',num2str(Entry_s
ep(Entry_number)),'nm','_',num2str(hours),...

'_',Airport_ID(airport_number,1:3),' == 1']);
                                % if flights_****_Entry*nm_*_*** == 1


eval(['delay_hours','_',num2str(Year(year_number)),'_Entry',num2str(Ent
ry_sep(Entry_number)),'nm','_',...

num2str(hours),'_',Airport_ID(airport_number,1:3),'
=','delay','_',num2str(Year(year_number)),'_Entry',...

num2str(Entry_sep(Entry_number)),'nm(flight_number);'])
                                    % delay_hours_****_Entry*nm_*_*** =
delay_****_Entry*nm(flight_number);

                                else


eval(['delay_hours','_',num2str(Year(year_number)),'_Entry',num2str(Ent
ry_sep(Entry_number)),'nm','_',...

num2str(hours),'_',Airport_ID(airport_number,1:3),'=','delay_hours','_'
,num2str(Year(year_number)),'_Entry',...

num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours),'_',Airport_ID
(airport_number,1:3),'+','delay','_',...

num2str(Year(year_number)),'_Entry',num2str(Entry_sep(Entry_number)),'n
m(flight_number);']);

                                end % if flights_****_Entry*nm_* == 1

                            end % for airport_number =
1:length(Airport_ID)

                        end % for airport_number = 1:length(Airport_ID)

                    break;

                    end % if end_time_****_Entry*nm(flight_number) <= 1

                end % for hours = 1:end_sim

            end % for flight_number = 1:length(delay_****_Entry*nm)

            for hours = 1:end_sim

                for airport_number = 1:length(Airport_ID)
```

```
eval(['delay_hours','_',num2str(Year(year_number)),'_Entry',num2str(Ent
ry_sep(Entry_number)),'nm_',num2str(hours),'_',...

Airport_ID(airport_number,1:3),'_Iteration',num2str(Iteration_number),'
= delay_hours','_',num2str(Year(year_number)),...

'_Entry',num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours),'_',A
irport_ID(airport_number,1:3),';']);


eval(['flights','_',num2str(Year(year_number)),'_Entry',num2str(Entry_s
ep(Entry_number)),'nm_',num2str(hours),'_',...

Airport_ID(airport_number,1:3),'_Iteration',num2str(Iteration_number),'
= flights','_',num2str(Year(year_number)),...

'_Entry',num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours),'_',A
irport_ID(airport_number,1:3),';']);


                end % for airport_number = 1:length(Airport_ID)

            end % for hours = 1:end_sim

        end % for Iteration_number = 1:Iteration

     end % for Entry_number = 1:length(Entry_sep)

end % for year_number = 1:length(Year)


% to get demand (# of operations enter SIMMOD) in each hour
disp('process data step 2');


for year_number = 1:length(Year)

    disp(['Year ',num2str(Year(year_number))]);

    for Entry_number = 1:length(Entry_sep)

        for Iteration_number = 1:Iteration

            disp(['Entry ', num2str(Entry_sep(Entry_number)),'nm','
Iteration ',num2str(Iteration_number)]);

            for flight_number =
eval(['1:length(','start_time','_',num2str(Year(year_number)),'_Entry',
num2str(Entry_sep(Entry_number)),'nm',...
                        '_Iteration',num2str(Iteration_number),')'])
                % for flight_number = 2:length(delay_****_Entry*nm)
```

```
                for hours = 1:end_sim % find delays and number of
operations processed in SIMMOD

                    % calculate flight demand in each hour

                    if
eval(['start_time','_',num2str(Year(year_number)),'_Entry',num2str(Entr
y_sep(Entry_number)),'nm','_Iteration',...

num2str(Iteration_number),'(flight_number)','<=',num2str(hours)])


                        for airport_number = 1:length(Airport_ID)

                            if
strcmp(Airport_ID(airport_number,1:3),eval(['airport_id','_',num2str(Ye
ar(year_number)),'_Entry',...

num2str(Entry_sep(Entry_number)),'nm','_Iteration',num2str(Iteration_nu
mber),'(flight_number,1:3)']))
                                %if
                                %strcmp(Airport_ID,airport_id_****_Entr
y*nm_***


eval(['demand','_',num2str(Year(year_number)),'_Entry',num2str(Entry_se
p(Entry_number)),'nm','_',num2str(hours),...

'_',Airport_ID(airport_number,1:3),'_Iteration',num2str(Iteration_numbe
r),'=','demand','_',num2str(Year(year_number)),'_Entry',...

num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours),'_',Airport_ID
(airport_number,1:3),...

'_Iteration',num2str(Iteration_number),'+1;']);
                                % flights_****_Entry*nm_* =
flights_****_Entry*nm_* + 1;

                                break;

                            end % if
strcmp(Airport_ID,airport_id_****_Entry*nm_***

                        end % for airport_number = 1:length(Airport_ID)

                        break;

                    end % for hours = 1:end_sim

                end % for flight_number = 2:length(delay_****_Entry*nm)

            end % for for Iteration_number = 1:Iteration

        end % for hours = 1:end_sim
```

```
    end % for Iteration_number = 1:Iteration

end % for year_number = 1:length(Year)

disp('Sum up all Iterations');

for year_number = 1:length(Year)

    disp(['Year ',num2str(Year(year_number))]);

    for Entry_number = 1:length(Entry_sep)

        disp(['Entry ', num2str(Entry_sep(Entry_number)),'nm']);


        for Iteration_number = 1:Iteration

            for hours = 1:end_sim

                for airport_number = 1:length(Airport_ID) % match
airport ID

                    if Iteration_number == 1


eval(['delay_hours','_',num2str(Year(year_number)),'_Entry',num2str(Ent
ry_sep(Entry_number)),'nm','_',num2str(hours),'_',...

Airport_ID(airport_number,1:3),'=','delay_hours','_',num2str(Year(year_
number)),'_Entry',num2str(Entry_sep(Entry_number)),...

'nm','_',num2str(hours),'_',Airport_ID(airport_number,1:3),'_Iteration'
,num2str(Iteration_number),';']);

eval(['flights','_',num2str(Year(year_number)),'_Entry',num2str(Entry_s
ep(Entry_number)),'nm','_',num2str(hours),'_',...

Airport_ID(airport_number,1:3),'=','flights','_',num2str(Year(year_numb
er)),'_Entry',num2str(Entry_sep(Entry_number)),...

'nm','_',num2str(hours),'_',Airport_ID(airport_number,1:3),'_Iteration'
,num2str(Iteration_number),';']);

eval(['demand','_',num2str(Year(year_number)),'_Entry',num2str(Entry_se
p(Entry_number)),'nm','_',num2str(hours),'_',...

Airport_ID(airport_number,1:3),'=','demand','_',num2str(Year(year_numbe
r)),'_Entry',num2str(Entry_sep(Entry_number)),...

'nm','_',num2str(hours),'_',Airport_ID(airport_number,1:3),'_Iteration'
,num2str(Iteration_number),';']);

                    else
```

```
eval(['delay_hours','_',num2str(Year(year_number)),'_Entry',num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours),...

'_',Airport_ID(airport_number,1:3),'=','delay_hours','_',num2str(Year(year_number)),'_Entry',num2str(Entry_sep(Entry_number)),...

'nm','_',num2str(hours),'_',Airport_ID(airport_number,1:3),'+','delay_hours','_',num2str(Year(year_number)),'_Entry',...

num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours),'_',Airport_ID(airport_number,1:3),'_Iteration',num2str(Iteration_number),';']);


eval(['flights','_',num2str(Year(year_number)),'_Entry',num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours),...

'_',Airport_ID(airport_number,1:3),'=','flights','_',num2str(Year(year_number)),'_Entry',num2str(Entry_sep(Entry_number)),...

'nm','_',num2str(hours),'_',Airport_ID(airport_number,1:3),'+','flights','_',num2str(Year(year_number)),'_Entry',...

num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours),'_',Airport_ID(airport_number,1:3),'_Iteration',num2str(Iteration_number),';']);


eval(['demand','_',num2str(Year(year_number)),'_Entry',num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours),...

'_',Airport_ID(airport_number,1:3),'=','demand','_',num2str(Year(year_number)),'_Entry',num2str(Entry_sep(Entry_number)),...

'nm','_',num2str(hours),'_',Airport_ID(airport_number,1:3),'+','demand','_',num2str(Year(year_number)),'_Entry',...

num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours),'_',Airport_ID(airport_number,1:3),'_Iteration',num2str(Iteration_number),';']);

                    end % if Iteration_number == 1

                end % if Iteration_number == 1

            end % for airport_number = 1:length(Airport_ID)

        end % for Iteration_number = 1:Iteration

    end % for Entry_number = 1:length(Entry_sep)

end % for year_number = 1:length(Year)


disp('calculate average delay');

for year_number = 1:length(Year)
```

```
    disp(['Year ',num2str(Year(year_number))]);

    for Entry_number = 1:length(Entry_sep)

        disp(['Entry ', num2str(Entry_sep(Entry_number)),'nm']);


        for Iteration_number = 1:Iteration

            for hours = 1:end_sim

                for airport_number = 1:length(Airport_ID) % match
airport ID

                    if
eval(['delay_hours','_',num2str(Year(year_number)),'_Entry',num2str(Ent
ry_sep(Entry_number)),'nm','_',num2str(hours),...
                        '_',Airport_ID(airport_number,1:3),'==0'])
                    % if delay_hours_****_Entry*nm_*_*** == 0


eval(['average_total_delay','_',num2str(Year(year_number)),'_Entry',num
2str(Entry_sep(Entry_number)),'nm','_',num2str(hours),...
                            '_',Airport_ID(airport_number,1:3),'=0;']);

                    % average_delay_hours_****_Entry*nm_*_*** = 0

                    else

eval(['average_total_delay','_',num2str(Year(year_number)),'_Entry',num
2str(Entry_sep(Entry_number)),'nm','_',num2str(hours),...

'_',Airport_ID(airport_number,1:3),'=','delay_hours','_',num2str(Year(y
ear_number)),'_Entry',num2str(Entry_sep(Entry_number)),...

'nm','_',num2str(hours),'_',Airport_ID(airport_number,1:3),'/','flights
','_',num2str(Year(year_number)),...

'_Entry',num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours),'_',A
irport_ID(airport_number,1:3),';']);
                        % average_delay_****_Entry*nm_*_*** =
delay_hours_****_Entry*nm_*_*** /flights_****_Entry*nm_*_***;

                    end % % if delay_hours_****_Entry*nm_** == 0

                end % for airport_number = 1:length(Airport_ID)

            end % for hour = 1:end_sim

        end % for Iteration_number = 1:Iteration

    end % for Entry_number = 1:length(Entry_sep)
```

```
end % for year_number = 1:length(Year)


save average_total_delay_airport average_total_delay*;


disp('plot data')

for year_number = 1:length(Year)

    disp(['Year ',num2str(Year(year_number))]);


    for airport_number = 1:length(Airport_ID)

        figure;

        xlabel('Time')
        ylabel('Average Total Delay (Minutes per Operation)');
        title([Airport_ID(airport_number,1:3),' Delay Analysis (Total
Delay, ','Year ',num2str(Year(year_number)),')']);

        for Entry_number = 1:length(Entry_sep)

            color_code = ['r' 'b' 'k'];

            for hours = 2:end_sim

                grid;

                line([hours hours-
1],[eval(['average_total_delay','_',num2str(Year(year_number)),'_Entry'
,num2str(Entry_sep(Entry_number)),'nm','_',...

num2str(hours),'_',Airport_ID(airport_number,1:3),'*60']),eval(['averag
e_total_delay','_',num2str(Year(year_number)),'_Entry',...

num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours-
1),'_',Airport_ID(airport_number,1:3),'*60'])],'color',...
                    color_code(Entry_number),'LineWidth',2);

            end % for hour = 1:end_sim

        end % for Entry_number = 1:length(Entry_sep)

        figure;
        xlabel('Time (Hour)')
        ylabel('Number of Operations');
        title(['Number of Operations Entering and Leaving Simulation
','Year - (',num2str(Year(year_number)),')']);
        %     legend('# of Operations Processed in SIMMOD','Red:
Intrail Separation = 7nm', 'Blue: Intrail Separation = 8nm', ...
```

```
        %          'Black: Intrail Separation = 9nm','# of Operations
Enters
        %          SIMMOD','Cyan: Operations Enters SIMMOD');

        for Entry_number = 1:length(Entry_sep)

            disp(['Entry ', num2str(Entry_sep(Entry_number)),'nm']);

            color_code = ['r' 'b' 'k'];

            for hours = 2:end_sim

                grid;

                line([hours hours-
1],[eval(['flights','_',num2str(Year(year_number)),'_Entry',num2str(Ent
ry_sep(Entry_number)),'nm','_',...

num2str(hours),'_',Airport_ID(airport_number,1:3),'/Iteration']),eval([
'flights','_',num2str(Year(year_number)),...

