A MODEL TO ASSESS THE MOBILITY OF THE NATIONAL AIRPSPACE SYSTEM (NAS).

By

Anand Seshadri

A thesis document submitted in partial fulfillment of the

requirements for the degree of Master of Science in

Civil Engineering

Virginia Polytechnic Institute & State University

Blacksburg, VA

17th December, 2003

Committee:

Dr.Antonio A Trani, Chair

Dr.Hojong Baik, Co-Chair

Dr Dusan Teodorovic.

Keywords: Travel Time, Mobility, Passenger Hours

A MODEL TO ASSESS THE MOBILITY OF THE NATIONAL AIRPSPACE SYSTEM (NAS).

By

Anand Seshadri

ABSTRACT

Mobility in a transportation system can be defined as travel time for all travelers using the transportation network. A good assessment of the mobility is essential for knowing the points of congestion in the network and the factors responsible for the congestion. Also the change in mobility from the baseline to the horizon year would give the modeler an idea of the effectiveness of the various transportation systems. One of the applications of the mobility measurement is the evaluation of aviation technologies proposed by FAA to ease the congestion. This paper addresses a method to estimate the mobility of the air transportation network in the baseline year (2000). Also presented is a method to estimate the mobility to the horizon year by considering congestion on the roadway.

ACKNOWLEDGEMENTS

This thesis document is the culmination of all the work that has been performed by me over the last 12 months. However this work could not have been completed without the valuable help and guidance that I got from my professors, family and friends. In this section I express my thanks to all the people who have contributed to the success of this project

First of all I would like to extend my thanks to Dr Antonio Trani for his valuable guidance and help provided to me over the course of the project. He is a very inspiring person and it is a privilege to work with him. I thank Dr Hojong Baik whose ideas and inspiration were very useful to my research. I also thank Dr Dusan Teodorovic for his inputs and guidance.

Next I thank my brother Arvind Seshadri who helped me settle down in the USA and who was a constant source of encouragement and support for the past two years. I thank my parents whose constant love and care have enabled me to face the difficulties and challenges during the course of my graduate study

Last but not the least I thank all of my friends at Virginia Tech without whom I could have done my work in half the time. I am grateful to Kane and Krishna for all the entertaining discussions that we had. and for their encouragement and advice. I also thank Aimee for all her help and making my stay in Virginia Tech a pleasure.

CONTENTS

TITLE PAGE NO

ABSTRACT

ACKNOWLEDGEMENTS

LIST OF TABLES AND FIGURES

INTRODUCTION 1

- Background 1
- New technologies in aviation 4
- Research scope 6
- Organization of the document 6

LITERATURE REVIEW 7

- Introduction 7
- Aircraft performance 8
- Description of the BADA model 16
- Comparison of the BADA and Virginia Tech fuel models 21
- Transportation systems modeling process 23
- Ground network analysis 25

METHODOLOGY AND DATA SOURCES **32**

- Introduction 32
- Data sources used in the model 32

MODEL DESCRIPTION 44

- Introduction 44
- Variables used in the model 44
- Model Assumptions 50
- Flight trajectory model 52
- Commercial aviation mobility model 60
- Network analysis model 61
- Horizon year modeling 67

MODEL RESULTS 70

- Model results 70

CONCLUSIONS AND RECOMMENDATIONS 91

- Conclusions and recommendations 91

BIBILIOGRAPHY 94

APPENDIX A 96

LIST OF FIGURES PAGE

Figure 1.1 Growth of Enplanements in the United States (TAF) …….3
Figure 2.1 Typical Thrust Variation for a Turbofan Engine……...10
Figure 2.2 Aircraft in the climb phase……11
Figure 2.3 Aircraft in Cruise Mode…. ……..13
Figure 2.4 Descent Fuel Consumption Curves……..15
Figure 2.5 Comparison of the Neural Network and the BADA Model……...22
Figure 2.6 Trip Generation…… 24
Figure 2.7 Connectivity of two networks……...26
Figure 2.8 Schematic Representation of the Aircraft Choice Model……..30
Figure 3.1 American Travel Survey……...34
Figure 3.2 NPTS Data……..36
Figure 3.3 Census MSA Population Density Data……38
Figure 3.4 HPMS Data……40
Figure 3.5 DB1B Coupon Data……...42
Figure 4.1 Relationship between Speed and MSA Population……45
Figure 4.2 Variation of Travel Speed with Population Density……..46
Figure 4.3 Variation of Travel Speed with Travel Distance……47
Figure 4.4 Histogram of MSA vs. Non-MSA Speeds……48
Figure 4.5 Peak and Non-Peak Characteristics for MSA Regions…….49
Figure 4.6 BADA Performance Table File (PTF)…….53
Figure 4.7 General Aviation Travel Time Model…….55
Figure 4.8 Distance to Reach a Particular Altitude for the Cessna Citation……..56
Figure 4.9 Comparisons of the BADA and Bruguet Range Equations for an A-310 in Cruise Mode……57
Figure 4.10 Variation of the Composite Index with Height…….58
Figure 4.11 Distance to Descend from a Particular Altitude to Ground for the Airbus A 310……59
Figure 4.12 Population Density vs. Road Density Plot for the State of California……62
Figure 4.13 Trip Length Histogram from NPTS……..64
Figure 4.14 Trip Length Distributions for MSA Airport Trips……..65
Figure 4.15 Network Analysis Model……66
Figure 4.16 Variation of Speed with Time for Some MSAs……68
Figure 4.17 Variation of Speed with Population Density and the Corresponding Curve Fit……69
Figure 5.1 Cumulative Density Function for 1995 MSA Peak Characteristic……...71
Figure 5.2 Cumulative Density Function for 1995 MSA Non-Peak Characteristic…..72
Figure 5.3 Cumulative Density Function for 1995 Non-MSA Areas……. 73
Figure 5.4 Cumulative Density Function for 2020 MSA Peak Characteristic……...74
Figure 5.5 Cumulative Density Function for 2020 MSA Non-Peak Characteristic…...75
Figure 5.6 Cumulative Density Function for 2020 Non-MSA Areas……..76
Figure 5.7 Air Passenger flows in the United States…….77
Figure 5.8 Pie chart of Travel times across NAS……..78
Figure 5.9 MSA peak access time (Macroscopic approach)…….84
Figure 5.10 MSA non-peak access time (Macroscopic approach)……..85

Figure 5.11 Non-MSA access time (Macroscopic approach)……..86
Figure 5.12 MSA peak egress time (Macroscopic approach)……..87

LIST OF TABLES PAGE

Table 2.1 Level of Service Benchmarks……27
Table 5.1 1995 MSA Non-peak Speed CDFs……79
Table 5.2 1995 MSA Peak Speed CDF……..80
Table 5.3 2020 MSA Peak Speed CDFs……81
Table 5.4 2020 MSA Non-peak Speed CDFs……82
Table 5.5 1995 and 2020 Non-MSA Speed CDFs……83
Table 5.6 MSA Access times……..88
Table 5.7 MSA Egress times……..88
Table 5.8 Non-MSA Access times……89
Table 5.9 Non-MSA Egress times……90

CHAPTER 1

Introduction

1.1 Background

The air transportation system is an essential part of the transportation infrastructure of the United States. The air transportation system has played a major role in the economic expansion of the US. This expansion has in turn placed increasing demand on the transportation system which is quickly becoming saturated. This will lead to increasing delays and lower level of service across NAS.Therefore the the problem of optimizing the usage of existing infrastructure has assumed greater importance.This cannot be done unless one knows the level of usage of the system, the points of congestion and future trends in the system. To address this problem, NASA plans an air transportation system that will reduce the intercity travel time by half in the next ten years and by two-thirds by 2025 and reduce long-haul transcontinental travel time by half by 2025. In order to achieve this goal one must know the structure and organization of the future transportation system. This will require considerable changes to the present hub and spoke system and should include various alternatives such as the Small Aircraft Transportation System(SATS), Runway independent aircraft(RIA) etc.

1.1.1 Hub and Spoke system

Most of the commercial airlines today use the Hub and spoke system. A hub is a central airport where

flights are routed through and the spokes are the routes that planes take out of the hub airport. Most of the major airlines have many hubs because it enables them to fly greater number of routes. The hub and spoke system was started by the airlines after the Federal Government deregulated the airlines in 1978. Before deregulation the airlines were forced to fly between a fixed origin and destination which is called a point-to-point service. This resulted in most flights being half-empty and the airlines losing money. After deregulation most of the airlines have a central hub through which they route most of their flights. The advantage of the hub and spoke system is that the cost per seat mile is lowered since all the flights have greater number of passengers and offers the flier greater choices of routes.

1.1.2 Future Air Transportation System

The hub and spoke system has worked very well from 1978. Airlines have increased their profits and the level of service has also improved. Unfortunately the increasing amounts of air travel driven by economic growth has lead to increasing traffic at all the major hubs which are approaching their capacity. In a few years we will begin to experience long delays and breakdown of service at all the major hubs. Construction of new airport terminals also is not a feasible option because of economical, political and environmental concerns. Therefore the only way to reduce delays and improve service is to divert traffic away from the major hubs to small airports which have low traffic volumes. This means that the air transportation system will have to move partly from the existing hub and spoke system to the point to point system that existed before deregulation. The Small Air Transportation System(SATS) is a system where passengers will board a small aircraft at their origin point and fly directly to their destination thus bypassing the major airline hubs and reducing the demand at the hub airports. The effects of new transportation modes like SATS on air transportation system will require detailed analysis.

1.1.3 National Airspace System (NAS)

The National Airspace System is a complex collection of aircraft, facilities,procedures and systems. The aim of the NAS is to provide a safe environment for operation of the aircraft. The NAS includes resources such as aircraft, air traffic controllers, pilots, terminal area, and enroute airspace. The NAS has about 640 million enplanements in 2000, 19000 airports, 15000 air traffic controllers, 171 terminal radar control facilities.