'_Entry',num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours-
1),'_',Airport_ID(airport_number,1:3),'/Iteration'])],...
                    'color',color_code(Entry_number),'LineWidth',2);
                if Entry_number == 3

                    line([hours hours-
1],[eval(['demand','_',num2str(Year(year_number)),'_Entry',num2str(Entr
y_sep(Entry_number)),'nm','_',...

num2str(hours),'_',Airport_ID(airport_number,1:3),'/Iteration']),eval([
'demand','_',num2str(Year(year_number)),...

'_Entry',num2str(Entry_sep(Entry_number)),'nm','_',num2str(hours-
1),'_',Airport_ID(airport_number,1:3),'/Iteration'])],...
                        'color','c','LineWidth',2);

                end % if Entry_number == 3

            end % for hour = 1:24

        end % for Entry_number = 1:length(Entry_sep)

    end % for airport_number = 1:length(Airport_ID)

end % for year_number = 1:length(Year)
```

ATO Traffic Parser

```
% To create a SIMMOD Traffic File from the ATO Forecast struct array

% Input: forecast_Q2_PEAK_2006_2006_smoothed.mat

%         ATO_Forecast(FlightCounter).id_num
%         ATO_Forecast(FlightCounter).act_date
%         ATO_Forecast(FlightCounter).acid
%         ATO_Forecast(FlightCounter).flight_index
%         ATO_Forecast(FlightCounter).flight_plan_type
%         ATO_Forecast(FlightCounter).departure_time
%         ATO_Forecast(FlightCounter).departure_time_flag
%         ATO_Forecast(FlightCounter).arrival_time
%         ATO_Forecast(FlightCounter).arrival_time_flag
%         ATO_Forecast(FlightCounter).filed_alititude
%         ATO_Forecast(FlightCounter).filed_airspeed
%         ATO_Forecast(FlightCounter).etms_departure_airport
%         ATO_Forecast(FlightCounter).etms_arrival_airport
%         ATO_Forecast(FlightCounter).dept_lat
%         ATO_Forecast(FlightCounter).dept_lon
%         ATO_Forecast(FlightCounter).dept_elev
%         ATO_Forecast(FlightCounter).dept_cntry_code
%         ATO_Forecast(FlightCounter).arr_lat
%         ATO_Forecast(FlightCounter).arr_lon
%         ATO_Forecast(FlightCounter).arr_elev
%         ATO_Forecast(FlightCounter).arr_cntry_code
%         ATO_Forecast(FlightCounter).aircraft_type
%         ATO_Forecast(FlightCounter).physical_class
%         ATO_Forecast(FlightCounter).user_class
%         ATO_Forecast(FlightCounter).flew_flag
%         ATO_Forecast(FlightCounter).airspace_code
%         ATO_Forecast(FlightCounter).dept_icao_code
%         ATO_Forecast(FlightCounter).arr_icao_code
%         ATO_Forecast(FlightCounter).new_user_class
%         ATO_Forecast(FlightCounter).bada_type
%         ATO_Forecast(FlightCounter).bada_source_type

% Coded by Yue Xu, 10/11/07, version 1.0
% Version 2.0, deleted random number for departures, added departure
% service time


clc;

global ATO_Forecast

Airport_ID_List = ['LGA'; 'EWR'; 'JFK'];

FolderName = 'D:\Research\NewYork Metroplex\ATO Parser\';

FileName = 'forecast_Q2_PEAK_2006_2025_smoothed_filtered';
```

```
disp('Loading...');
load([FolderName, FileName, '.mat']);

disp('Loading Done');

[Number_of_Departures_JFK] = Departures_count('JFK');

[Number_of_Arrivals_JFK_31L] =
Arrivals_Balance(Number_of_Departures_JFK);

[Total_Number_of_Arrivals_JFK] = Arrivals_count('JFK');

Percentage_of_Arrivals_to_31L = Number_of_Arrivals_JFK_31L ./
Total_Number_of_Arrivals_JFK;

Percentage_of_Arrivals_to_31L = min(Percentage_of_Arrivals_to_31L,1);

RWY31L_Count = zeros(24,1);

comma = ',';

Flight_Index_Number = 0;

DEP_Service_SML = 30; % departure service time for Small aircraft in
minutes
DEP_Service_LRG = 40; % departure service time for Large aircraft in
minutes
DEP_Service_HVY = 50; % departure service time for Heavy aircraft in
minutes

% ------------------------
% Setup box around airports
% ------------------------

Polygon_Latitudes  = [42.34207 42.39581 41.88247 41.62858 41.16985
39.94778 39.67062 39.6875 39.89369 40.06081 40.70777 40.94836 41.98255
42.3156];

Polygon_Longitudes = [-73.99614 -73.60829 -72.92616 -72.36090 -71.91034
-73.22402 -73.56126 -74.18774 -74.97654 -75.14857 -75.50947 -75.43555 -
75.1346 -74.84523];

% ------------------------
% Writing file
% ------------------------

Number_Of_Airports              = length(Airport_ID_List);

Number_Of_Flights               = length(ATO_Forecast);

Number_Of_Departures            = 0;

Number_Of_Arrivals              = 0;
```

```
Number_Of_Military_Flights      = 0;

Number_Of_VFR_Flights           = 0;

Number_Of_Skipped_Flights       = 0;

Unique_Traffic_Callsign_Counter = 0;

Unique_Traffic_Callsigns        = [];

Unique_Traffic_Callsigns_Count  = [];

[RAMS_Aircraft, SIMMOD_Aircraft, aircraft_class] =
textread([FolderName,'RAMS_SIMMOD_AIRCRAFT_CONVERT.txt'],'%s %s %s');

load ('D:\Research\NewYork Metroplex\Route\Traverse_Time_Route.mat');


fid = fopen(['SIMMOD_NYC_', FileName, '.txt'], 'w');

for j = 1:Number_Of_Airports

    Airport_ID = Airport_ID_List(j,:);

    if strcmp(Airport_ID, 'LGA') == 1 % LGA

        Approach_Fixes_Names            = {'APP_N', 'APP_N_W', 'APP_N_E',
'APP_S', 'APP_S_W'}';

        Approach_Entry_Fixes_Names     = {'LGA_N_ENTRY_22',
'LGA_NW_ENTRY_22', 'LGA_NE_ENTRY_22', 'LGA_S_ENTRY_22',
'LGA_SW_ENTRY_22'}';

        Approach_Entry_Fixes_Longitude = [-73.60829 -74.84523 -72.36090
-74.97654 -75.50947];

        Approach_Entry_Fixes_Latitude  = [42.39580 42.3156 41.62858
39.89369 40.70777];

        Approach_Path_Names            = {'LGA_APP_N_22',
'LGA_APP_NW_22', 'LGA_APP_NE_22', 'LGA_APP_S_22', 'LGA_APP_SW_22'}';

%         Approach_Fixes_Longitude   = [-72.8746 -78.6650 -69.0217 -
76.5608 -78.4971]';

%         Approach_Fixes_Latitude    = [44.2868 43.1223 42.1270 38.3557
41.2866]';
```

```
        Departure_Path_Names        = {'LGA_DEP_NW_31',
'LGA_DEP_SW_31'};

        Departure_Fixes_Longitude   = [-75.4002 -75.4397]';

        Departure_Fixes_Latitude    = [41.6671 40.8435]';

    elseif strcmp(Airport_ID, 'EWR') == 1 % EWR

        Approach_Fixes_Names            = {'APP_N', 'APP_N_W', 'APP_N_E',
'APP_S', 'APP_S_E'}';

        Approach_Entry_Fixes_Names     = {'EWR_N_ENTRY_22L',
'EWR_NW_ENTRY_22L', 'EWR_NE_ENTRY_22L', 'EWR_SW_ENTRY_22L',
'EWR_SE_ENTRY_22L'}';

        Approach_Entry_Fixes_Longitude = [-73.99614 -75.1346 -72.92616
-75.14857 -73.22402 ];

        Approach_Entry_Fixes_Latitude  = [42.99614 41.98255 41.88247
40.06081 39.94778];

        Approach_Path_Names            = {'EWR_APP_N_22L',
'EWR_APP_NW_22L', 'EWR_APP_NE_22L', 'EWR_APP_SW_22L',
'EWR_APP_SE_22L'}';

%          Approach_Fixes_Longitude   = [-72.8746 -78.6650 -69.0217 -
76.5608 -71.2252]';

%          Approach_Fixes_Latitude    = [44.2868 43.1223 42.1270 38.3557
38.1939]';



        Departure_Path_Names        = {'EWR_DEP_NE_22R',
'EWR_DEP_S_22R', 'EWR_DEP_W_22R'};

        Departure_Fixes_Longitude   = [-72.6550 -74.7247 -75.4397]';

        Departure_Fixes_Latitude    = [41.6468 39.7016 40.4238]';

    elseif strcmp(Airport_ID, 'JFK') == 1 % JFK

        Approach_Fixes_Names            = {'APP_S_W', 'APP_N_E', 'APP_S',
'APP_S_E'}';

        Approach_Entry_Fixes_Names     = {'JFK_NW_ENTRY_31R_31L',
'JFK_NE_ENTRY_31R', 'JFK_SW_ENTRY_31R_31L', 'JFK_SE_ENTRY_31R_31L'}';

        Approach_Entry_Fixes_Longitude = [-75.33076 -71.91034 -74.18774
-73.56126];
```

```
        Approach_Entry_Fixes_Latitude  = [41.15460 41.16985 39.6875
39.67062];

        Approach_Path_Names            = {'JFK_APP_NW_31R',
'JFK_APP_NE_31R', 'JFK_APP_SW_31R', 'JFK_APP_SE_31R'}';

%         Approach_Fixes_Longitude   = [-78.4971 -69.0217 -76.5608 -
71.2252]';

%         Approach_Fixes_Latitude    = [41.2866 42.1270 38.3557
38.1939]';



        Departure_Path_Names         = {'JFK_DEP_W_31L', 'JFK_DEP_S_31L',
'JFK_DEP_E_31L', 'JFK_DEP_NE_31L', 'JFK_DEP_N_31L', 'JFK_DEP_NW_31L'};

        Departure_Fixes_Longitude   = [-75.3591 -73.4974 -72.0373 -
72.1718 -73.3378 -75.4002]';

        Departure_Fixes_Latitude    = [40.2065 39.8046 40.6533 41.5027
42.2895 41.6671]';

    End

    Number_Of_Approach_Fixes  = length(Approach_Fixes_Names);

    Number_Of_Departure_Paths = length(Departure_Path_Names);

    for i = 1:Number_Of_Flights

        Error = 'False';

        if mod(i, 1000) == 0

            disp(['Number of flights processed for ', Airport_ID, ' : ',
num2str(i)]);

        end

        Departure_Airport = ATO_Forecast(i).etms_departure_airport;

        Arrival_Airport   = ATO_Forecast(i).etms_arrival_airport;

        if strcmp(Departure_Airport, Airport_ID) == 1 ||
strcmp(Arrival_Airport, Airport_ID) == 1 % Only save flight to/from
Airport_ID

            % ------------------------
            % Get Operation Type
            % ------------------------

            if strcmp(Departure_Airport, Airport_ID) == 1 % Departure
```

```
                Number_Of_Departures = Number_Of_Departures + 1;

                OperationType = 'Departure';

            else %Arrival

                Number_Of_Arrivals = Number_Of_Arrivals + 1;

                OperationType = 'Arrival';

            end % if strcmp(etms_departure_airport, Airport_ID) == 1


            % ------------------------
            % Format Output
            % ------------------------

            if strcmp(OperationType, 'Arrival') == 1 % Adjust Entry
time of arrivals
%

                    EntryTime = ATO_Forecast(i).arrival_time(end -
7:end); % HH:MM:SS

                    EntryTime_InDays = datenum(EntryTime);

                EntryTime = datestr(EntryTime_InDays, 'HHMMSS');

                EntryTime_Plus24Hrs = EntryTime;

                EntryTime_Plus24Hrs(1:2) =
num2str(str2num(EntryTime(1:2)) + 24);

            elseif strcmp(OperationType, 'Departure') == 1

                EntryTime = ATO_Forecast(i).departure_time(end -
7:end); % HH:MM:SS

                EntryTime_InDays = datenum(EntryTime); %+
randn/60/24; % Time in Fraction of Days

                EntryTime = datestr(EntryTime_InDays, 'HHMMSS');

                EntryTime_Plus24Hrs = EntryTime;

                EntryTime_Plus24Hrs(1:2) =
num2str(str2num(EntryTime(1:2)) + 24);

            end
```

```
            Traffic_Callsign = ATO_Forecast(i).acid;