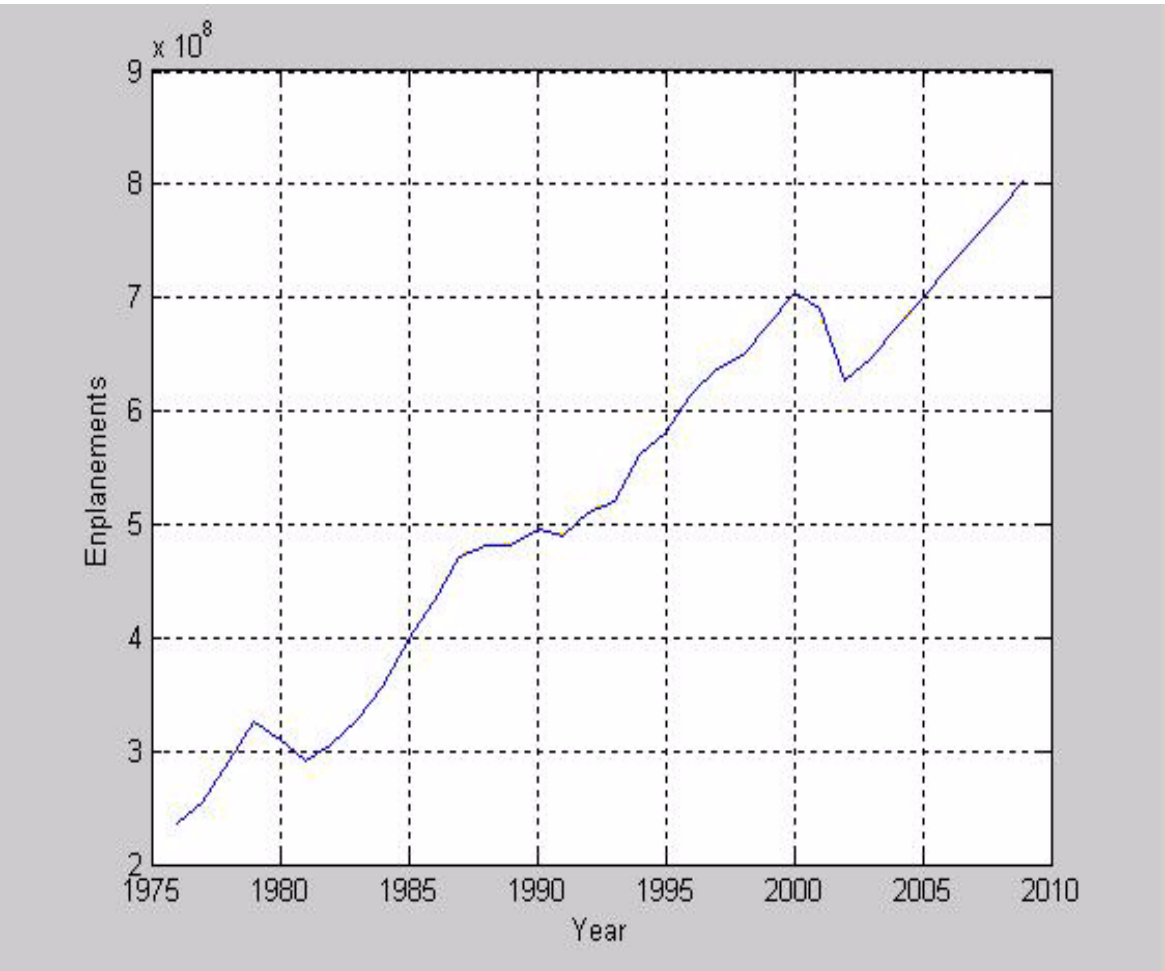


Figure 1.1 Growth of Enplanements in the United States (TAF).

1.1.4 New Technologies in Aviation

To ease the congestion in NAS due to increased volume of air traffic, a number of new concepts and technologies are being proposed. The impact of these new technologies on the transporation system and their role in reducing the congestion warrant careful analysis. The various new technologies being introduced are

1.1.4.1 User Request Evaluation Tools (URET).

URET was developed in response to concerns that the present airspace system is quickly becoming saturated and is not keeping in step with the increasing demand for air travel. Current Air Traffic rules while emphasizing on safety are quite restrictive and do not allow the pilot to fly optimal flight paths. Therefore the concept of free-flight was developed where the pilot would not fly structured preplanned routes but take the most optimal path. The pilots and ground controllers share information about flight paths in order to achieve an acceptable safety standard. This requires a revolutionary change in the information sharing process and architecture. As a part of designing the revolutionary information sharing processes the first free-flight model, Free Flight Phase I was based on the User Request Evaluation Tool (URET). The User Request Evaluation Tool, or URET, was developed at MITRE's Center for Advanced Aviation System Development (CAASD) to assist controllers with timely detection and resolution of predicted problems. By helping to manage workload and to allow more strategic planning, URET helps the system support a greater number of user-preferred flight profiles, increased user flexibility, and increased system capacity while maintaining the level of safety. URET processes real-time flight plan and track data with site adaptation, aircraft performance characteristics, and temperature and wind data to build four-dimensional flight profiles, or trajectories, for all flights within a facility or inbound to it. When a conflict (i.e., possible loss of separation) is detected, URET determines which sector to notify and displays an alert to that sector up to 20 minutes prior to the conflict. This longer look-ahead gives controllers more time for strategic planning. In addition to conflict detection capabilities, URET has introduced a new controller interface that supports flight data management and task prioritization at the sector using both textual and graphic displays.

1.1.4.2 Controller Pilot Datalink Communication (CPDLC). The current air traffic control system relies heavily on voice communication between pilots and controllers to relay information critical to the safe operation of the flight like flight paths, take off and landing schedules and aircraft separation. As air traffic increases, the voice channel between pilots and controllers also keeps getting saturated especially at peak locations at peak hours at large hub airports.The problem is acute when one realizes that a communication between a pilot and a controller is like a conference call with the controller and all the pilots flying the same region of space communicating on the same frequency. If two parties in a conference call speak at the same time, with neither one knowing that he interfered with the communication of the other, there is a lot of time wasted with no information recieved by either party. In this case it takes a lot more airtime to communicate than in the case where there is no interference and this problem multiplies with the number of parties involved in the call.This precisely describes the scenario in which the ATC voice channel is saturated. It is not usual for pilots and controller to 'step on' each other's communications. This wastes a lot time on the ATC voice channel as repeated communication attempts are made and seriously hinders the efficiency of the ATC and the pilots, resulting in a significant increase in delays.CPDLC was developed to ease the load on the vital ATC-pilot voice channel by off-loading the non-vital communications between the pilot and controller to a data channel thus freeing the voice channel for critical real-time information.In FAA conducted simulations CPDLC reduced the occupancy of the voice channel by 75 percent. This is turn increased the efficiency of the ATC and the safety of the airspace.

1.1.4.3 Runway independent aircraft (RIA). These vehicles use Vertical take-off and landing technology (VTOL),based on the helicopter to operate without the use of runways.These vehicles can greatly ease the load on saturated runways, thereby reducing delays and increase the capacity of large hub airports. However these vehicles will not be very large and can carry 40 to 50 passengers at a time.They operate from small airports, therefore passengers need not go to large hub airports to take a flight to their destination,further reducing the congestion at large airports.These vehicles are still in the process of development and it is expected that they will be in service in another 10-15 years.

1.2) Research Scope

The mobility study outlined in this document, measures the mobility of air passengers in the US in the baseline year(2000) and describes a method to predict mobility in the year 2020. Measuring mobility in the present and future years would give the modeler an idea of the efficiency of the network. This document calculates the door-to-door to door travel time for passengers taking General Aviation (GA) and Commercial Airline (CA) given the passenger O-D Tables for both the modes.

1.3) Organization of the Document

This document is organized as follows. In the second chapter a review of existing methods of travel time prediction and other models used in the analysis is presented. In the next chapter the data sources used in the analysis is presented. In the following chapter the model is described in detail. In the concluding chapter the results from the analysis is presented. The various source codes used in the analysis are described in the Appendix.

CHAPTER 2 Literature Review

2.1 Introduction

The number of enplanments are going to double in the next 20 years and the air transportation system will get increasingly congested. To ease this congestion NASA has proposed a set of new aviation technologies like URET,CPDLC etc. To determine the impact and worthiness of these new technologies, one must first measure the mobility of the air transportation network in the present year and the future year and hence the increase in network congestion and decide whether it is feasible to implement the proposed technologies. There a number of ways to measure transportation network mobility in a region. The most traditional way is to estimate travel time from door to destination.The assumption here is that shorter average travel time would imply a better network mobility for all users. Since the typical journey by the air mode consists of both flying time and driving time from door-to-airport and airport-to-destination, a good estimate of driving times in necessary to have a fairly accurate measure of the mobility of the air transportation network. The driving time would depend on the demand side variables like socio-economic characteristics of the region like population density and the supply side variables such as road density. The air travel time would depend on the type of aircraft flown, the distance between the two airports and the kind of trajectory followed. Therefore for a good estimate of the air travel time a proper understanding of aircraft performance in the climb,cruise and descent phases is

essential.This is described in the next section.

2.2 Aircraft Performance

The typical aircraft trajectory consists of three phases-climb,cruise and descent. In the climb portion the aircraft uses the lift generated by the wings to overcome atmostpheric drag and its own weight to reach an optimum altitude where the the fuel consumed is minimum and the true airspeed (TAS) is maximum. In the cruise portion, where the aircraft spends most of flying time and most of the fuel is consumed, the aircraft is in steady state where the lift balances the weight and the drag balances the thrust. In the descent portion which is usually the final portion of the flight portion, the aircraft trajectory is dictated by the ATC and adjacent aircraft than the aerodynamics of the aircraft. Therefore the forces the aircraft faces during all three phases are lift,weight,thrust and drag.

2.2.1 Lift

Lift is the force that opposes the aircraft weight and is responsible for the aircraft staying up in the air. The lift is generated from the Bernuolli's principle which states when an object moves through a fluid it creates a low pressure around it. Mathematically this can be written as.

$$P + \frac{1}{2}\rho V^2 = P_0 \tag{2.1}$$

where V is the velocity of the body, P_0 is the rest or zero-velocity pressure, P is the pressure when velocity of the fluid is V. The difference between the two pressures generates the lifting force.

2.2.2 Drag.

Drag is a force that offers resistance to the motion of a body through a fluid. It can be thought as the fluid counterpart of friction. Drag occurs because of velocity differential between the object and fluid. When the object moves the fluid it breaks the fluid streamlines into pieces called eddies.These eddies are one of the main of the reasons for the drag. The wing shape of the aircraft is such that there are no eddies created and therefore the drag is low. Therefore it can be seen that drag depends on the velocity of the object, shape of the object and density of the fluid. The expression for the drag force is:

$$D = \frac{1}{2}\rho V^2 S C_D \tag{2.2}$$

where D is the drag, V is the velocity, r is the density of the fluid, S is the area of contact, C_D is a dimensionless constant called the drag coefficient. The drag coefficient depends on properties of the body like the roughness and shape. Typically drag consists of two components- skin friction drag(drag due to friction between the body and fluid) and drag due to induced drag(drag that occurs due to lift generated). This relationship is called the drag-polar and can be written mathematically as:

$$C_D = \left(C_{D0} + \frac{C_L^2}{\pi A R e}\right) \tag{2.3}$$

where C_D is the drag, C_{D0} is the zero-lift drag coefficient(or skin-friction drag coefficient), C_L is the lift coefficient, AR is the aspect ratio of the wing(the square of the wing-length divided by the area of the wing).

2.2.3 Thrust

Tractive forces are generated principally from the thrust produced by a turbofan or propeller-powered engine.The engine thrust produced is a function of the airspeed and density of the surrounding air. The thrust variation for a typical turbofan engine is shown below:

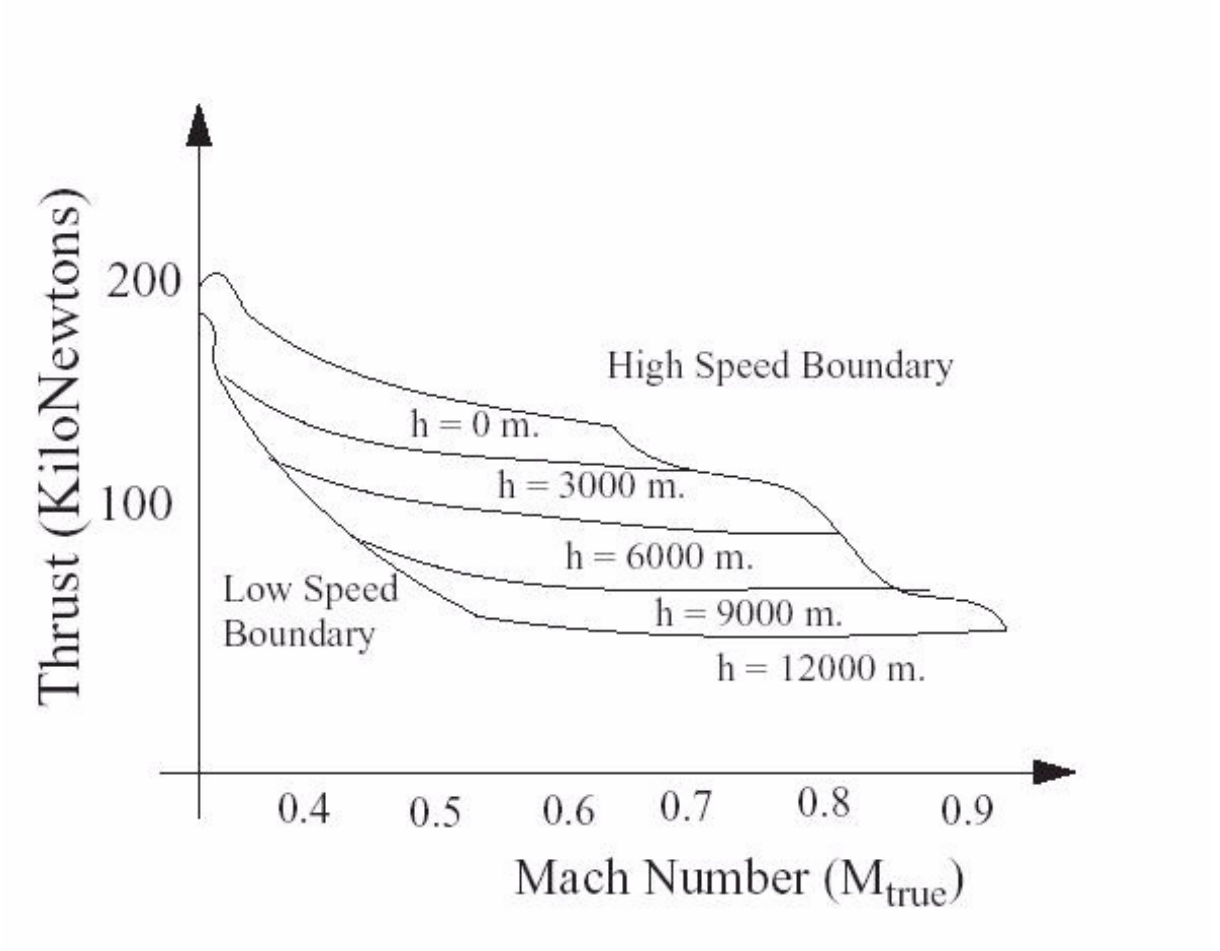


Figure 2.1 Typical Thrust Variation for a Turbofan Engine (16).

The above figure indicates that the aircraft engines have prescribed boundaries of operation. When the thrust and drag curves intersect at high mach number, that is the maximum mach number that the aircraft can reach. For a turbofan-driven aircraft, the maximum mach number ranges from 0.78 to 0.92. For a propeller driven aircraft this value is around 0.55 due to the compressability effects from the propeller. On the other hand there is a minimum value of mach number that the aircraft has to maintain in order to stay up in the air. This is because there is an upper limit to the lift coefficient C_L and also at low mach numbers unsteady flows behind the wings destroy any lifting force that is generated.

2.2.4 Climb performance

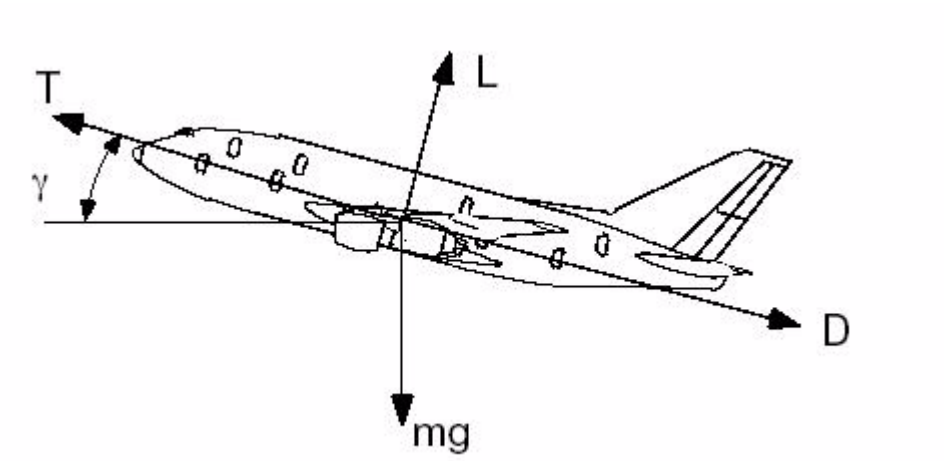


Figure 2.2 Aircraft in the Climb Phase (16).

Considering force equilibrium,along the flight path.

$$m\frac{\delta V}{\delta t} = T \angle D \angle mg\sin\gamma \tag{2.4}$$

where m is the mass of the vehicle,T and D are thrust and drag forces and g is the flight path angle. Normal to flight path,

$$m\frac{\delta\gamma}{\delta t}V = L \angle mg\cos\gamma \tag{2.5}$$

Generally it is assumed that the term rate of change in flight angle is very small so it can be neglected. Simplifying equation 2.4 we get,

$$\sin\gamma = \frac{T \angle D}{mg} \angle \frac{1}{g}\frac{\delta V}{\delta t} \tag{2.6}$$

the first term in the above expression accounts for possible changes in the potential state of the vehicle and the second term is the acceleration of the vehicle while in the climb mode. In general aircraft climb at constant IAS, the TAS of the vehicle changes very slowly and therefore the second term can be neglected. The rate of climb can be got by multiplying the sin g with V. Therefore we get

$$\frac{\delta h}{\delta t} = V\sin\gamma = V\frac{(T \angle D)}{mg} \tag{2.7}$$

Here V in the expression is the True airspeed(TAS). By substituting the expression for drag from the equation 2.3 we get,

$$\frac{\delta h}{\delta t} = \frac{\left(V\left(T \angle \frac{1}{2}\rho V^2 S\left(C_{D0} + \frac{C_L^2}{\pi ARe}\right)\right)\right)}{mg} \tag{2.8}$$

Since thrust decreases with altitude, is it clear that *dh/dt* will also decrease with height also. The rate of climb decreases with height.This is because of the reduction of the thrust generating capability of the aircraft with height.The effect of V is difficult it to determine in this equation but there is an optimal climb velocity profile for which *dh/dt* is maximum.

2.2.5 Cruise Performance

In cruise the aircraft is in steady state where the lift balances weight and the thrust balances drag. The state of the vehicle is shown below.

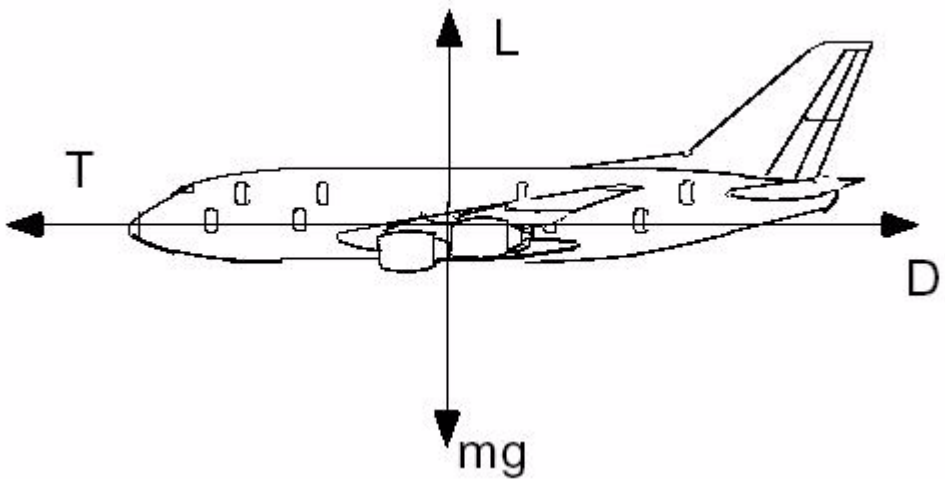


Figure 2.3 Aircraft in Cruise Mode (16).