            % Skip if Military Flights

            if strcmp(ATO_Forecast(i).user_class, 'M') == 1

                disp(['WARNING: Military flight skipped for index: ',
num2str(i)]);

            Number_Of_Military_Flights = Number_Of_Military_Flights + 1;

                continue;

            end

            Flight_Index_Number = Flight_Index_Number + 1;

            % Save unique traffic callsigns. If duplicate found, add
letter to end of it. THis is to avoid RAMS getting crazy.

            if isempty(strmatch(Traffic_Callsign,
Unique_Traffic_Callsigns, 'exact')) == 1 % Unique traffic callsign
found

                Unique_Traffic_Callsign_Counter =
Unique_Traffic_Callsign_Counter + 1;


Unique_Traffic_Callsigns{Unique_Traffic_Callsign_Counter,1} =
Traffic_Callsign;


Unique_Traffic_Callsigns_Count(Unique_Traffic_Callsign_Counter,1) = 1;

                Traffic_Callsign = [Traffic_Callsign, 'A1'];

                Traffic_Callsign_Plus24Hrs = Traffic_Callsign;

                Traffic_Callsign_Plus24Hrs(end) = '2';

            else % Duplicate traffic callsign found

                Index_Of_Unique_Traffic_Callsign =
strmatch(Traffic_Callsign, Unique_Traffic_Callsigns, 'exact');


Unique_Traffic_Callsigns_Count(Index_Of_Unique_Traffic_Callsign,1) =
Unique_Traffic_Callsigns_Count(Index_Of_Unique_Traffic_Callsign,1) + 1;

                % Add letter A B C... to end of traffic call sign
```

```
                Traffic_Callsign = [Traffic_Callsign, char(64 +
Unique_Traffic_Callsigns_Count(Index_Of_Unique_Traffic_Callsign,1)),
'1'];

                Traffic_Callsign_Plus24Hrs = Traffic_Callsign;

                Traffic_Callsign_Plus24Hrs(end) = '2';

            end

            % Get Aircraft Model %

            if isempty(ATO_Forecast(i).bada_type) == 0

                Aircraft_Model = ATO_Forecast(i).bada_type;

            else

                if strcmp(ATO_Forecast(i).flight_plan_type, 'VFR') == 1

                    Aircraft_Model = 'BE20';

                else

                    Aircraft_Model = '???';

                    disp(['ERROR: Invalid aircraft model type for
flight index: ', num2str(i)]);

                    Number_Of_Skipped_Flights =
Number_Of_Skipped_Flights + 1;

                    continue;

          end % if strcmp(ATO_Forecast(i).flight_plan_type, 'VFR') == 1

       end % if isempty(ATO_Forecast(i).bada_type) == 0


            % Get Airline Name %

            if strcmp(ATO_Forecast(i).acid(1:3),'V_F') % VFR flight

                Airline_ID = 'V_F';

            else

                Airline_ID = ATO_Forecast(i).acid(1:3); % IFR flight,
Airline ID is the first three letter

            end % if strcmp(ATO_Forecast(i).acid(1:3),'V_F') % VFR
flight
```

```
            % Get time in decimal formats

            Time_Hr     = str2num(EntryTime(1:2));
            Time_Minute = str2num(EntryTime(3:4));
            Time_Second = str2num(EntryTime(5:6));
            Time_in_Decimal = (Time_Hr + (Time_Minute * 60 +
Time_Second) / 3600) - 5; % UTC TO lOCAL


            if ATO_Forecast(i).user_class == 'C'

                Flight_Category = 'C';

            elseif ATO_Forecast(i).user_class == 'F'

                Flight_Category = 'F';

            elseif ATO_Forecast(i).user_class == 'G'

                Flight_Category = 'G';

            elseif ATO_Forecast(i).user_class == 'M'

                Flight_Category = 'M';

            elseif ATO_Forecast(i).user_class == 'T'

                Flight_Category = 'O';

            elseif ATO_Forecast(i).user_class == 'O'

                Flight_Category = 'O';

            elseif strcmp(ATO_Forecast(i).flight_plan_type, 'VFR') == 1

                Flight_Category = 'G';

            else

                Flight_Category = '???';

                disp(['ERROR: Unknown user class "',
ATO_Forecast(i).user_class, '" for flight index: ', num2str(i)]);

              Number_Of_Skipped_Flights = Number_Of_Skipped_Flights + 1;

                continue;

            end
```

```
            NAV_Equipment = 'DefaultACNavEquipment';


            % Find closest approach entry point from departure airport

            if strcmp(OperationType, 'Arrival') == 1

                % Skip if VFR flight

                if strcmp(ATO_Forecast(i).flight_plan_type, 'VFR') == 0

                    Altitude_Entry_Level =
ATO_Forecast(i).dept_elev/100; % Convert to Flight Level

                    Altitude_Cruise_Level =
ATO_Forecast(i).filed_alititude;

                    Altitude_Exit_Level =
ATO_Forecast(i).arr_elev/100; % Convert to Flight Level



                    % Remove points inside box

                    if isempty(ATO_Forecast(i).waypoints) == 1

                        disp(['ERROR: Flight has no waypoints! Flight
index: ', num2str(i)]);

                        Number_Of_Skipped_Flights =
Number_Of_Skipped_Flights + 1;

                        continue

                    end

                    Waypoints_Latitude =
ATO_Forecast(i).waypoints.latitude;

                    Waypoints_Longitude =
ATO_Forecast(i).waypoints.longitude;

                    [InOut_Box_Points] = inpolygon(Waypoints_Longitude,
Waypoints_Latitude, Polygon_Longitudes, Polygon_Latitudes);

                    Index_Of_Outside_Points = find(InOut_Box_Points ==
0);

                    Waypoints_Latitude =
Waypoints_Latitude(Index_Of_Outside_Points);
```

```
                    Waypoints_Longitude =
Waypoints_Longitude(Index_Of_Outside_Points);



                    % Find closest approach entry point based on last
waypoint

                    if length(Waypoints_Longitude) == 0

                        hold on;

                        plot(Polygon_Longitudes, Polygon_Latitudes);

                        plot(ATO_Forecast(i).dept_lon,
ATO_Forecast(i).dept_lat, 'x')

                        plot(ATO_Forecast(i).arr_lon,
ATO_Forecast(i).arr_lat, '+')

                        disp(['ERROR: Flight too short! Flight index: ',
num2str(i)]);

                        Number_Of_Skipped_Flights =
Number_Of_Skipped_Flights + 1;

                        continue;

                    end

                    Last_Waypoint_Longitude = Waypoints_Longitude(end);

                    Last_Waypoint_Latitude = Waypoints_Latitude(end);

                    Distance_LastWaypoint_To_Approach =
deg2nm(distance(Last_Waypoint_Latitude, Last_Waypoint_Longitude,
Approach_Entry_Fixes_Latitude, Approach_Entry_Fixes_Longitude));

                    [Closest_Approach_Fix_Distance,
Closest_Approach_Fix_Index] = min(Distance_LastWaypoint_To_Approach);

                    %                 Closest_Approach_Fix_Name =
char(Approach_Fixes_Names{Closest_Approach_Fix_Index});

                    Closest_Approach_Entry_Fix_Name =
char(Approach_Entry_Fixes_Names{Closest_Approach_Fix_Index});

                    Closest_Approach_Path_Name      =
char(Approach_Path_Names{Closest_Approach_Fix_Index});

                else % VFR Flight

                    Number_Of_Waypoints = 0;
```

```
                    Waypoints_Latitude  = [];
                    Waypoints_Longitude = [];
                    Number_Of_VFR_Flights = Number_Of_VFR_Flights + 1;

                    Altitude_Entry_Level = 180; % Convert to Flight
Level
                    Altitude_Cruise_Level = 180;
                    Altitude_Exit_Level = 0; % Convert to Flight Level

                    % Assign VFR to random aproach
                    Approach_Fix_Index =
find(randperm(Number_Of_Approach_Fixes) == 1);
                    Closest_Approach_Entry_Fix_Name =
char(Approach_Entry_Fixes_Names(Approach_Fix_Index));
                    Closest_Approach_Path_Name      =
char(Approach_Path_Names{Approach_Fix_Index});
                end
                Departure_Airport = 'ADEP'; % Default Departure Airport

            elseif strcmp(OperationType, 'Departure') == 1 % Find
closest departure exit point to destination airport


                % Skip if VFR flight
                if strcmp(ATO_Forecast(i).flight_plan_type, 'VFR') == 0
                    Altitude_Entry_Level =
ATO_Forecast(i).arr_elev/100; % Convert to Flight Level
                    Altitude_Cruise_Level =
ATO_Forecast(i).filed_alititude;
                    Altitude_Exit_Level =
ATO_Forecast(i).dept_elev/100; % Convert to Flight Level
```

```
                    % Find closest departure exit point to destination
airport

                    Arrival_Airport_Longitude =
double(ATO_Forecast(i).arr_lon);

                    Arrival_Airport_Latitude =
double(ATO_Forecast(i).arr_lat);

                    Distance_Departure_To_Destination =
deg2nm(distance(Arrival_Airport_Latitude, Arrival_Airport_Longitude,
Departure_Fixes_Latitude, Departure_Fixes_Longitude));

                    [Closest_Departure_Fix_Distance,
Closest_Departure_Fix_Index] = min(Distance_Departure_To_Destination);

                    Closest_Departure_Path_Name =
char(Departure_Path_Names{Closest_Departure_Fix_Index});

                else % VFR Flight


                    Number_Of_Waypoints = 0;

                    Waypoints_Latitude  = [];

                    Waypoints_Longitude = [];

                    Number_Of_VFR_Flights = Number_Of_VFR_Flights + 1;

                    Altitude_Entry_Level = 0; % Convert to Flight Level

                    Altitude_Cruise_Level = 180;

                    Altitude_Exit_Level = 180; % Convert to Flight
Level

                    % Assign VFR to random departure

                    Departure_Paths_Index =
find(randperm(Number_Of_Departure_Paths) == 1);

                    Closest_Departure_Path_Name =
char(Departure_Path_Names{Departure_Paths_Index});


                end

                Arrival_Airport = 'ADES'; % Default Departure Airport

            end
```

```
            if strcmp(OperationType, 'Arrival') == 1

                Departure_Runway = 'RWY';

                if strcmp(Airport_ID, 'LGA') == 1

                    Arrival_Runway = 'LGA22';

                elseif strcmp(Airport_ID, 'EWR') == 1

                    Arrival_Runway = 'EWR22L';

                elseif strcmp(Airport_ID, 'JFK') == 1

                    if strcmp(Closest_Approach_Path_Name,
'JFK_APP_NE_31R') == 1

                        Arrival_Runway = 'JFK31R';

                    elseif strcmp(Closest_Approach_Path_Name,
'JFK_APP_NW_31R') == 1 || strcmp(Closest_Approach_Path_Name,
'JFK_APP_SW_31R') == 1 || strcmp(Closest_Approach_Path_Name,
'JFK_APP_SE_31R') == 1

                        Random_Number = rand;

                        Arrival_Time =
(datenum(ATO_Forecast(i).arrival_time(end - 7:end)) -
datenum('00:00:00')) * 24; % Hrs

                        Arrival_Time_Ceil = ceil(Arrival_Time);

                        if Arrival_Time_Ceil > 24 % Assign past
midnight to first bin

                            Arrival_Time_Ceil = 1;

                        end

                        Percentage_of_Arrivals_to_31L_ThisHour =
Percentage_of_Arrivals_to_31L(Arrival_Time_Ceil);

                        if Random_Number >
Percentage_of_Arrivals_to_31L_ThisHour

                            Arrival_Runway = 'JFK31R';

                        else
```

```
                            RWY31L_Count(Arrival_Time_Ceil) =
RWY31L_Count(Arrival_Time_Ceil) + 1;

                            Arrival_Runway = 'JFK31L';

                            Closest_Approach_Path_Name(end) = 'L';

                        end

                    end

                end

            elseif strcmp(OperationType, 'Departure') == 1

                if strcmp(Airport_ID, 'LGA') == 1

                    Departure_Runway = 'LGA31';

                elseif strcmp(Airport_ID, 'EWR') == 1

                    Departure_Runway = 'EWR22R';

                elseif strcmp(Airport_ID, 'JFK') == 1

                    Departure_Runway = 'JFK31L';

                end

                Arrival_Runway   = 'RWY';

            end

            % ----- calculate time of injection for arrivals ---------
 %

            if strcmp(OperationType, 'Arrival') == 1 % To get injection
time to the entry point

                for route_count = 1: 1:length(Route_Name)

                    if strcmp(Closest_Approach_Path_Name,
Route_Name(route_count))

                        for aircraft_type_name =
1:length(RAMS_Aircraft) % find the aircraft type in RAMS and convert to
SIMMOD aircraft

                            if
strcmp(RAMS_Aircraft(aircraft_type_name),Aircraft_Model) == 1
```

```
                                SIMMOD_Model =
SIMMOD_Aircraft(aircraft_type_name);

                                % Time to inject = time of arrival -
route traverse time

                                if
strcmp(aircraft_class(aircraft_type_name),'SML') % Small aircraft

         Time_in_Decimal = Time_in_Decimal - Route_Time(route_count,1);

                                elseif
strcmp(aircraft_class(aircraft_type_name),'LRG') % Small aircraft

         Time_in_Decimal = Time_in_Decimal - Route_Time(route_count,2);

                                elseif
strcmp(aircraft_class(aircraft_type_name),'HVY') % Small aircraft

         Time_in_Decimal = Time_in_Decimal - Route_Time(route_count,3);

                                end

                                break;

                            end % if
strcmp(RAMS_Aircraft(aircraft_type_name),Aircraft_Model) == 1

                        end % for aircraft_type_name =
1:length(RAMS_Aircraft)

                    end % if strcmp(Closest_Approach_Path_Name,
Route_Name(route_count))

                end % for route_count = 1: 1:length(Route_Name)

            end % if strcmp(OperationType, 'Arrival') == 1

            if strcmp(OperationType, 'Departure') == 1 % To get
departure time minus departure service time

                for aircraft_type_name = 1:length(RAMS_Aircraft) % find
the aircraft type in RAMS and convert to SIMMOD aircraft

                    if
strcmp(RAMS_Aircraft(aircraft_type_name),Aircraft_Model) == 1

            SIMMOD_Model == SIMMOD_Aircraft(aircraft_type_name);

            Closest_Departure_Path_Name_SIMMOD =
      [Closest_Departure_Path_Name,'_',char(aircraft_class(aircraft_typ
      e_name))];
```

```
                        if
strcmp(aircraft_class(aircraft_type_name),'SML') % Small aircraft

             Time_in_Decimal = Time_in_Decimal - DEP_Service_SML/60;

                        elseif
strcmp(aircraft_class(aircraft_type_name),'LRG') % Small aircraft

             Time_in_Decimal = Time_in_Decimal - DEP_Service_LRG/60;

                        elseif
strcmp(aircraft_class(aircraft_type_name),'HVY') % Small aircraft

             Time_in_Decimal = Time_in_Decimal - DEP_Service_HVY/60;

                        end % if
strcmp(aircraft_class(aircraft_type_name),'SML')

                        break;

                    end  % if
strcmp(RAMS_Aircraft(aircraft_type_name),Aircraft_Model) == 1


                end % for aircraft_type_name = 1:length(RAMS_Aircraft)

            end % if strcmp(OperationType, 'Departure') == 1



            % ------------------------
            % Write Output
            % ------------------------




            % ---------------
            % Traffic Profile File
            % ---------------

            % First day

            if Time_in_Decimal > 0 && Time_in_Decimal <=  19 % UTC 5AM-
24PM, equivalent to Eastern Time 0AM - 19PM

                if strcmp(OperationType, 'Arrival') == 1

                    fprintf(fid,
'%s%s%f%s%s%s%s%s%s%s%s%s%d%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s\n',
Traffic_Callsign,comma,Time_in_Decimal, comma, comma, comma, comma, 'A',
comma, Airline_ID,comma,Flight_Index_Number,comma, char(SIMMOD_Model),
comma, Closest_Approach_Path_Name, comma, comma, comma, comma,
```

```
'LateNess_Flight', comma, comma, 'ADES', comma, Arrival_Airport, comma,
comma, comma, comma);


                elseif strcmp(OperationType, 'Departure') == 1

                    fprintf(fid,
'%s%s%f%s%s%s%s%s%s%s%s%d%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s\n',
Traffic_Callsign,comma,Time_in_Decimal, comma, comma, comma, comma, 'D',
comma, Airline_ID,comma,Flight_Index_Number,comma, char(SIMMOD_Model),
comma, Closest_Departure_Path_Name_SIMMOD, comma, comma, comma, comma,
'LateNess_Flight', comma, comma, Departure_Airport, comma, 'ADES',
comma, comma, comma, comma);

                end % if strcmp(OperationType, 'Arrival') == 1

            elseif Time_in_Decimal <= 0 % UTC 0-5, equivalent to
Eastern Time 19PM-24PM

                Time_in_Decimal = Time_in_Decimal + 24;    % convert to
19PM-24PM flights

                if strcmp(OperationType, 'Arrival') == 1

                    for aircraft_type_name = 1:length(RAMS_Aircraft) %
find the aircraft type in RAMS and convert to SIMMOD aircraft

                        if
strcmp(RAMS_Aircraft(aircraft_type_name),Aircraft_Model) == 1

          SIMMOD_Model = SIMMOD_Aircraft(aircraft_type_name);

                            break;

                        end


                    end % for aircraft_type_name =
1:length(RAMS_Aircraft)

                    fprintf(fid,
'%s%s%f%s%s%s%s%s%s%s%s%d%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s\n',
Traffic_Callsign,comma,Time_in_Decimal, comma, comma, comma, comma, 'A',
comma, Airline_ID,comma,Flight_Index_Number,comma, char(SIMMOD_Model),
comma, Closest_Approach_Path_Name, comma, comma, comma, comma,
'LateNess_Flight', comma, comma, 'ADES', comma, Arrival_Airport, comma,
comma, comma, comma);

                elseif strcmp(OperationType, 'Departure') == 1

                    fprintf(fid,
'%s%s%f%s%s%s%s%s%s%s%s%d%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s\n',
Traffic_Callsign,comma,Time_in_Decimal, comma, comma, comma, comma, 'D',
```

```
comma, Airline_ID,comma,Flight_Index_Number,comma, char(SIMMOD_Model),
comma, Closest_Departure_Path_Name_SIMMOD, comma, comma, comma, comma,
'LateNess_Flight', comma, comma, Departure_Airport, comma,'ADES' ,
comma, comma, comma, comma);

                end % if strcmp(OperationType, 'Arrival') == 1

            end


        end % if strcmp(etms_departure_airport, Airport_ID) == 1 ||
strcmp(etms_arrival_airport, Airport_ID) == 1

    end % for i = 1:Number_Of_Flights

end % for j = 1:Number_Of_Airports

fclose('all');

disp(' ');

disp(['Number of departures from NYC: ',
num2str(Number_Of_Departures)]);

disp(['Number of arrivals from NYC  : ', num2str(Number_Of_Arrivals)]);

disp(['Number of skipped military flights in NYC  : ',
num2str(Number_Of_Military_Flights)]);

disp(['Number of VFR flights in NYC  : ',
num2str(Number_Of_VFR_Flights)]);

disp(['Number of skipped flights in NYC  : ',
num2str(Number_Of_Skipped_Flights)]);
```

Emission Analysis

```
% This program is to estimate SATS emission impacts on the national
level
% Input Files:SATS_Airport_Operations.txt
%               Column 1: Airport ID
%               Column 2: State
%               Column 3: Longitude
%               Column 4: Latitude
%               Column 5: Single Engine Aircraft Based at the airport
%               Column 6: Multi Engine Aircraft Based at the airport
%               Column 7: Jet Aircraft Based at the airport
%               Column 8: Helicopter Based at the airport
%               Column 9: Air Taxi Ops at the airport
%               Column 10: Local GA Ops at the airport
%               Column 11: Itinerant GA Ops at the airport
%               Column 12: SATS Ops at the airport
% Output Files: Emission Outputs.mat
%               File 1: Baseline emissions without SATS Ops
%               File 2: Emission with SATS Ops, mixed fleet (30% SATS
SE, 30% SATS ME, 40% SATS VLJ)
%               File 3: Emission with SATS Ops, VLJ Only fleet
% Programed by Yue Xu, 09/21/05, version 1.0
% Updated by Yue Xu, 10/04/05, version 1.1: queueing time is added for
idle mode
% Updated by Yue Xu, 10/14/05, version 1.2: number of runways is
considered
% for queuing time and local operation split.

clear all;
clc;

load a2aSATS_pTripTable_Total_Redistributed_Business;
a2aSATS_pTrip_Biz_Des = sum(a2aSATS_pTripTable_Total_Redistributed); %
# of person business trips arriving the airport
a2aSATS_pTrip_Biz_Origin =
sum(a2aSATS_pTripTable_Total_Redistributed'); % # of person business
trips departing the airport

clear a2aSATS_pTripTable_Total_Redistributed_Business;

load a2aSATS_pTripTable_Total_Redistributed_NonBusiness;
a2aSATS_pTrip_NonBiz_Des =
sum(a2aSATS_pTripTable_Total_Redistributed); % # of person non-business
trips arriving the airport
a2aSATS_pTrip_NonBiz_Origin =
sum(a2aSATS_pTripTable_Total_Redistributed'); % # of person non-biz
trips departing the airport

clear a2aSATS_pTripTable_Total_Redistributed_NonBusiness;
```

```
PersonTrips = a2aSATS_pTrip_Biz_Des + a2aSATS_pTrip_Biz_Origin +
a2aSATS_pTrip_NonBiz_Des + a2aSATS_pTrip_NonBiz_Origin;
SATS_Ops = PersonTrips /2 / 2.4; % in the original file, the landing
and departing are sumed up, that means LTO cycles are
% half of the total SATS Ops, and a 2.4 load factor is assumed.

% Read data from the FAA Airport database
[Airport_ID Lon Lat Based_SE Based_ME Based_Jet Based_Helicopter
Air_Taxi GA_Local GA_itinerant Runway] = textread...
    ('SATS_Airport_Operations.txt','%s %f %f %f %f %f %f %f %f %f %d');

GA_Growth_Rate = 0.03;
Air_Taxi     = (Air_Taxi)*((1 + GA_Growth_Rate)^10);
GA_Local     = (GA_Local)*((1 + GA_Growth_Rate)^10);
GA_itinerant = (GA_itinerant)*((1 + GA_Growth_Rate)^10);

% Emission Rate Table (g/Kg) (gram per kg fule flow)%
% Format: [CO_Rate HC_Rate NOx_Rate SOx_Rate Fuel_Flow Time_in_mode]
% For TGO ops, format: [CO_Rate_per_operation HC_Rate_per_operation
% NOx_Rate_per_op SOx_Rate_per_op]; unit: g/operation
% For GSE, format: [CO_Rate_per_LTO HC_Rate_per_LTO VOC_Rate_per_LTO
NOx_Rate_per_LTO
% SOx_Rate_per_LTO PM10_Rate_per_LTO PM2.5_Rate_per_LTO]; unit:
g/operation

% Piston single-engine emission rate, Cessna 172 is used as the
representative aircraft
Piston_1E_Rate_Takeoff  = [1080 9.17 2.7 0.11 0.0168 1.75*60];
Piston_1E_Rate_Climbout = [961 9.5 4.3 0.11 0.0125 3.28*60];
Piston_1E_Rate_Approach = [995 11 3.7 0.11 0.00769 7.15*60];
Piston_1E_Rate_Idle     = [592 138 1.9 0.11 0.00145 0.15*60];
Piston_1E_Rate_TGO      = [7559.42  155.30  27.59   0.82];
Piston_1E_Rate_GSE      = [2.3 1.3 1.3 5.4 4.6 5.2 5.0];

% Piston twin-engine emission rate, Aztec is used as the representative
aircraft
Piston_2E_Rate_Takeoff  = [1442 12.4 0.36 0.11 0.03272 0.98*60];
Piston_2E_Rate_Climbout = [1471 16.6 0.235 0.11 0.02577 1.78*60];
Piston_2E_Rate_Approach = [1260 13.4 1.39 0.11 0.0125 5.18*60];
Piston_2E_Rate_Idle     =  [1294 68.08 0.39 0.11 0.00315 0.29*60];
Piston_2E_Rate_TGO      = [23577.83 501.54  13.59   1.85];
Piston_2E_Rate_GSE      = [2.3 1.3 1.3 5.4 4.6 5.2 5.0];

% TurboProp single-engine emission rate, 400A Hustler is used as the
representative aircraft
TurboProp_1E_Rate_Takeoff  = [5.1 1.75 7.98 0.54 0.0643 0.82*60];
TurboProp_1E_Rate_Climbout = [6.48 2.02 7.55 0.54 0.0596 1.1*60];
TurboProp_1E_Rate_Approach = [34.77 22.69 4.64 0.54 0.0344 8.1*60];
TurboProp_1E_Rate_Idle     = [115.12 101.46 1.96 0.54 0.0185 0.09*60];
TurboProp_1E_Rate_TGO      = [634.13    806.11  132.80  12.62];
TurboProp_1E_Rate_GSE      = [45.0 11.1 11.7 133.5 42.8 23.8 23.1];

% TurboProp twin-engine emission rate, Cessna Citation 441 is used as
the representative aircraft
TurboProp_2E_Rate_Takeoff  = [1.36 0.03 11.1 1 0.0566 0.9*60];
```