The expressions for lift,drag and the drag-polar remain the same as in the climb phase.An important part of the cruise analysis is the estimation of the maximum range of the aircraft. Range is the maximum that the vehicle can travel without the need for refueling.The range represents a trade-off between the fuel and payload or cargo that the aircraft can carry.

The differential distance that the aircraft travels over a small interval of time dt is Vdt, where V is the TAS (true airspeed).

$$dR = Vdt \tag{2.9}$$

Since the aircraft only loses weight due to fuel consumption we can define the rate of change of the weight over time as the product of the thrust specific fuel consumption (TSFC) and the tractive effort required to move the vehicle at speed V. The specific range of the aircraft is defined as the ratio of distance flown to the fuel consumed and is a measure of fuel efficiency of the aircraft.

Therefore specific range of the aircraft can be written as

$$\frac{\delta R}{\delta W} = \frac{V}{TSFC(T)} \tag{2.10}$$

To estimate the range we multiply the right hand side by L/W which is acceptable since L=W and replace T by D since T=D

$$\frac{\delta R}{\delta W} = \frac{V}{TSFC(T)}\frac{L}{W} \tag{2.11}$$

Separating the variables and integrating we get

$$R = \frac{V}{TSFC}\left(\frac{L}{D}\right)\ln\left(\frac{W_i}{W_f}\right) \tag{2.12}$$

The above equation is called the Bruguet range equation. It should be noted that the above equation is only approximate since the ratio L/D is not constant throughout the flight since the aircraft consumes fuel through the flight and gets lighter and therefore the amount of lift needed to keep it in equilibrium also keeps reducing. Nevertheless the above equation is a good first order approximation for estimating the aircraft range.For a more accurate estimation of the range of very long range aircraft, the cruise phase can be divided into a number of segments and in each segment the L/D can be replaced by the average L/D in that particular segment.

2.2.6 Descent Performance.

The descent phase is the final phase in the flight. The descent profiles are rather difficult to calculate since the aircraft descends depending on the instruction provided by the Air Traffic control and is airport specific. Nevertheless the following observations have made about descent profiles

1) Usually transport aircraft descend at the rate of 900 m/min(3000 feet/min) during the initial phases of descent

2) Below 3000m aircraft enter a dense terminal area and are usually required to manouver around other air vehicles to establish coordinated flows to runways.

3) In the United States it is required to reduce the IAS(indicated airspeed) to 250 knots or less at heights less than 3000 m to avoid accidents in the event of a bird strike

4) Below 3000 m,the descent rate decrease to 500m/min or less and the descent profiles follows a series of steps at predesignated altitudes

5) Aircraft manufacturers report typical distance vs fuel consumption and distance and altitude curves similar to the one shown below.

It should again be emphasized that descent fuel consumption are airport-ATC specific and not completely dependent on aerodynamics unlike the climb and cruise phases.

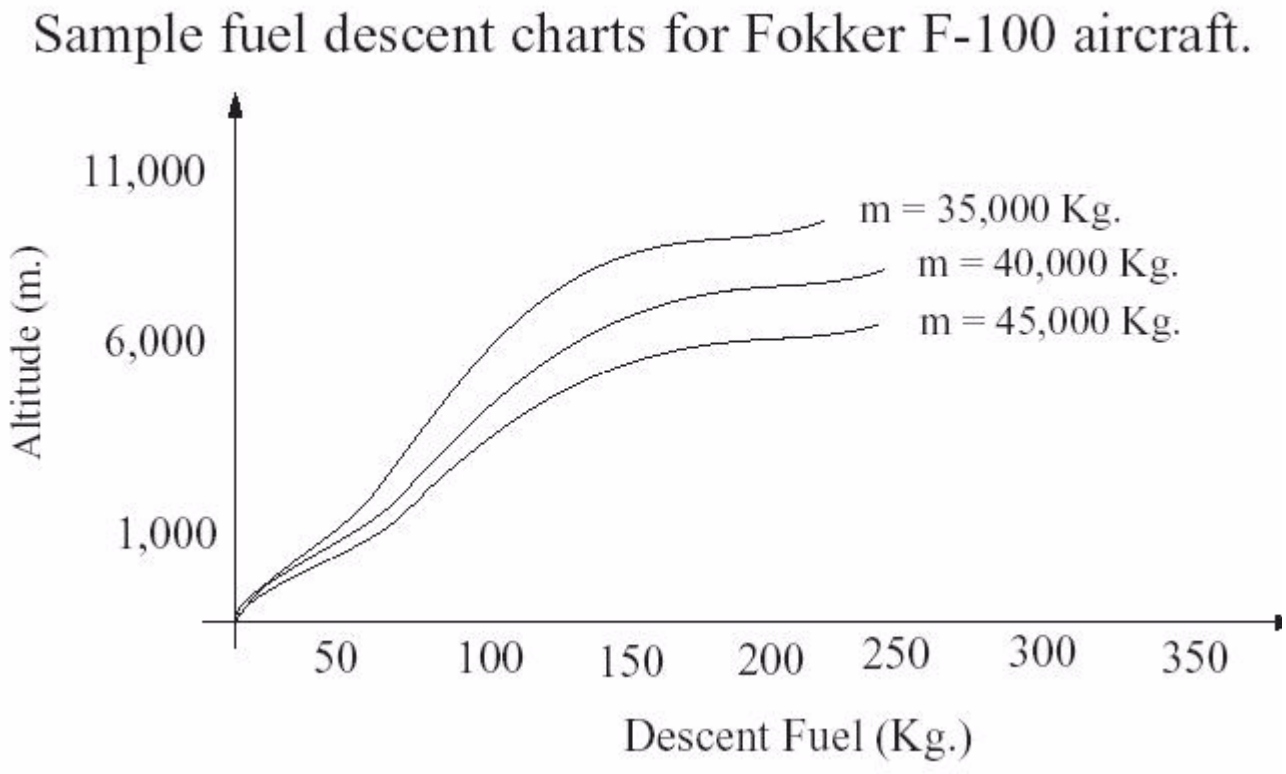


Figure 2.4 Descent Fuel Consumption Curves (16).

2.3 Description of the BADA Model

The BADA model is a analytical aerodynamic model developed by Eurocontrol to determine the performance of aircraft during the climb, cruise and descent phases. This model was used to implement the flight trajectory model to determine travel time for GA air passengers. The BADA model is on the same lines as the aircraft performance equations described in Chapter 2 but is more accurate, since fewer simplifying assumptions are made. Nevertheless the BADA model has some shortcomings which are highlighted by comparing it with another model developed at Virginia Tech by Dr Antonio Trani and Mr Frank Cheung. The BADA model is also used in the ASAC mission generator developed by LMI which addresses the impact of advanced aviation technologies on the mobility of the air transportation network.

The typical flight consists of the following phases

(i) Departure - at the gate waiting to taxi

(ii) Taxi out - to the takeoff runway

(iii) Takeoff - From the point the aircraft starts rolling on the runway to a point where it clears an imaginary obstracle 11m(35 ft) above the ground.

(iv) Climb - From 35 ft to cruise altitude

(v) Cruise - Steady state flight from the top of climb (TOC) to the top of descent (TOD)

(vi) Descent/approach - From the top of descent to the point of touchdown

(vii) Taxi in - approach the landing gate from the runway.

Out of these phases detailed analyses are performed only for the climb,cruise and descent phases, time for taxi-in, taxi-out and takeoff phases can be extracted from FAA CODAS database and aircraft manufacturers.

2.3.1 Climb and Cruise Operations Phase

The basic equation describing the climb phase in the BADA model is the same as equation (2.4) and (2.5) . For the cruise mode, force equilibrium along and normal to the flight path gives

$$T = D \tag{2.13}$$

and

$$L = mg \tag{2.14}$$

The BADA model uses a total energy method to estimate the rate of climb, in the BADA model the dV/dt term in the rate of climb expression is not assumed to be zero as was stated in the previous section.

The work done by the forces on the aircraft is equal to change in kinetic and potential energy

$$(T \angle D)V_{TAS} = mg\frac{dh}{dt} + mV_{TAS}\frac{dV_{TAS}}{dt} \tag{2.15}$$

This can be written as:

$$(T \angle D)V_{TAS} = mg\frac{dh}{dt} + mV_{TAS}\frac{dV_{TAS}}{dh}\frac{dh}{dt} \tag{2.16}$$

Rearranging the terms

$$\frac{dh}{dt} = \frac{(T \angle D)V_{TAS}}{mg}\left(1 + \frac{V_{TAS}}{g}\frac{dV_{TAS}}{dh}\right)^{\angle 1} \tag{2.17}$$

The second term can be expressed as a an “energy share factor”, since it gives us how much energy is available for climbing as opposed to the amount available for acceleration.

However the expressions for fuel consumption, thrust and drag are different in the BADA model leading to different expressions for rate of climb and maximum range.

The drag polar of the BADA model is of the form

$$C_D = (C_{D0} + C_{D2} \times C_L^2)(1 + C_{D16} \times M^{16}) \tag{2.18}$$

where C_{D0} is the zero-lift drag coefficient and C_{D2} is the drag-coefficient due to induced lift and C_{D16} is the transonic correction factor. The the third term is to account for the fact that the zero-lift and induced drag coefficient themselves vary with the true mach number, particularly for high subsonic velocities. The non-inclusion of this factor in the drag computation results is significant inaccuracies in fuel consumption, as will be shown later.

$$C_L = \frac{2L}{\rho V^2 A} \tag{2.19}$$

Substituting equation (2.19) in (2.18)

$$D = L\left(C_{D0}\frac{\rho M^2 a^2}{2L} + C_{D2}\frac{2L}{\rho M^2 a^2}\right) \tag{2.20}$$

where D is the drag, L is the lift , a is the velocity of sound, M is the true mach number and r is the density of air. Therefore the L/D ratio which was assumed to be a constant in the earlier analysis can be got from the equation (3.7). The BADA models also give expressions for fuel consumption in terms of thrust and true air speed(TAS).