```
TurboProp_2E_Rate_Climbout = [1.56 0.04 10.8 1 0.0532 0.81*60];
TurboProp_2E_Rate_Approach = [4.35 0.19 9.49 1 0.0377 5.44*60];
TurboProp_2E_Rate_Idle     = [18.8 1.48 4.13 1 0.0144 0.13*60];
TurboProp_2E_Rate_TGO      = [127.55    10.65   358.09  36.11];
TurboProp_2E_Rate_GSE      = [35.5 8.4 8.9 110.1 33.5 17.0 16.5];

% Jet emission rate, MU300 is used as the representative aircraft
Jet_Rate_Takeoff  = [2.1 0.09 9.23 0.54 0.1697 0.93*60];
Jet_Rate_Climbout = [3.18 0.19 8.56 0.54 0.143 0.98*60];
Jet_Rate_Approach = [32 5.15 5.29 0.54 0.059 4.49*60];
Jet_Rate_Idle     = [97 40 2.63 0.54 0.0261 0.21*60];
Jet_Rate_GSE      = [11.7 3.9 4.1 28.8 14.0 12.0 11.7];

% SATS Aircraft emission rate table

% SATS Single-Engine aircraft, Socata Tampico is used as the
representative aircraft
SATS_SE_Rate_Takeoff  = [1192 11.42 1.82 0.11 0.01155 1.77*60];
SATS_SE_Rate_Climbout = [888.3 9.63 5.6 0.11 0.00773 3.72*60];
SATS_SE_Rate_Approach = [944.4 12.2 3.4 0.11 0.00474 9.06*60];
SATS_SE_Rate_Idle     = [618 36.1 1 0.11 0.00098 0.11*60];
SATS_SE_Rate_GSE      = [2.3 1.3 1.3 5.4 4.6 5.2 5.0];

% SATS Multi-Engine aircraft, Socata Tampico is used as the
representative aircraft
SATS_ME_Rate_Takeoff  = [3.6 0 6.5 1 0.075 0.82*60];
SATS_ME_Rate_Climbout = [5 0 6 1 0.068 1.1*60];
SATS_ME_Rate_Approach = [19.1 0.5 4.4 1 0.039 8.1*60];
SATS_ME_Rate_Idle     = [57.8 9.3 3.1 1 0.021 0.09*60];
SATS_ME_Rate_GSE      = [35.5 8.4 8.9 110.1 33.5 17 16.5];

% SATS VLJ aircraft, Socata Tampico is used as the representative
aircraft
SATS_VLJ_Rate_Takeoff  = [0.81 0.09 6.7714 0.54 0.14 1.08*60];
SATS_VLJ_Rate_Climbout = [0.97 0.1 10.9333 0.54 0.12 0.97*60];
SATS_VLJ_Rate_Approach = [5.23 0.14 2.2 0.54 0.03 4.52*60];
SATS_VLJ_Rate_Idle     = [42.3 5.94 1.595 0.54 0.012 0.22*60];
SATS_VLJ_Rate_GSE      = [11.7 3.9 4.1 28.8 14.0 12.0 11.7];


Piston_1E_Landing = 27.59;  % Single-Engine Piston annual landings from
GAATA, unit: million
Piston_2E_Landing = 2.52;  % Twin-Engine Piston annual landings from
GAATA, unit: million
TurboProp_1E_Landing = 1.37; % Single-Engine TurboProp annual landings
from GAATA, unit: million
TurboProp_2E_Landing = 2.63; % Twin-Engine TurboProp annual landings
from GAATA, unit: million
Jet_Landing = 2.18;  % Jet annual landings from GAATA, unit: million
Helicopter_Landing = 0;


Total_Landing = Piston_1E_Landing + Piston_2E_Landing +
TurboProp_1E_Landing + TurboProp_2E_Landing + Jet_Landing +
Helicopter_Landing;
```

```
% Assign taxi and queue time for the airport
% assumption: total ops = baseline ops + sats ops
% if total_ops / number of SATS qualified runways / 365 (daily) > 160,
queue time = 30 mins
% if total_ops / number of SATS qualified runways / 365 (daily) > 80,
queue time = 20 mins
% if total_ops / number of SATS qualified runways / 365 (daily) < 80,
queue time = 10 mins

for airport_number = 1:length(Airport_ID)

    if
(Air_Taxi(airport_number)+GA_Local(airport_number)+GA_itinerant(airport
_number)+SATS_Ops(airport_number)*2)/Runway(airport_number)/365>=200
        Queue_time(airport_number) = 30 * 60; % if total_ops / number
of runways > 200, queue time = 30*60 sec
        LTO_Cycles_Itinerant(airport_number) =
GA_itinerant(airport_number) / 2;
        LTO_Cycles_Local(airport_number) = 0.8 *
(GA_Local(airport_number)/2); % assign 80% of the local operations to
LTO Cycles
        LTO_Ops(airport_number) = LTO_Cycles_Itinerant(airport_number)
+ LTO_Cycles_Local(airport_number) + Air_Taxi(airport_number)/2;
        TGO_Ops(airport_number) = 0.2 * GA_Local(airport_number)/2; %
assign 20% of the local operations to TGO

    elseif (Air_Taxi(airport_number)+
GA_Local(airport_number)+GA_itinerant(airport_number)+SATS_Ops(airport_
number)*2)/Runway(airport_number)/365>=100
        Queue_time(airport_number) = 20 * 60; % if total_ops > 100 &
<200, queue time = 20*60 seconds
        LTO_Cycles_Itinerant(airport_number) =
GA_itinerant(airport_number) / 2;
        LTO_Cycles_Local(airport_number) = 0.5 *
(GA_Local(airport_number)/2); % assign 50% of the local operations to
LTO Cycles
        LTO_Ops(airport_number) = LTO_Cycles_Itinerant(airport_number)
+ LTO_Cycles_Local(airport_number) + Air_Taxi(airport_number)/2;
        TGO_Ops(airport_number) = 0.5 * GA_Local(airport_number)/2; %
assign 50% of the local operations to TGO

    else
        Queue_time(airport_number) = 10 * 60; % if total_ops < 100,
queue time = 10*60 seconds
        LTO_Cycles_Itinerant(airport_number) =
GA_itinerant(airport_number) / 2;
        LTO_Cycles_Local(airport_number) = 0.2 *
(GA_Local(airport_number)/2); % assign 50% of the local operations to
LTO Cycles
        LTO_Ops(airport_number) = LTO_Cycles_Itinerant(airport_number)
+ LTO_Cycles_Local(airport_number) + Air_Taxi(airport_number)/2;
        TGO_Ops(airport_number) = 0.8 * GA_Local(airport_number)/2; %
assign 50% of the local operations to TGO
    end % if
end % for airport_number = 1:length(Airport_ID)
```

```
clear airport_number;


Total_Ops = LTO_Ops + TGO_Ops;
Total_Ops = Total_Ops';

% Assign operations to each aircraft group
for airport_number = 1:length(Airport_ID)
    if Based_SE(airport_number) ~=0 | Based_ME(airport_number) ~=0 |
Based_Jet(airport_number) ~=0 | Based_Helicopter ~=0

        Weighted_average_SE(airport_number) =
Based_SE(airport_number)*(Piston_1E_Landing + TurboProp_1E_Landing);
        Weighted_average_ME(airport_number) =
Based_ME(airport_number)*(Piston_2E_Landing + TurboProp_2E_Landing);
        % No TGO ops for Helicopters and Jets
        Weighted_average_LTO(airport_number) =
Weighted_average_SE(airport_number) +
Weighted_average_ME(airport_number)...
            + Based_Helicopter(airport_number)*Helicopter_Landing +
Based_Jet(airport_number)*Jet_Landing;
        Weighted_average_TGO(airport_number) =
Weighted_average_SE(airport_number) +
Weighted_average_ME(airport_number);

        Piston_1E_LTO_Ops(airport_number) = (Piston_1E_Landing /
(Piston_1E_Landing + TurboProp_1E_Landing)) *...
            (Weighted_average_SE(airport_number) /
Weighted_average_LTO(airport_number)) * LTO_Ops(airport_number);
        Piston_1E_TGO_Ops(airport_number) = (Piston_1E_Landing /
(Piston_1E_Landing + TurboProp_1E_Landing)) *...
            (Weighted_average_SE(airport_number) /
Weighted_average_TGO(airport_number)) * TGO_Ops(airport_number);

        Piston_2E_LTO_Ops(airport_number) = (Piston_2E_Landing /
(Piston_2E_Landing + TurboProp_2E_Landing)) *...
            (Weighted_average_ME(airport_number) /
Weighted_average_LTO(airport_number)) * LTO_Ops(airport_number);
        Piston_2E_TGO_Ops(airport_number) = (Piston_1E_Landing /
(Piston_1E_Landing + TurboProp_1E_Landing)) *...
            (Weighted_average_ME(airport_number) /
Weighted_average_TGO(airport_number)) * TGO_Ops(airport_number);

        TurboProp_1E_LTO_Ops(airport_number) = (TurboProp_1E_Landing /
(Piston_1E_Landing + TurboProp_1E_Landing)) *...
            (Weighted_average_SE(airport_number) /
Weighted_average_LTO(airport_number)) * LTO_Ops(airport_number);
        TurboProp_1E_TGO_Ops(airport_number) = (TurboProp_1E_Landing /
(Piston_1E_Landing + TurboProp_1E_Landing)) *...
            (Weighted_average_SE(airport_number) /
Weighted_average_TGO(airport_number)) * TGO_Ops(airport_number);

        TurboProp_2E_LTO_Ops(airport_number) = (TurboProp_2E_Landing /
(Piston_2E_Landing + TurboProp_2E_Landing)) *...
```

```
            (Weighted_average_ME(airport_number) /
Weighted_average_LTO(airport_number)) * LTO_Ops(airport_number);
        TurboProp_2E_TGO_Ops(airport_number) = (TurboProp_2E_Landing /
(Piston_1E_Landing + TurboProp_2E_Landing)) *...
            (Weighted_average_ME(airport_number) /
Weighted_average_TGO(airport_number)) * TGO_Ops(airport_number);

        Jet_LTO_Ops(airport_number) = Jet_Landing /
Weighted_average_LTO(airport_number) * ...
            (LTO_Ops(airport_number) + TGO_Ops(airport_number));

%         Heli_LTO_Ops(airport_number) = Heli_Landing /
Weighted_average_LTO(airport_number) * ...
%             (LTO_Ops(airport_number) + TGO_Ops(airport_number)); %
Helicopters
    else

        % no based aircraft data available, use general percentage
        Piston_1E_LTO_Ops(airport_number) = Piston_1E_Landing /
Total_Landing * LTO_Ops(airport_number);
        Piston_1E_TGO_Ops(airport_number) = Piston_1E_Landing /
(Total_Landing-Jet_Landing-Helicopter_Landing) *
TGO_Ops(airport_number);
        Piston_2E_LTO_Ops(airport_number) = Piston_2E_Landing /
Total_Landing * LTO_Ops(airport_number);
        Piston_2E_TGO_Ops(airport_number) = Piston_2E_Landing /
(Total_Landing-Jet_Landing-Helicopter_Landing) *
TGO_Ops(airport_number);;

        TurboProp_1E_LTO_Ops(airport_number) = TurboProp_1E_Landing /
Total_Landing * LTO_Ops(airport_number);
        TurboProp_1E_TGO_Ops(airport_number) = TurboProp_1E_Landing /
(Total_Landing-Jet_Landing-Helicopter_Landing) *
TGO_Ops(airport_number);
        TurboProp_2E_LTO_Ops(airport_number) = TurboProp_2E_Landing /
Total_Landing * LTO_Ops(airport_number);
        TurboProp_2E_TGO_Ops(airport_number) = TurboProp_2E_Landing /
(Total_Landing-Jet_Landing-Helicopter_Landing) *
TGO_Ops(airport_number);

        Jet_LTO_Ops(airport_number) = Jet_Landing / Total_Landing *
LTO_Ops(airport_number);

%         Heli_LTO_Ops(airport_number) = Heli_Landing / Total_Landing *
LTO_Ops(airport_number);
    end % if Based_SE(airport_number) ~=0 | Based_ME(airport_number)
~=0 | Based_Jet(airport_number) ~=0 | Based_Helicopter ~=0
end % for airport_number = 1:length(Airport_ID)

clear airport_number;

% Calculate emissions at each airport
% ATTN: Idle emission = emission rate * fuel flow * (time in mode +
queue time)
```

```
% Piston single-engine emissions
Piston_1E_Annual_CO = Piston_1E_LTO_Ops * (Piston_1E_Rate_Takeoff(1) *
Piston_1E_Rate_Takeoff(5) * Piston_1E_Rate_Takeoff(6)...
    + Piston_1E_Rate_Climbout(1) * Piston_1E_Rate_Climbout(5) *
Piston_1E_Rate_Climbout(6)...
        + Piston_1E_Rate_Approach(1) * Piston_1E_Rate_Approach(5) *
Piston_1E_Rate_Approach(6)...
            + Piston_1E_Rate_Idle(1) * Piston_1E_Rate_Idle(5) *
Piston_1E_Rate_Idle(6))...
                + Piston_1E_TGO_Ops * Piston_1E_Rate_TGO(1)...
                    + Piston_1E_LTO_Ops .* (Piston_1E_Rate_Idle(1) *
Piston_1E_Rate_Idle(5) * Queue_time)...
                        + Piston_1E_TGO_Ops * Piston_1E_Rate_GSE(1);

Piston_1E_Annual_HC = Piston_1E_LTO_Ops * (Piston_1E_Rate_Takeoff(2) *
Piston_1E_Rate_Takeoff(5) * Piston_1E_Rate_Takeoff(6)...
    + Piston_1E_Rate_Climbout(2) * Piston_1E_Rate_Climbout(5) *
Piston_1E_Rate_Climbout(6)...
        + Piston_1E_Rate_Approach(2) * Piston_1E_Rate_Approach(5) *
Piston_1E_Rate_Approach(6)...
            + Piston_1E_Rate_Idle(2) * Piston_1E_Rate_Idle(5) *
Piston_1E_Rate_Idle(6))...
                + Piston_1E_TGO_Ops * Piston_1E_Rate_TGO(2)...
                    + Piston_1E_LTO_Ops .* (Piston_1E_Rate_Idle(2) *
Piston_1E_Rate_Idle(5) * Queue_time)...
                        + Piston_1E_TGO_Ops * Piston_1E_Rate_GSE(2);

Piston_1E_Annual_NOx = Piston_1E_LTO_Ops * (Piston_1E_Rate_Takeoff(3) *
Piston_1E_Rate_Takeoff(5) * Piston_1E_Rate_Takeoff(6)...
    + Piston_1E_Rate_Climbout(3) * Piston_1E_Rate_Climbout(5) *
Piston_1E_Rate_Climbout(6)...
        + Piston_1E_Rate_Approach(3) * Piston_1E_Rate_Approach(5) *
Piston_1E_Rate_Approach(6)...
            + Piston_1E_Rate_Idle(3) * Piston_1E_Rate_Idle(5) *
Piston_1E_Rate_Idle(6))...
                + Piston_1E_TGO_Ops * Piston_1E_Rate_TGO(3)...
                    + Piston_1E_LTO_Ops .* (Piston_1E_Rate_Idle(3) *
Piston_1E_Rate_Idle(5) * Queue_time)...
                        + Piston_1E_TGO_Ops * Piston_1E_Rate_GSE(4);

Piston_1E_Annual_SOx = Piston_1E_LTO_Ops * (Piston_1E_Rate_Takeoff(4) *
Piston_1E_Rate_Takeoff(5) * Piston_1E_Rate_Takeoff(6)...
    + Piston_1E_Rate_Climbout(4) * Piston_1E_Rate_Climbout(5) *
Piston_1E_Rate_Climbout(6)...
        + Piston_1E_Rate_Approach(4) * Piston_1E_Rate_Approach(5) *
Piston_1E_Rate_Approach(6)...
            + Piston_1E_Rate_Idle(4) * Piston_1E_Rate_Idle(5) *
Piston_1E_Rate_Idle(6))...
                + Piston_1E_TGO_Ops * Piston_1E_Rate_TGO(4)...
                    + Piston_1E_LTO_Ops .* (Piston_1E_Rate_Idle(4) *
Piston_1E_Rate_Idle(5) * Queue_time)...
                        + Piston_1E_TGO_Ops * Piston_1E_Rate_GSE(5);

Piston_1E_Emission =
[Piston_1E_Annual_CO;Piston_1E_Annual_HC;Piston_1E_Annual_NOx;Piston_1E
_Annual_SOx];
```

```
% Piston multi-engine emissions
Piston_2E_Annual_CO = Piston_2E_LTO_Ops * (Piston_2E_Rate_Takeoff(1) *
Piston_2E_Rate_Takeoff(5) * Piston_2E_Rate_Takeoff(6)...
    + Piston_2E_Rate_Climbout(1) * Piston_2E_Rate_Climbout(5) *
Piston_2E_Rate_Climbout(6)...
        + Piston_2E_Rate_Approach(1) * Piston_2E_Rate_Approach(5) *
Piston_2E_Rate_Approach(6)...
            + Piston_2E_Rate_Idle(1) * Piston_2E_Rate_Idle(5) *
Piston_2E_Rate_Idle(6))...
                + Piston_2E_TGO_Ops * Piston_2E_Rate_TGO(1)...
                    + Piston_2E_LTO_Ops .* (Piston_2E_Rate_Idle(1) *
Piston_2E_Rate_Idle(5) * Queue_time)...
                        + Piston_2E_TGO_Ops * Piston_2E_Rate_GSE(1);

Piston_2E_Annual_HC = Piston_2E_LTO_Ops * (Piston_2E_Rate_Takeoff(2) *
Piston_2E_Rate_Takeoff(5) * Piston_2E_Rate_Takeoff(6)...
    + Piston_2E_Rate_Climbout(2) * Piston_2E_Rate_Climbout(5) *
Piston_2E_Rate_Climbout(6)...
        + Piston_2E_Rate_Approach(2) * Piston_2E_Rate_Approach(5) *
Piston_2E_Rate_Approach(6)...
            + Piston_2E_Rate_Idle(2) * Piston_2E_Rate_Idle(5) *
Piston_2E_Rate_Idle(6))...
                + Piston_2E_TGO_Ops * Piston_2E_Rate_TGO(2)...
                    + Piston_2E_LTO_Ops .* (Piston_2E_Rate_Idle(2) *
Piston_2E_Rate_Idle(5) * Queue_time)...
                        + Piston_2E_TGO_Ops * Piston_2E_Rate_GSE(2);

Piston_2E_Annual_NOx = Piston_2E_LTO_Ops * (Piston_2E_Rate_Takeoff(3) *
Piston_2E_Rate_Takeoff(5) * Piston_2E_Rate_Takeoff(6)...
    + Piston_2E_Rate_Climbout(3) * Piston_2E_Rate_Climbout(5) *
Piston_2E_Rate_Climbout(6)...
        + Piston_2E_Rate_Approach(3) * Piston_2E_Rate_Approach(5) *
Piston_2E_Rate_Approach(6)...
            + Piston_2E_Rate_Idle(3) * Piston_2E_Rate_Idle(5) *
Piston_2E_Rate_Idle(6))...
                + Piston_2E_TGO_Ops * Piston_2E_Rate_TGO(3)...
                    + Piston_2E_LTO_Ops .* (Piston_2E_Rate_Idle(3) *
Piston_2E_Rate_Idle(5) * Queue_time)...
                        + Piston_2E_TGO_Ops * Piston_2E_Rate_GSE(4);

Piston_2E_Annual_SOx = Piston_2E_LTO_Ops * (Piston_2E_Rate_Takeoff(4) *
Piston_2E_Rate_Takeoff(5) * Piston_2E_Rate_Takeoff(6)...
    + Piston_2E_Rate_Climbout(4) * Piston_2E_Rate_Climbout(5) *
Piston_2E_Rate_Climbout(6)...
        + Piston_2E_Rate_Approach(4) * Piston_2E_Rate_Approach(5) *
Piston_2E_Rate_Approach(6)...
            + Piston_2E_Rate_Idle(4) * Piston_2E_Rate_Idle(5) *
Piston_2E_Rate_Idle(6))...
                + Piston_2E_TGO_Ops * Piston_2E_Rate_TGO(4)...
                    + Piston_2E_LTO_Ops .* (Piston_2E_Rate_Idle(4) *
Piston_2E_Rate_Idle(5) * Queue_time)...
                        + Piston_2E_TGO_Ops * Piston_2E_Rate_GSE(5);
```

```
Piston_2E_Emission =
[Piston_2E_Annual_CO;Piston_2E_Annual_HC;Piston_2E_Annual_NOx;Piston_2E
_Annual_SOx];

% TurboProp single-engine emissions
TurboProp_1E_Annual_CO = TurboProp_1E_LTO_Ops *
(TurboProp_1E_Rate_Takeoff(1) * TurboProp_1E_Rate_Takeoff(5) *
TurboProp_1E_Rate_Takeoff(6)...
    + TurboProp_1E_Rate_Climbout(1) * TurboProp_1E_Rate_Climbout(5) *
TurboProp_1E_Rate_Climbout(6)...
        + TurboProp_1E_Rate_Approach(1) * TurboProp_1E_Rate_Approach(5)
* TurboProp_1E_Rate_Approach(6)...
            + TurboProp_1E_Rate_Idle(1) * TurboProp_1E_Rate_Idle(5) *
TurboProp_1E_Rate_Idle(6))...
                + TurboProp_1E_TGO_Ops * TurboProp_1E_Rate_TGO(1)...
                    + TurboProp_1E_LTO_Ops .* (TurboProp_1E_Rate_Idle(1)
* TurboProp_1E_Rate_Idle(5) * Queue_time)...
                        + TurboProp_1E_TGO_Ops *
TurboProp_1E_Rate_GSE(1);

TurboProp_1E_Annual_HC = TurboProp_1E_LTO_Ops *
(TurboProp_1E_Rate_Takeoff(2) * TurboProp_1E_Rate_Takeoff(5) *
TurboProp_1E_Rate_Takeoff(6)...
    + TurboProp_1E_Rate_Climbout(2) * TurboProp_1E_Rate_Climbout(5) *
TurboProp_1E_Rate_Climbout(6)...
        + TurboProp_1E_Rate_Approach(2) * TurboProp_1E_Rate_Approach(5)
* TurboProp_1E_Rate_Approach(6)...
            + TurboProp_1E_Rate_Idle(2) * TurboProp_1E_Rate_Idle(5) *
TurboProp_1E_Rate_Idle(6))...
                + TurboProp_1E_TGO_Ops * TurboProp_1E_Rate_TGO(2)...
                    + TurboProp_1E_LTO_Ops .* (TurboProp_1E_Rate_Idle(2)
* TurboProp_1E_Rate_Idle(5) * Queue_time)...
                        + TurboProp_1E_TGO_Ops *
TurboProp_1E_Rate_GSE(2);

TurboProp_1E_Annual_NOx = TurboProp_1E_LTO_Ops *
(TurboProp_1E_Rate_Takeoff(3) * TurboProp_1E_Rate_Takeoff(5) *
TurboProp_1E_Rate_Takeoff(6)...
    + TurboProp_1E_Rate_Climbout(3) * TurboProp_1E_Rate_Climbout(5) *
TurboProp_1E_Rate_Climbout(6)...
        + TurboProp_1E_Rate_Approach(3) * TurboProp_1E_Rate_Approach(5)
* TurboProp_1E_Rate_Approach(6)...
            + TurboProp_1E_Rate_Idle(3) * TurboProp_1E_Rate_Idle(5) *
TurboProp_1E_Rate_Idle(6))...
                + TurboProp_1E_TGO_Ops * TurboProp_1E_Rate_TGO(3)...
                    + TurboProp_1E_LTO_Ops .* (TurboProp_1E_Rate_Idle(3)
* TurboProp_1E_Rate_Idle(5) * Queue_time)...
                        + TurboProp_1E_TGO_Ops *
TurboProp_1E_Rate_GSE(4);

TurboProp_1E_Annual_SOx = TurboProp_1E_LTO_Ops *
(TurboProp_1E_Rate_Takeoff(4) * TurboProp_1E_Rate_Takeoff(5) *
TurboProp_1E_Rate_Takeoff(6)...
    + TurboProp_1E_Rate_Climbout(4) * TurboProp_1E_Rate_Climbout(5) *
TurboProp_1E_Rate_Climbout(6)...
```

```
        + TurboProp_1E_Rate_Approach(4) * TurboProp_1E_Rate_Approach(5)
* TurboProp_1E_Rate_Approach(6)...
            + TurboProp_1E_Rate_Idle(4) * TurboProp_1E_Rate_Idle(5) *
TurboProp_1E_Rate_Idle(6))...
                + TurboProp_1E_TGO_Ops * TurboProp_1E_Rate_TGO(4)...
                    + TurboProp_1E_LTO_Ops .* (TurboProp_1E_Rate_Idle(4)
* TurboProp_1E_Rate_Idle(5) * Queue_time)...
                        + TurboProp_1E_TGO_Ops *
TurboProp_1E_Rate_GSE(5);

TurboProp_1E_Emission =
[TurboProp_1E_Annual_CO;TurboProp_1E_Annual_HC;TurboProp_1E_Annual_NOx;
TurboProp_1E_Annual_SOx];