For turbojets the fuel consumption is

$$F = Tf_1\left(1 + \frac{Ma}{f_2}\right) \tag{2.21}$$

where f_1 and f_2 are aircraft specific constants and fuel consumption is in kg/hr. For turboprops the expression is

$$F = TMaf_1\left(1 \angle \frac{Ma}{f_{T2}}\right) \tag{2.22}$$

The BADA model also expression for idle and maximum thrust of the aircraft. The idle thrust is used to compute fuel consumption during the taxi in and taxi out phase and the maximum thrust is used during the climb phase.

The idle thrust is given by

$$T_{idle} = C_{T1}\left(1 \angle \frac{\dot{h}}{C_{T2}} + C_{T3}M\right) \tag{2.23}$$

where T_{idle} is the idle thrust in Newtons, h is the height of the aircraft in metres and C_{T1},C_{T2} and C_{T3} are aircraft specific constants. The idle fuel flow is given by

$$F_{idle} = f_i\left(1 \angle \frac{h}{f_{i2}}\right) \tag{2.24}$$

where F_{idle} is the idle fuel consumption in kg/hr and h is the height of the aircraft in m and f_i and f_{i2} are aircraft-specific constants. The maximum thrust is given by

$$T_{max} = T_{M1}\left(1 \angle \frac{h}{T_{M2}} + \frac{T_{M3}}{h^2}\right) \tag{2.25}$$

for turbojets, where T_{max} is in Newtons and h is the aircraft height in metres and T_{M1},T_{M2} and T_{M3} are aircraft-specific constants and

$$T_{max} = T_{M1}\left(1 \angle \frac{h}{T_{M2}}\right) + T_{M3} \tag{2.26}$$

for turboprops and

$$T_{max} = T_{M1}\left(1 \angle \frac{h}{T_{M2}}\right) + \frac{T_{M3}}{V_{TAS}} \tag{2.27}$$

for piston engines.The BADA models enables one to derive an exact expression for fuel consumed for traveling a particular distance, similar to the derivation of the Bruguet range equation. However since the expression for thrust and drag are different in the BADA model, the expression for the range is also different from the Bruguet range equation

For flight at constant speed and altitude, the rate of change of weight of the aircraft with respect to time is given by

$$\frac{dW}{dt} = \angle gF \tag{2.28}$$

where g is the acceleration due to gravity and F is the fuel consumption is kg/hr and dW/dt is in Newton/hr, therefore the rate of change of weight with respect to distance is given by

$$\frac{dW}{dx} = \frac{\angle gF}{V_{kt}} \tag{2.29}$$

where V_{kt} is the true airspeed (TAS) in knots and the negative sign indicates that the weight decreases with distance. Substituting equations (2.13), (2.14), (2.17) and (2.18) in (2.25) we get an equation of the form

$$\frac{dW}{dx} = \angle a \angle bW^2 \tag{2.30}$$

solving this differential equation by separation of variables, we get

$$W = \sqrt{\frac{a}{b}}\tan\left(\text{atan}\left(\sqrt{\frac{b}{a}}W_0\right) \angle (x \angle x_0)\sqrt{ab}\right) \tag{2.31}$$

which is different from the Bruguet range equation, where the constants *a* and *b* can be written as

$$\sqrt{\frac{a}{b}} = \frac{\rho V^2}{2} A \sqrt{\frac{C_{D0}}{C_{D2}}} \tag{2.32}$$

and

$$\sqrt{ab} = \frac{g f_1}{V_{kt}} (1 + C_{D16} \times M^{16}) \left(1 + \frac{v}{f_2}\right) \sqrt{C_{D0} C_{D2}} \tag{2.33}$$

The BADA model has several files, two of which of which have been used for the above analysis. These are the Performance table files (PTF) and operations performance files (OPF). These files are provided for each aircraft type. The PTF files contain a summary table of speeds, climb/descent rates and fuel consumption at various flight levels. The OPF files specify parameter values for the mass, flight envelope, drag, engine thrust and fuel consumption.

2.4 Comparison of BADA and Virginia Tech Fuel Consumption Models

In this section the accuracy of the BADA fuel consumption model is compared with a neural network model developed at Virginia Tech by Dr Antonio Trani and Frank Cheung. As explained the standard parabolic drag polar in not accurate particularly at high mach numbers since in the parabolic drag polar the zero-lift drag coefficient (C_{D0}), is not constant but varies with the true mach number. This fact is reflected in the BADA model by inclusion of the transonic correction factor (C_{D16}). Unfortunately in the BADA model for most aircraft this term is zero introducing an error in the drag formulation. Since most of the aircraft fuel is spent in overcoming drag (in the cruise mode), the BADA model overestimates the range and underestimates the flying time for most aircraft. The two models are compared in this section.

The aircraft used for this purpose was the Fokker F-100 which a short-range turbofan powered aircraft. The fuel consumptions for this aircraft at different phases of the flight (climb and cruise) and different altitudes,weights and mach numbers were taken from the flight manual. The neural network was first

"trained" on this dataset using the Levenberg-Marquardt algorithm. The training of the neural network involves determining the weights and transfer functions of the network using the above mentioned algorithm. The next step is testing the neural network on a different dataset. For this purpose points were generated randomly between the test points and performance of the two models was tested by comparing their predicted values with tha actual flight manual data, the results of which are shown in Figure 2.5.From this graph it clear that the neural network model is much more accurate than the BADA data

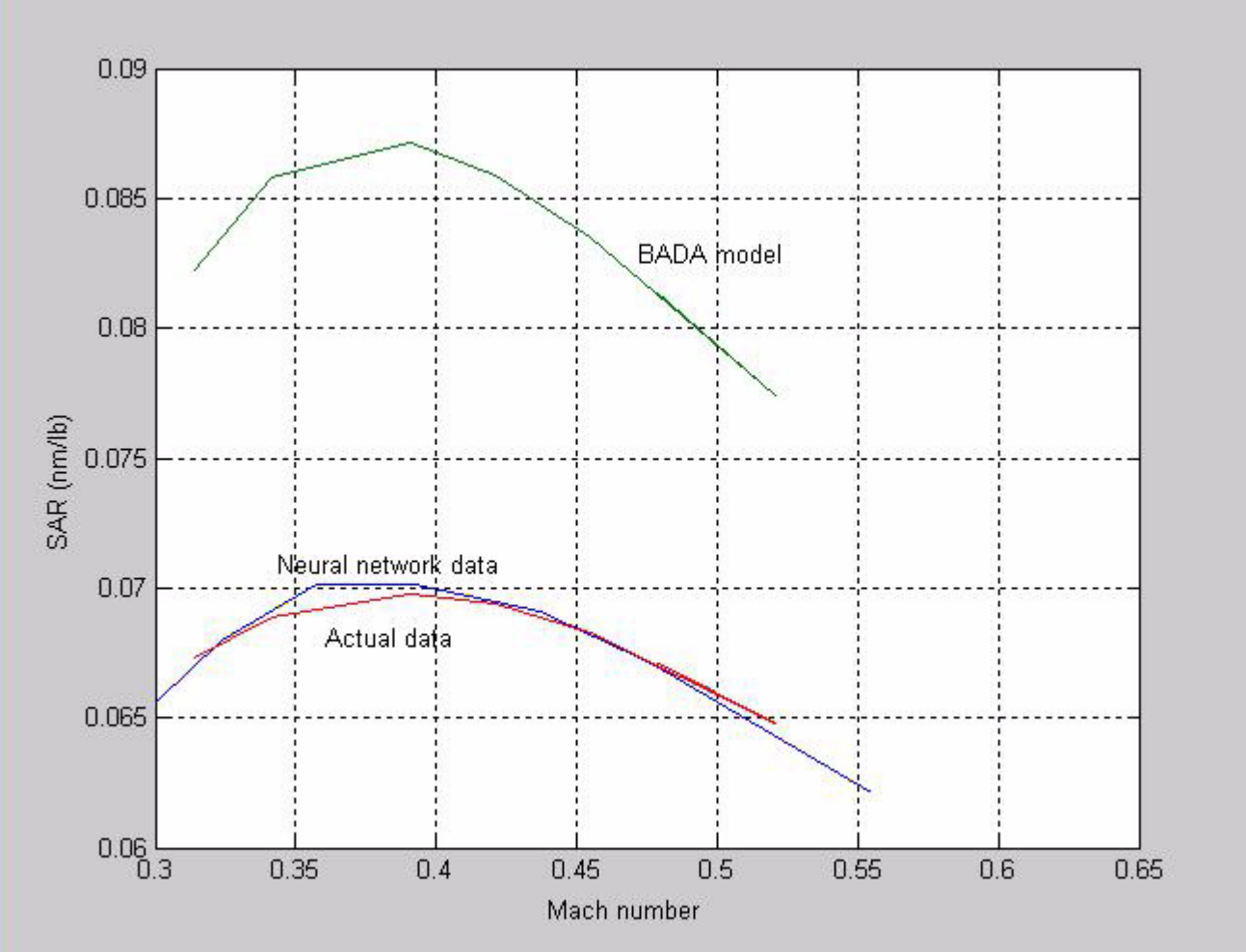


Figure 2.5 Comparison of the Neural Network and the BADA Model (16).

2.3 Transportation Systems Modeling Process

2.3.1 GA OD Table

The GA OD Table is an output of the SATS Transportation systems analysis. This OD table contains the number of GA person-trips by GA mode (Single engine,Multi-engine and Jet engine) between every pair of airports in the continental US. This is important for the moblility study because the number of operations will the affect the mobility of the Air Transportation network as an increase in the number of operations will invariability lead to an increase in the number of conflicts, thus increasing the travel time. Also the GA OD table would give the modeler of the demand function at each airport, since many airports would have no operations from them and this information would be useful in evaluating the mobility of the Air Transportation network. The GA OD table is obtained from the traditional four step transportation systems approach which consists of Trip generation,Trip distribution,Mode choice and Assignment (Senanu,Baik et al 2000) (17). These four steps are explained in detail below

2.3.1 Trip Generation.

The **trip generation** is used to predict the number of trips by trip purpose produced by each zone of activity and attracted to each zone.The output of this procedure is a simple Origin-Destination matrix with two vectors: one for productions and one for attractions. Trip generation models relate the socio-economic variables of the zone under consideration to the number of trips produced by that zone. Examples of relevant socio-economic variables are employment,industrial capital, population and gross regional and national product. The objective of the trip generation process is to predict the number of trips between the origin zone (O_i) and destination zone (D_j). Common methods of executing Trip generation models are regression,trip rate analysis and cross classification analysis. The process is shown in Figure 2.6

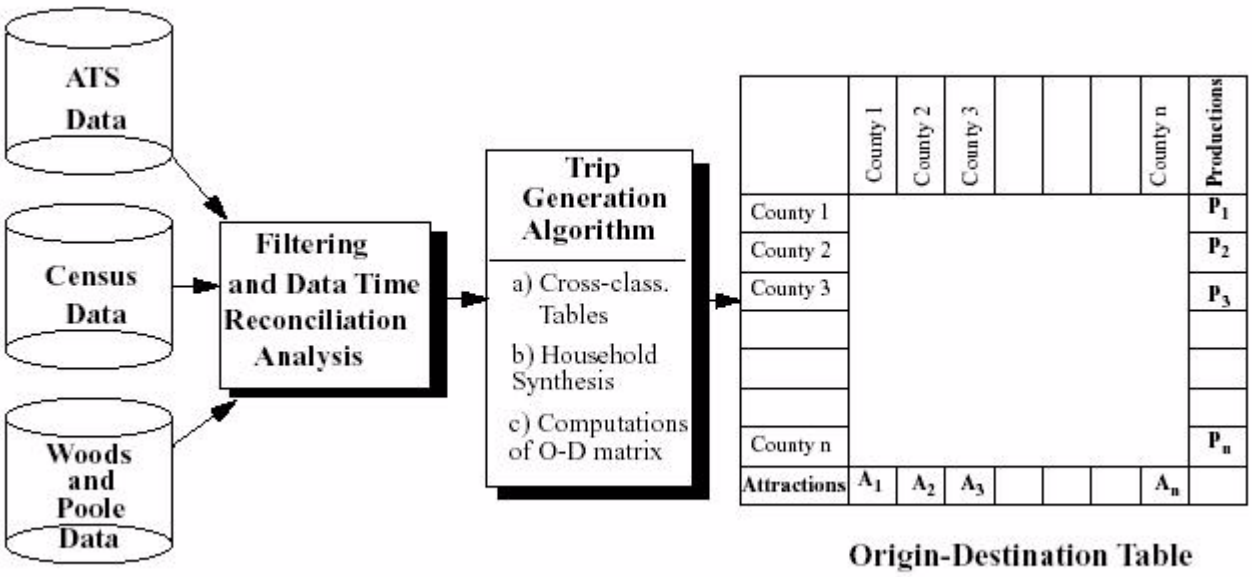


Figure 2.6 Trip Generation Process Flowchart (17).

2.3.2 Trip Distribution

In the **trip distribution** is used to predict origin-destination (O-D) flows, that is, trip ends predicted by the trip generation model are linked to form trip interchanges between zones. This results in a large tripinterchange matrix (or sometimes called an origin-destination, O-D, table) showing the number of trips between an origin to a destination county. The purpose of this step is to derive a realistic Origin-Destination matrix (O-D) to achieve credible travel patterns between centers of transportation activity, counties in this case. The rationale of trip distribution is as follows: all trip-attracting zones,*j* , in the region of interest are competing with each other to attract trips produced by each zone *i.*Everything else being equal, more trips will be attracted by zones that have a higher level of attractiveness. The attractiveness is expressed as a function of salient socio-economic factors of the zone and the relative proximity of the attraction zone to others. The most popular model for the trip distribution process is

the Gravity Model which is based on the Newton's law of gravitation. This model states that the force of the attraction between two zones is directly proportional to the product of the masses of the two zones and inversely proportional to the square of the distance between them.

2.3.3 Mode Choice

Mode Choice predicts the percentage of person-trips selecting each mode of transportation while traveling ing between two zones in the region of interest. The general aviation mode competes with automobile,commercial airline, bus, train, etc.In the mode choice model the trip interchange matrix obtained in the trip distribution step is decomposed into a number of trip interchange matrices consistent with the number of modes studied. In the above analysis there are are five competing modes: 1) ground modes (called others), 2) general aviation single-engine aircraft, 3) general aviation multi-engine, 4) general aviation jet, and 5) commercial aircraft. In general, modal split analysis uses the outputs of the gravity formulation and assigns trips to each mode considering the perceived utility of each mode individually. The various methods of modeling mode choice are Nested Multinomial Logit models, Discrete choice models and Diversion curves.

2.3.4 Traffic Assignment (Airport Choice).

Trip Assignment places the O-D flows for each mode on specific routes of travel through the respective networks. In this step the airport-airspace network interactions to assess the impact of SATS operations in NAS are studied in detail. The Airport choice model is used to convert the trips by mode to person trips between 3346 paved airports in the Continental United States.The airport choice model output gives general aviation aircraft operations by single, multi and jet engine aircraft through public use airports (included in the database) in the continental US. The Airport choice is discussed in detail later in this chapter.

2.4 Ground Network Analysis

As mentioned earlier the air travel network consists of both air and ground components and the ground network travel times can sometimes account for half of the total travel time, a proper technique of calculating the ground network mobility for both the base and horizon year thus becomes necessary.

2.4.1 Network Connectivity

According to classical network theory network connectivity is defined as the number of links in that network divided by the maximum possible number of links in the network, with the graph remaining planar. A graph is said to be planar if it can be drawn in a plane in such a way such that no two links intersect except in a common node. The maximum possible number of links in a planar graph is given by 3(m-2) where m is the number of nodes. The degree of network connectivity is determined by the g index of the network which is the ratio of the number of nodes in the network and the maximum possible number of nodes in the network.

For example consider the two networks shown in Figure 2.7.

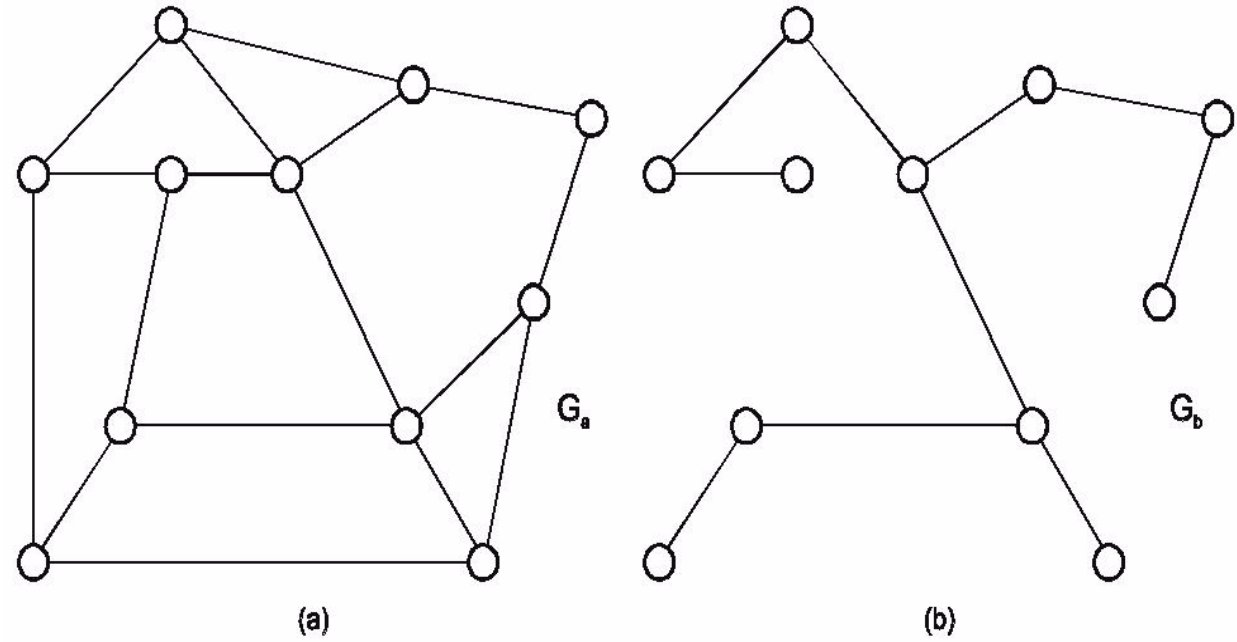


Figure 2.7 Connectivity of Two Networks.

In the first network the g index is 0.63 and in the second it is 0.37, and so the first network is more connected than the second. It therefore follows that the first network would have greater mobility and lesser travel time. Therefore determining the network connectivity is necessary for the evaluation of

any mobility metrics. However the network connectivity as defined above is very difficult to determine practically. Therefore the road density(miles of road/sq mile) was taken as the equivalent network connectivity, since greater road density implies greater network connectivity.

2.4.2 Network Level of Service

Network level of service or LOS is another variable having a major impact on the mobility of the network. Level of service is defined as the ability of the network to carry given amount of traffic. It depends on various factors such as traffic speed, travel time, volume and density of traffic, obstructions, lane width. Depending on these conditions LOS on the ground transportation network can vary from ideal to forced (A-F). These levels are given in detail in the table 2.1 (Garber and Hoel) (18).

Table 2.1. Level of Service Benchmarks

LOS	conditions	speed(mph)	density(pc/mi/ln)
A	free flow, little or no delay	V>55	11
B	stable flow,speeds restricted somewhat by volumes	50<V<=55	18
C	stable flow,speeds restricted more closely	45<V<=50	26
D	approaching unstable flow, tolerable speeds	40<V<=45	35
E	unstable flow,low speeds	V<=40	45
F	forced flow,frequent stoppages	Variable	Variable

2.4.3 Travel Times on Networks and Link Performance Functions

Travel time is the the main indicator of the mobility of the ground transportation network. Travel times depend on many factors such as speed,density,flow and grade. These factors are related to one-another. For example it is known that speed and volume follow a parabolic behavior and speed and density follow a linear relationship. However there exist explicit relationships that connect travel time and volume on a link. These are called Link Performance function and all try to fit a non-linear function to the observed flow-travel time relationship. Examples of Link performance functions

1) Bureau of public Transportation (BPR) link performance function

$$t = t_o \times \left(1 + \left(\frac{v}{c}\right)^4\right) \tag{2.34}$$

where t = travel time for link volume v

t_o = travel time for “free flow” condition

c=capacity of the link

2) CATS Link performance function

$$t = \frac{t_o \times \left(2^{\frac{v}{c}} + 1\right)}{2} \tag{2.35}$$

3) Davidson Link performance function

$$t = t_o \times \left(1 + J\left(\frac{v}{c \angle v}\right)\right) \tag{2.36}$$

the above equation is not used when v/c>1.0. From the above equations it is clear that peak and non-peak travel times will be different and therefore need to be treated separately.