% TurboProp multi-engine emissions
TurboProp_2E_Annual_CO = TurboProp_2E_LTO_Ops *
(TurboProp_2E_Rate_Takeoff(1) * TurboProp_2E_Rate_Takeoff(5) *
TurboProp_2E_Rate_Takeoff(6)...
    + TurboProp_2E_Rate_Climbout(1) * TurboProp_2E_Rate_Climbout(5) *
TurboProp_2E_Rate_Climbout(6)...
        + TurboProp_2E_Rate_Approach(1) * TurboProp_2E_Rate_Approach(5)
* TurboProp_2E_Rate_Approach(6)...
            + TurboProp_2E_Rate_Idle(1) * TurboProp_2E_Rate_Idle(5) *
TurboProp_2E_Rate_Idle(6))...
                + TurboProp_2E_TGO_Ops * TurboProp_2E_Rate_TGO(1)...
                    + TurboProp_2E_LTO_Ops .* (TurboProp_2E_Rate_Idle(1)
* TurboProp_2E_Rate_Idle(5) * Queue_time)...
                        + TurboProp_2E_TGO_Ops *
TurboProp_2E_Rate_GSE(1);

TurboProp_2E_Annual_HC = TurboProp_2E_LTO_Ops *
(TurboProp_2E_Rate_Takeoff(2) * TurboProp_2E_Rate_Takeoff(5) *
TurboProp_2E_Rate_Takeoff(6)...
    + TurboProp_2E_Rate_Climbout(2) * TurboProp_2E_Rate_Climbout(5) *
TurboProp_2E_Rate_Climbout(6)...
        + TurboProp_2E_Rate_Approach(2) * TurboProp_2E_Rate_Approach(5)
* TurboProp_2E_Rate_Approach(6)...
            + TurboProp_2E_Rate_Idle(2) * TurboProp_2E_Rate_Idle(5) *
TurboProp_2E_Rate_Idle(6))...
                + TurboProp_2E_TGO_Ops * TurboProp_2E_Rate_TGO(2)...
                    + TurboProp_2E_LTO_Ops .* (TurboProp_2E_Rate_Idle(2)
* TurboProp_2E_Rate_Idle(5) * Queue_time)...
                        + TurboProp_2E_TGO_Ops *
TurboProp_2E_Rate_GSE(2);

TurboProp_2E_Annual_NOx = TurboProp_2E_LTO_Ops *
(TurboProp_2E_Rate_Takeoff(3) * TurboProp_2E_Rate_Takeoff(5) *
TurboProp_2E_Rate_Takeoff(6)...
    + TurboProp_2E_Rate_Climbout(3) * TurboProp_2E_Rate_Climbout(5) *
TurboProp_2E_Rate_Climbout(6)...
        + TurboProp_2E_Rate_Approach(3) * TurboProp_2E_Rate_Approach(5)
* TurboProp_2E_Rate_Approach(6)...
            + TurboProp_2E_Rate_Idle(3) * TurboProp_2E_Rate_Idle(5) *
TurboProp_2E_Rate_Idle(6))...
                + TurboProp_2E_TGO_Ops * TurboProp_2E_Rate_TGO(3)...
```

```
                    + TurboProp_2E_LTO_Ops .* (TurboProp_2E_Rate_Idle(3)
* TurboProp_2E_Rate_Idle(5) * Queue_time)...
                        + TurboProp_2E_TGO_Ops *
TurboProp_2E_Rate_GSE(4);

TurboProp_2E_Annual_SOx = TurboProp_2E_LTO_Ops *
(TurboProp_2E_Rate_Takeoff(4) * TurboProp_2E_Rate_Takeoff(5) *
TurboProp_2E_Rate_Takeoff(6)...
    + TurboProp_2E_Rate_Climbout(4) * TurboProp_2E_Rate_Climbout(5) *
TurboProp_2E_Rate_Climbout(6)...
        + TurboProp_2E_Rate_Approach(4) * TurboProp_2E_Rate_Approach(5)
* TurboProp_2E_Rate_Approach(6)...
            + TurboProp_2E_Rate_Idle(4) * TurboProp_2E_Rate_Idle(5) *
TurboProp_2E_Rate_Idle(6))...
                + TurboProp_2E_TGO_Ops * TurboProp_2E_Rate_TGO(4)...
                    + TurboProp_2E_LTO_Ops .* (TurboProp_2E_Rate_Idle(4)
* TurboProp_2E_Rate_Idle(5) * Queue_time)...
                        + TurboProp_2E_TGO_Ops *
TurboProp_2E_Rate_GSE(5);

TurboProp_2E_Emission =
[TurboProp_2E_Annual_CO;TurboProp_2E_Annual_HC;TurboProp_2E_Annual_NOx;
TurboProp_2E_Annual_SOx];

% Jet engine emissions
Jet_Annual_CO = Jet_LTO_Ops * (Jet_Rate_Takeoff(1) * Jet_Rate_Takeoff(5)
* Jet_Rate_Takeoff(6)...
    + Jet_Rate_Climbout(1) * Jet_Rate_Climbout(5) *
Jet_Rate_Climbout(6)...
        + Jet_Rate_Approach(1) * Jet_Rate_Approach(5) *
Jet_Rate_Approach(6)...
            + Jet_Rate_Idle(1)  * Jet_Rate_Idle(5)  * Jet_Rate_Idle(6))...
                + Jet_LTO_Ops .* (Jet_Rate_Idle(1) * Jet_Rate_Idle(5) *
Queue_time)...
                    + Jet_LTO_Ops * Jet_Rate_GSE(1);

Jet_Annual_HC = Jet_LTO_Ops * (Jet_Rate_Takeoff(2) * Jet_Rate_Takeoff(5)
* Jet_Rate_Takeoff(6)...
    + Jet_Rate_Climbout(2) * Jet_Rate_Climbout(5) *
Jet_Rate_Climbout(6)...
        + Jet_Rate_Approach(2) * Jet_Rate_Approach(5) *
Jet_Rate_Approach(6)...
            + Jet_Rate_Idle(2)  * Jet_Rate_Idle(5)  * Jet_Rate_Idle(6))...
                + Jet_LTO_Ops .* (Jet_Rate_Idle(2) * Jet_Rate_Idle(5) *
Queue_time)...
                    + Jet_LTO_Ops * Jet_Rate_GSE(2);

Jet_Annual_NOx = Jet_LTO_Ops * (Jet_Rate_Takeoff(3) *
Jet_Rate_Takeoff(5) * Jet_Rate_Takeoff(6)...
    + Jet_Rate_Climbout(3) * Jet_Rate_Climbout(5) *
Jet_Rate_Climbout(6)...
        + Jet_Rate_Approach(3) * Jet_Rate_Approach(5) *
Jet_Rate_Approach(6)...
            + Jet_Rate_Idle(3)  * Jet_Rate_Idle(5)  * Jet_Rate_Idle(6))...
                + Jet_LTO_Ops .* (Jet_Rate_Idle(3) * Jet_Rate_Idle(5) *
Queue_time)...
```

```
                    + Jet_LTO_Ops * Jet_Rate_GSE(4);

Jet_Annual_SOx = Jet_LTO_Ops * (Jet_Rate_Takeoff(4) *
Jet_Rate_Takeoff(5) * Jet_Rate_Takeoff(6)...
    + Jet_Rate_Climbout(4) * Jet_Rate_Climbout(5) *
Jet_Rate_Climbout(6)...
        + Jet_Rate_Approach(4) * Jet_Rate_Approach(5) *
Jet_Rate_Approach(6)...
            + Jet_Rate_Idle(4)  * Jet_Rate_Idle(5)  * Jet_Rate_Idle(6))...
                + Jet_LTO_Ops .* (Jet_Rate_Idle(4) * Jet_Rate_Idle(5) *
Queue_time)...
                    + Jet_LTO_Ops * Jet_Rate_GSE(5);

Jet_Emission =
[Jet_Annual_CO;Jet_Annual_HC;Jet_Annual_NOx;Jet_Annual_SOx];

% Baseline Emission without SATS
Baseline_Emission = Piston_1E_Emission + Piston_2E_Emission +
TurboProp_1E_Emission + TurboProp_2E_Emission + Jet_Emission;



% SATS Single-Engine emissions, mixed fleet
SATS_SE_Annual_Mixed_CO = SATS_Ops * 0.3 * (SATS_SE_Rate_Takeoff(1) *
SATS_SE_Rate_Takeoff(5) * SATS_SE_Rate_Takeoff(6)...
    + SATS_SE_Rate_Climbout(1) * SATS_SE_Rate_Climbout(5) *
SATS_SE_Rate_Climbout(6)...
        + SATS_SE_Rate_Approach(1) * SATS_SE_Rate_Approach(5) *
SATS_SE_Rate_Approach(6)...
            + SATS_SE_Rate_Idle(1) * SATS_SE_Rate_Idle(5) *
SATS_SE_Rate_Idle(6)...
                + SATS_SE_Rate_GSE(1))... % Ground Support Equipments
emission
                    + SATS_Ops * 0.3 .* (SATS_SE_Rate_Idle(1) *
SATS_SE_Rate_Idle(5) * Queue_time);

SATS_SE_Annual_Mixed_HC = SATS_Ops * 0.3 * (SATS_SE_Rate_Takeoff(2) *
SATS_SE_Rate_Takeoff(5) * SATS_SE_Rate_Takeoff(6)...
    + SATS_SE_Rate_Climbout(2) * SATS_SE_Rate_Climbout(5) *
SATS_SE_Rate_Climbout(6)...
        + SATS_SE_Rate_Approach(2) * SATS_SE_Rate_Approach(5) *
SATS_SE_Rate_Approach(6)...
            + SATS_SE_Rate_Idle(2) * SATS_SE_Rate_Idle(5) *
SATS_SE_Rate_Idle(6)...
                + SATS_SE_Rate_GSE(2))... % Ground Support Equipments
emission
                    + SATS_Ops * 0.3 .* (SATS_SE_Rate_Idle(2) *
SATS_SE_Rate_Idle(5) * Queue_time);

SATS_SE_Annual_Mixed_NOx = SATS_Ops * 0.3 * (SATS_SE_Rate_Takeoff(3) *
SATS_SE_Rate_Takeoff(5) * SATS_SE_Rate_Takeoff(6)...
    + SATS_SE_Rate_Climbout(3) * SATS_SE_Rate_Climbout(5) *
SATS_SE_Rate_Climbout(6)...
        + SATS_SE_Rate_Approach(3) * SATS_SE_Rate_Approach(5) *
SATS_SE_Rate_Approach(6)...
```

```
            + SATS_SE_Rate_Idle(3) * SATS_SE_Rate_Idle(5) *
SATS_SE_Rate_Idle(6)...
                + SATS_SE_Rate_GSE(4))...% Ground Support Equipments
emission
                    + SATS_Ops * 0.3 .* (SATS_SE_Rate_Idle(3) *
SATS_SE_Rate_Idle(5) * Queue_time);

SATS_SE_Annual_Mixed_SOx = SATS_Ops * 0.3 * (SATS_SE_Rate_Takeoff(4) *
SATS_SE_Rate_Takeoff(5) * SATS_SE_Rate_Takeoff(6)...
    + SATS_SE_Rate_Climbout(4) * SATS_SE_Rate_Climbout(5) *
SATS_SE_Rate_Climbout(6)...
        + SATS_SE_Rate_Approach(4) * SATS_SE_Rate_Approach(5) *
SATS_SE_Rate_Approach(6)...
            + SATS_SE_Rate_Idle(4) * SATS_SE_Rate_Idle(5) *
SATS_SE_Rate_Idle(6)...
                + SATS_SE_Rate_GSE(5))... % Ground Support Equipments
emission
                + SATS_Ops * 0.3 .* (SATS_SE_Rate_Idle(4) *
SATS_SE_Rate_Idle(5) * Queue_time);

SATS_SE_Emission =
[SATS_SE_Annual_Mixed_CO;SATS_SE_Annual_Mixed_HC;SATS_SE_Annual_Mixed_N
Ox;SATS_SE_Annual_Mixed_SOx];


% SATS Multi-Engine emissions, mixed fleet
SATS_ME_Annual_Mixed_CO = SATS_Ops * 0.3 * (SATS_ME_Rate_Takeoff(1) *
SATS_ME_Rate_Takeoff(5) * SATS_ME_Rate_Takeoff(6)...
    + SATS_ME_Rate_Climbout(1) * SATS_ME_Rate_Climbout(5) *
SATS_ME_Rate_Climbout(6)...
        + SATS_ME_Rate_Approach(1) * SATS_ME_Rate_Approach(5) *
SATS_ME_Rate_Approach(6)...
            + SATS_ME_Rate_Idle(1) * SATS_ME_Rate_Idle(5) *
SATS_ME_Rate_Idle(6)...
                + SATS_ME_Rate_GSE(1))... % Ground Support Equipments
emission
                    + SATS_Ops * 0.3 .* (SATS_ME_Rate_Idle(1) *
SATS_ME_Rate_Idle(5) * Queue_time);