2.5 Airport Choice Model

The Airport choice model is the fourth step in the traditional four-step transportation systems approach. The Airport choice model's aim is to convert the county-to-county trips obtained from the Trip distribution to airport-to-airport trips by assigning the country trips to surrounding airports. The airport choice model uses a psuedo-gravity approach which assumes that travelers will take routes with least travel time and with greater volume of travel services (Transportation systems Baseline systems assesment study 2002, Trani,Baik et al). The ideal explanatory variables would be access and egress times to the airport and number of operations at the airport but since these are difficult to obtain for GA operations these were substituted with access and egress distances to the airport and number of aircraft based in the airport respectively.The model was calibrated using parameters a_1 and a_2 as exponents for the aircraft based attractiveness and route length attractiveness values respectively. During the calibration process the estimated values of aircraft based at towered airports are compared with reported values and the parameters adjusted to obtain the lowest root mean square error values. This calibration only accounts for towered airports because the Terminal Area Forecast (TAF) statistics are not, in general, very reliable for untowered airports.Upon determining the appropriate value for the calibrating parameters, the model is re-run to obtain the airport to airport person-trip table. The output is a 3346 x 3346 airport to airport person-trip table. This table is further split into person-trips by aircraft type (single, multi and jet engine). Factors considered in splitting the trips between the various aircraft types include the number of each aircraft typebased at the airport, average annual utilization rates for the different aircraft types, occupancy (number of seats) and the distance between the selected airport pair. This yields three 3346 x 3346 person-trip tables for each aircraft type. To obtain the number of aircraft trips the person-trips is divided by an average occupancy factor associated with that aircraft type. The final output is made up of three 3346 x 3346 trip tables. This will serve as an input for the network analysis stage of the transportation modeling process.

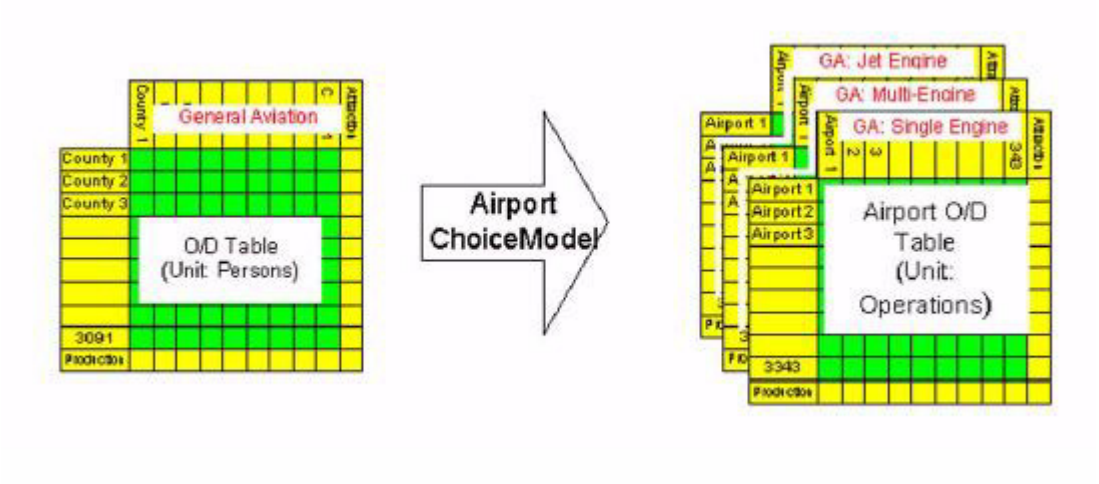


Figure 2.8 Schematic Representation of the Airport Choice Model.

Given an origin county *i* with *n* airports associated with it, a destination county *j* with *m* airports associated with it, and T_{ij} person-trips between the two counties, the methodology assumes that the number of persons that will make a trip from county *i* to *j* through a pair of selected airports will be correlated to two attractiveness factors that are dependent on the aircraft based at the airports and the total trip length (route attractiveness).

The derived Attractiveness is of the from,

$$Tas_{ijkl} = \{Airbness_{kl}\}^{\alpha_1} \times \{Routebness\}^{\alpha_2} \tag{2.37}$$

where a1 is the calibrating parameter for aircraft attractiveness and a2 is the calibrating parameter for route attractiveness.

The route attractiveness is,

$$Routebness_{ijkl} = Accessdist_{ik} + Egressdist_{jl} + Airportairportdist_{kl} \tag{2.38}$$

where $Accessdist_{ik}$ is the great circle distance from the weighted centroid of the origin county to the airport under consideration and $Egressdist_{jl}$ is the great circle distance from the weighted centroid of the destination county to the airport under consideration. $AirportAirportDist_{kl}$ is the great circle distance from the airport in the origin county to the airport in the destination county.

The aircraft-based attractiveness is the second factor that influences the number of trips from airport to airport. This factor is expressed as,

$$\left(Aircraftbaseattract_{kl} = \left\{ \frac{O_k}{\sum O_k} \right\} \times \left\{ \frac{D_l}{\sum D_l} \right\} \right) \tag{2.39}$$

where O_k is the aircraft based at airport k associated with the origin county i and D_l is the aircraft based in airport l in destination county j.

Given a set or routes between two county pairs the trip interchanges (t_{ijkl}) between airports can be estimated.

CHAPTER 3

Methodology and Data Sources

3.1 Introduction

As mentioned in the second chapter the mobility of the air travelers consists of two parts-the Airside part-which consists of the time spent in the air and time spent while connecting between two different flights at the airports and time spent in refueling at an intermediate airport when the distance between origin and destination is greater than the the maximum range capability of the aircraft. The landside part consists of the travel time between door to airport and airport-to-door at the origin and destination respectively. The mobility of the airside part of the model is measured using a flight trajectory which is based on the BADA 3.0 model for GA flights.For measuring the mobility of the airside part for Commercial flights the OAG and DB1B databases were used.This is described later in the chapter.

3.2 Data Sources Used in the Model

The landside part of the model required real travel time data, which was statistically significant over the entire United States and which also had corresponding socio-economic,demographic and infrastructure information which are perhaps the important factors affecting travel time. The data sources used were the Nationwide Personal Transportation survey (NPTS), American Travel Survey (ATS) and Highway performance Monitoring system (HPMS),Official Airline guide all of which are designed by the US Department of Transportation (USDOT). These databases are described in detail below:

3.2.1 American Travel Survey.

The American Travel Survey (ATS) (6) is perhaps the most valuable data sources for long-distance trip analysis. ATS contains detailed information not only on trip characteristics but also on traveler personal information. ATS defines long-distance travel as any round trip journey with a one-way travel distance longer than 100 miles. The main characteristics of the ATS survey are:

(i) Long-distance trips surveyed between January to December in 1995.

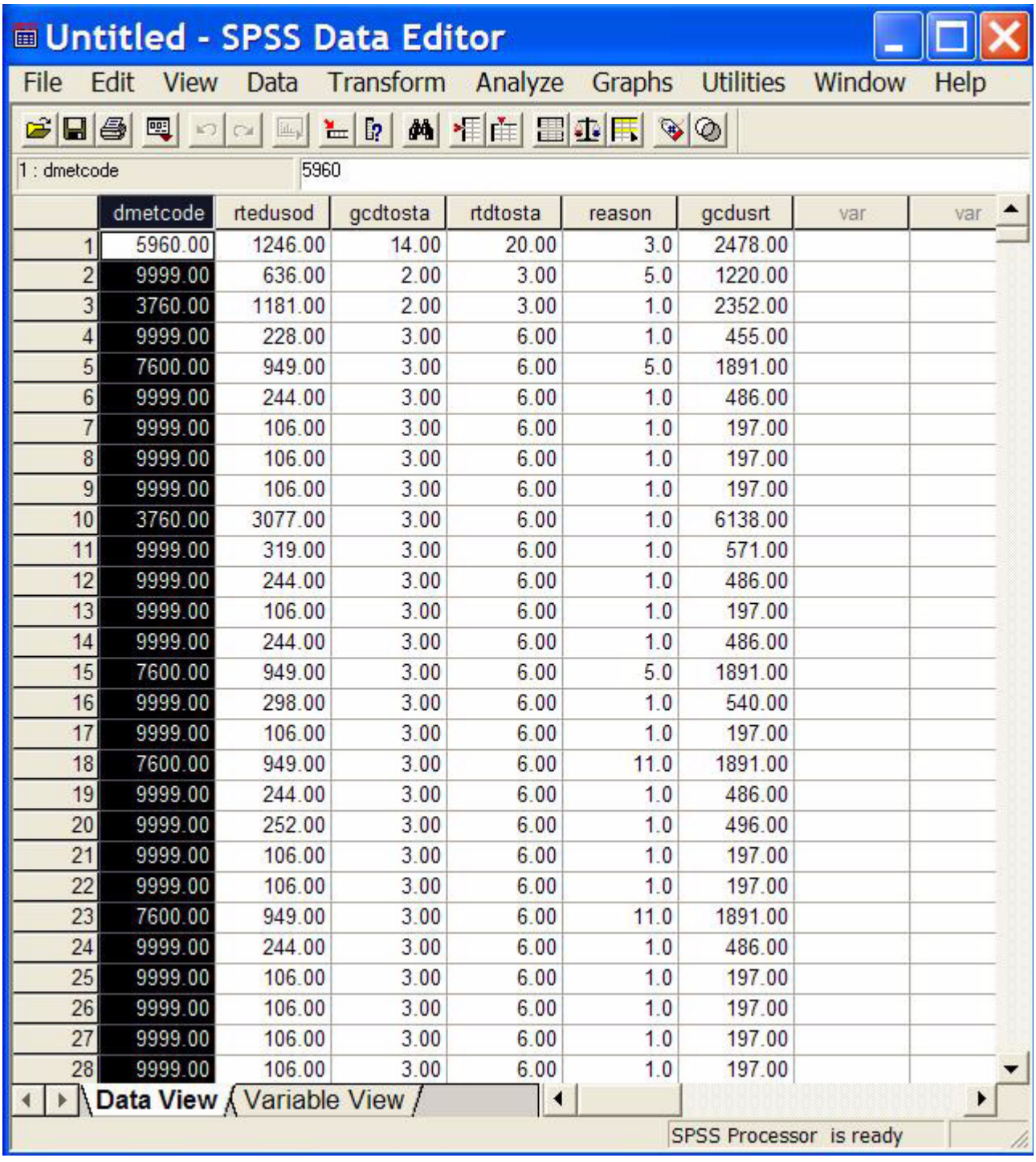
(ii) All-state data collection (including Hawaii and Alaska).