SATS_ME_Annual_Mixed_HC = SATS_Ops * 0.3 * (SATS_ME_Rate_Takeoff(2) *
SATS_ME_Rate_Takeoff(5) * SATS_ME_Rate_Takeoff(6)...
    + SATS_ME_Rate_Climbout(2) * SATS_ME_Rate_Climbout(5) *
SATS_ME_Rate_Climbout(6)...
        + SATS_ME_Rate_Approach(2) * SATS_ME_Rate_Approach(5) *
SATS_ME_Rate_Approach(6)...
            + SATS_ME_Rate_Idle(2) * SATS_ME_Rate_Idle(5) *
SATS_ME_Rate_Idle(6)...
                + SATS_ME_Rate_GSE(2))... % Ground Support Equipments
emission
                    + SATS_Ops * 0.3 .* (SATS_ME_Rate_Idle(2) *
SATS_ME_Rate_Idle(5) * Queue_time);

SATS_ME_Annual_Mixed_NOx = SATS_Ops * 0.3 * (SATS_ME_Rate_Takeoff(3) *
SATS_ME_Rate_Takeoff(5) * SATS_ME_Rate_Takeoff(6)...
    + SATS_ME_Rate_Climbout(3) * SATS_ME_Rate_Climbout(5) *
SATS_ME_Rate_Climbout(6)...
```

```
        + SATS_ME_Rate_Approach(3) * SATS_ME_Rate_Approach(5) *
SATS_ME_Rate_Approach(6)...
            + SATS_ME_Rate_Idle(3) * SATS_ME_Rate_Idle(5) *
SATS_ME_Rate_Idle(6)...
                + SATS_ME_Rate_GSE(4))...% Ground Support Equipments
emission
                    + SATS_Ops * 0.3 .* (SATS_ME_Rate_Idle(3) *
SATS_ME_Rate_Idle(5) * Queue_time);

SATS_ME_Annual_Mixed_SOx = SATS_Ops * 0.3 * (SATS_ME_Rate_Takeoff(4) *
SATS_ME_Rate_Takeoff(5) * SATS_ME_Rate_Takeoff(6)...
    + SATS_ME_Rate_Climbout(4) * SATS_ME_Rate_Climbout(5) *
SATS_ME_Rate_Climbout(6)...
        + SATS_ME_Rate_Approach(4) * SATS_ME_Rate_Approach(5) *
SATS_ME_Rate_Approach(6)...
            + SATS_ME_Rate_Idle(4) * SATS_ME_Rate_Idle(5) *
SATS_ME_Rate_Idle(6)...
                + SATS_ME_Rate_GSE(5))... % Ground Support Equipments
emission
                + SATS_Ops * 0.3 .* (SATS_ME_Rate_Idle(4) *
SATS_ME_Rate_Idle(5) * Queue_time);

SATS_ME_Emission =
[SATS_ME_Annual_Mixed_CO;SATS_ME_Annual_Mixed_HC;SATS_ME_Annual_Mixed_N
Ox;SATS_ME_Annual_Mixed_SOx];

% SATS VLJ emissions, mixed fleet
SATS_VLJ_Annual_Mixed_CO = SATS_Ops * 0.4 * (SATS_VLJ_Rate_Takeoff(1) *
SATS_VLJ_Rate_Takeoff(5) * SATS_VLJ_Rate_Takeoff(6)...
    + SATS_VLJ_Rate_Climbout(1) * SATS_VLJ_Rate_Climbout(5) *
SATS_VLJ_Rate_Climbout(6)...
        + SATS_VLJ_Rate_Approach(1) * SATS_VLJ_Rate_Approach(5) *
SATS_VLJ_Rate_Approach(6)...
            + SATS_VLJ_Rate_Idle(1) * SATS_VLJ_Rate_Idle(5) *
SATS_VLJ_Rate_Idle(6)...
                + SATS_VLJ_Rate_GSE(1))... % Ground Support Equipments
emission
                    + SATS_Ops * 0.4 .* (SATS_VLJ_Rate_Idle(1) *
SATS_VLJ_Rate_Idle(5) * Queue_time);

SATS_VLJ_Annual_Mixed_HC = SATS_Ops * 0.4 * (SATS_VLJ_Rate_Takeoff(2) *
SATS_VLJ_Rate_Takeoff(5) * SATS_VLJ_Rate_Takeoff(6)...
    + SATS_VLJ_Rate_Climbout(2) * SATS_VLJ_Rate_Climbout(5) *
SATS_VLJ_Rate_Climbout(6)...
        + SATS_VLJ_Rate_Approach(2) * SATS_VLJ_Rate_Approach(5) *
SATS_VLJ_Rate_Approach(6)...
            + SATS_VLJ_Rate_Idle(2) * SATS_VLJ_Rate_Idle(5) *
SATS_VLJ_Rate_Idle(6)...
                + SATS_VLJ_Rate_GSE(2))... % Ground Support Equipments
emission
                    + SATS_Ops * 0.4 .* (SATS_VLJ_Rate_Idle(2) *
SATS_VLJ_Rate_Idle(5) * Queue_time);

SATS_VLJ_Annual_Mixed_NOx = SATS_Ops * 0.4 * (SATS_VLJ_Rate_Takeoff(3)
* SATS_VLJ_Rate_Takeoff(5) * SATS_VLJ_Rate_Takeoff(6)...
```

```
    + SATS_VLJ_Rate_Climbout(3) * SATS_VLJ_Rate_Climbout(5) *
SATS_VLJ_Rate_Climbout(6)...
        + SATS_VLJ_Rate_Approach(3) * SATS_VLJ_Rate_Approach(5) *
SATS_VLJ_Rate_Approach(6)...
            + SATS_VLJ_Rate_Idle(3) * SATS_VLJ_Rate_Idle(5) *
SATS_VLJ_Rate_Idle(6)...
                + SATS_VLJ_Rate_GSE(4))...% Ground Support Equipments
emission
                    + SATS_Ops * 0.4 .* (SATS_VLJ_Rate_Idle(3) *
SATS_VLJ_Rate_Idle(5) * Queue_time);

SATS_VLJ_Annual_Mixed_SOx = SATS_Ops * 0.4 * (SATS_VLJ_Rate_Takeoff(4)
* SATS_VLJ_Rate_Takeoff(5) * SATS_VLJ_Rate_Takeoff(6)...
    + SATS_VLJ_Rate_Climbout(4) * SATS_VLJ_Rate_Climbout(5) *
SATS_VLJ_Rate_Climbout(6)...
        + SATS_VLJ_Rate_Approach(4) * SATS_VLJ_Rate_Approach(5) *
SATS_VLJ_Rate_Approach(6)...
            + SATS_VLJ_Rate_Idle(4) * SATS_VLJ_Rate_Idle(5) *
SATS_VLJ_Rate_Idle(6)...
                + SATS_VLJ_Rate_GSE(5))... % Ground Support Equipments
emission
                + SATS_Ops * 0.4 .* (SATS_VLJ_Rate_Idle(4) *
SATS_VLJ_Rate_Idle(5) * Queue_time);

SATS_VLJ_Emission =
[SATS_VLJ_Annual_Mixed_CO;SATS_VLJ_Annual_Mixed_HC;SATS_VLJ_Annual_Mixe
d_NOx;SATS_VLJ_Annual_Mixed_SOx];

% SATS Emission with mixed fleet
SATS_Mixed_Emission = SATS_SE_Emission + SATS_ME_Emission +
SATS_VLJ_Emission;

% SATS Emission, VLJ Only Fleet

% SATS VLJ emissions, mixed fleet
SATS_VLJonly_CO = SATS_Ops * (SATS_VLJ_Rate_Takeoff(1) *
SATS_VLJ_Rate_Takeoff(5) * SATS_VLJ_Rate_Takeoff(6)...
    + SATS_VLJ_Rate_Climbout(1) * SATS_VLJ_Rate_Climbout(5) *
SATS_VLJ_Rate_Climbout(6)...
        + SATS_VLJ_Rate_Approach(1) * SATS_VLJ_Rate_Approach(5) *
SATS_VLJ_Rate_Approach(6)...
            + SATS_VLJ_Rate_Idle(1) * SATS_VLJ_Rate_Idle(5) *
SATS_VLJ_Rate_Idle(6)...
                + SATS_VLJ_Rate_GSE(1))...
                    + SATS_Ops .* (SATS_VLJ_Rate_Idle(1) *
SATS_VLJ_Rate_Idle(5) * Queue_time);

SATS_VLJonly_HC = SATS_Ops * (SATS_VLJ_Rate_Takeoff(2) *
SATS_VLJ_Rate_Takeoff(5) * SATS_VLJ_Rate_Takeoff(6)...
    + SATS_VLJ_Rate_Climbout(2) * SATS_VLJ_Rate_Climbout(5) *
SATS_VLJ_Rate_Climbout(6)...
        + SATS_VLJ_Rate_Approach(2) * SATS_VLJ_Rate_Approach(5) *
SATS_VLJ_Rate_Approach(6)...
            + SATS_VLJ_Rate_Idle(2) * SATS_VLJ_Rate_Idle(5) *
SATS_VLJ_Rate_Idle(6)...
                + SATS_VLJ_Rate_GSE(2))...
```

```
                    + SATS_Ops .* (SATS_VLJ_Rate_Idle(2) *
SATS_VLJ_Rate_Idle(5) * Queue_time);

SATS_VLJonly_NOx = SATS_Ops * (SATS_VLJ_Rate_Takeoff(3) *
SATS_VLJ_Rate_Takeoff(5) * SATS_VLJ_Rate_Takeoff(6)...
    + SATS_VLJ_Rate_Climbout(3) * SATS_VLJ_Rate_Climbout(5) *
SATS_VLJ_Rate_Climbout(6)...
        + SATS_VLJ_Rate_Approach(3) * SATS_VLJ_Rate_Approach(5) *
SATS_VLJ_Rate_Approach(6)...
            + SATS_VLJ_Rate_Idle(3) * SATS_VLJ_Rate_Idle(5) *
SATS_VLJ_Rate_Idle(6)...
                + SATS_VLJ_Rate_GSE(4))...
                    + SATS_Ops .* (SATS_VLJ_Rate_Idle(3) *
SATS_VLJ_Rate_Idle(5) * Queue_time);

SATS_VLJonly_SOx = SATS_Ops * (SATS_VLJ_Rate_Takeoff(4) *
SATS_VLJ_Rate_Takeoff(5) * SATS_VLJ_Rate_Takeoff(6)...
    + SATS_VLJ_Rate_Climbout(4) * SATS_VLJ_Rate_Climbout(5) *
SATS_VLJ_Rate_Climbout(6)...
        + SATS_VLJ_Rate_Approach(4) * SATS_VLJ_Rate_Approach(5) *
SATS_VLJ_Rate_Approach(6)...
            + SATS_VLJ_Rate_Idle(4) * SATS_VLJ_Rate_Idle(5) *
SATS_VLJ_Rate_Idle(6)...
                + SATS_VLJ_Rate_GSE(5))...
                    + SATS_Ops .* (SATS_VLJ_Rate_Idle(4) *
SATS_VLJ_Rate_Idle(5) * Queue_time);

SATS_VLJ_only_Emission =
[SATS_VLJonly_CO;SATS_VLJonly_HC;SATS_VLJonly_NOx;SATS_VLJonly_SOx];

Baseline_Emission = Baseline_Emission' / 1E6;           % Baseline
emission in tonnage
SATS_Mixed_Emission = SATS_Mixed_Emission' / 1E6;       % SATS Mixed
fleet emission in tonnage
SATS_VLJ_only_Emission = SATS_VLJ_only_Emission' / 1E6; % SATS VLJ Only
fleet emission in tonnage

% Percent of emission increase introduced by SATS Ops

SATS_Ops = SATS_Ops';

for airport_number = 1:length(Airport_ID)
    if Baseline_Emission(airport_number) == 0 % no ops info, so
emission is 0
        Emission_Increase(airport_number,1:4) = [0 0 0 0];
        Operation_Increase(airport_number,:) = [0];
    else
        Emission_Increase(airport_number,:) =
SATS_VLJ_only_Emission(airport_number,:) ./Baseline_Emission(airport_nu
mber,:);
        Operation_Increase(airport_number,:) =
SATS_Ops(airport_number,:) ./Total_Ops(airport_number,:);
    end % if
end % for
```

```
Emission_Increase = Emission_Increase * 100;
Operation_Increase = Operation_Increase * 100;
Ops_with_SATS = SATS_Ops + Total_Ops;

save Emission_Output Airport_ID Lon Lat Baseline_Emission
SATS_Mixed_Emission SATS_VLJ_only_Emission...
    Emission_Increase Operation_Increase SATS_Ops Ops_with_SATS;

clear;

load Emission_Output;
```