(iii) Information on 337,520 household trips and 556,026 person trips

(iv) More than 300 variables for each trip.

(v) Available in two formats: text format and SAS format. (SAS format is used in this analysis.)

Depending on the level of aggregation required, two types of data sets exist in ATS: 1) Household trip data, and 2) Person trip data. Both data sets are generated from the same data source, but summarized differently. In the person-trip data, all trips are recorded at the person level rather than household level. For example, if a household with three family members made a non-business trip by car, and all of the family members joined this trip, then the trip will create a single data record in the household trip data and there will be no description about how many people traveled together. On the other hand, three record lines for each family member are recorded in the person-trip data. A sample data set of ATS is shown graphically in Figure 2.2.The ATS survey divides each state into two types of regions: 1) Metropolitan Statistical Area (MSA) and 2) non-MSA areas. Virginia consists of three MSA area (Richmond, Norfolk, and D.C.) and one non-MSA area.The existing ATS data identified a total of 161 MSA areas and 50 non-MSA areas across the U.S. According to ATS, there were more than one billion person trips in 1995 (ATS, 1995). Of these 96% were domestic and the remaining 4% international. Considering domestic travel, more than 78% of the total trips were conducted for non-business (776 million trips) and 22%(of the total trips were made for business trips (225 million trips). The ATS was used to get the variation of trip distances to the airport as a part of the landside part of the model for determining travel time. The above mentioned analysis was performed by Mr Myunghyun Lee in Northern Virginia graduate school of Virginia tech.



	dmetcode	rtedusod	gcdtosta	rtdtosta	reason	gcdusrt	var	var
1	5960.00	1246.00	14.00	20.00	3.0	2478.00		
2	9999.00	636.00	2.00	3.00	5.0	1220.00		
3	3760.00	1181.00	2.00	3.00	1.0	2352.00		
4	9999.00	228.00	3.00	6.00	1.0	455.00		
5	7600.00	949.00	3.00	6.00	5.0	1891.00		
6	9999.00	244.00	3.00	6.00	1.0	486.00		
7	9999.00	106.00	3.00	6.00	1.0	197.00		
8	9999.00	106.00	3.00	6.00	1.0	197.00		
9	9999.00	106.00	3.00	6.00	1.0	197.00		
10	3760.00	3077.00	3.00	6.00	1.0	6138.00		
11	9999.00	319.00	3.00	6.00	1.0	571.00		
12	9999.00	244.00	3.00	6.00	1.0	486.00		
13	9999.00	106.00	3.00	6.00	1.0	197.00		
14	9999.00	244.00	3.00	6.00	1.0	486.00		
15	7600.00	949.00	3.00	6.00	5.0	1891.00		
16	9999.00	298.00	3.00	6.00	1.0	540.00		
17	9999.00	106.00	3.00	6.00	1.0	197.00		
18	7600.00	949.00	3.00	6.00	11.0	1891.00		
19	9999.00	244.00	3.00	6.00	1.0	486.00		
20	9999.00	252.00	3.00	6.00	1.0	496.00		
21	9999.00	106.00	3.00	6.00	1.0	197.00		
22	9999.00	106.00	3.00	6.00	1.0	197.00		
23	7600.00	949.00	3.00	6.00	11.0	1891.00		
24	9999.00	244.00	3.00	6.00	1.0	486.00		
25	9999.00	106.00	3.00	6.00	1.0	197.00		
26	9999.00	106.00	3.00	6.00	1.0	197.00		
27	9999.00	106.00	3.00	6.00	1.0	197.00		
28	9999.00	106.00	3.00	6.00	1.0	197.00		

Figure 3.1 Sample of the American Travel Survey (6).

3.2.2 Nationwide Personal Transportation Survey (NPTS).

The NPTS (5) is a household travel survey that provides data on the amount and nature of personal travel in the U.S. Data are collected from a sample of households and expansion factors are applied to obtain annual, national estimates of trips, miles of travel, household vehicles, etc. The survey collects data on all **personal** trips, by all modes, for all purposes. The NPTS data provide the only authoritative source for the characteristics of personal travel, particularly as linked to the demographics of the traveler, for the nation. These data allow analysis of trends in travel and the relative use of different modes of transportation. The NPTS was conducted through the telephone. The NPTS has data such as

(i) Household level - household size, number of household vehicles, income, location.

(ii) Person level - age, gender, education, relationship within the household, driver status, annual miles driven if a worker, worker status, if drive as an essential part of work if employed, seat belt use.

(iii) Customer service questions - rating of potential problems in traveling, such as mobility, congestion, safety, traffic conditions, and pavement conditions.

(iv) Vehicle level - annual miles driven (based on odometer readings recorded typically two months apart), make, model, model year.

(v) Trip level - trip purpose, mode, length (in miles and minutes), time of day, vehicle characteristics (if a household vehicle was used), number of occupants, driver characteristics (for private vehicle trips only and if a household member was driving).

The NPTS data are used primarily for gaining a better understanding of the elements of travel behavior, such as: (i) Amount and characteristics of travel at the individual and household level

(ii) Travel characteristics, such as trip chaining, use of the various modes, amount and purpose of travel by time of day and day of week, vehicle occupancy, and a host of other attributes

(iii) The relationship between demographics and travel, e.g., the 1995 NPTS shows increases in personal mobility among women, older Americans, youth, and to some degree, low-income households

(iv) The public's perceptions of the transportation system

As in the ATS the NPTS gives the Metropolitan Statistical Area (MSA) where the trip started. The NPTS was used to determine the distribution of travel speeds.

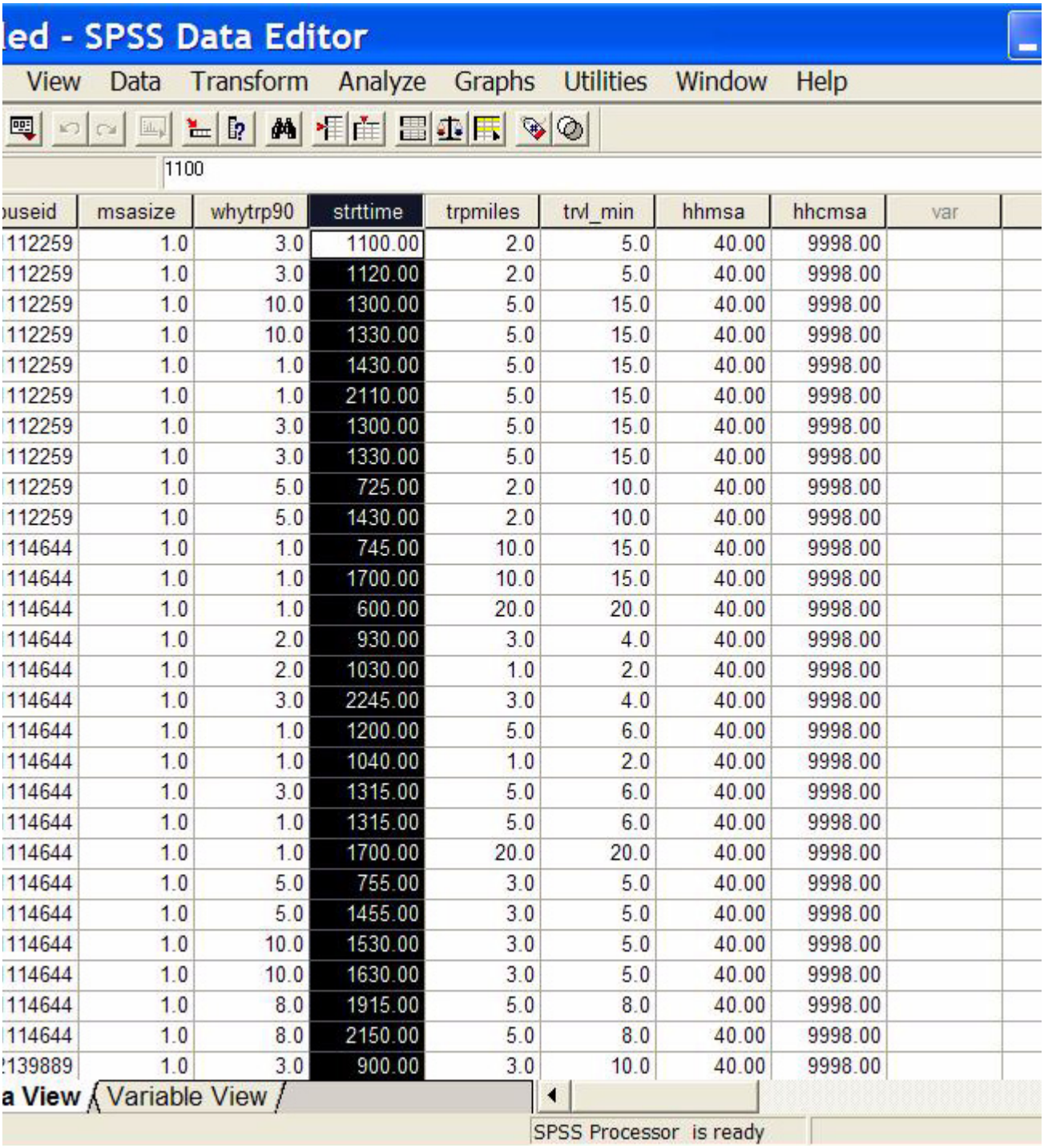


Figure 3.2 Sample of NPTS Data (5).

3.2.3 Census Data (14)

The Census is survey carried out every five years which gives the demographic and the socio-economic characteritics of the United States. It also gives information about trends in sub-groups within the general population. Examples of such information are:

(i) Persons of age 65 and older.

(ii) The American Indian and Native Alaskan population

(iii) The Asian and Black Population

(iv) Average income of the Hispanic population

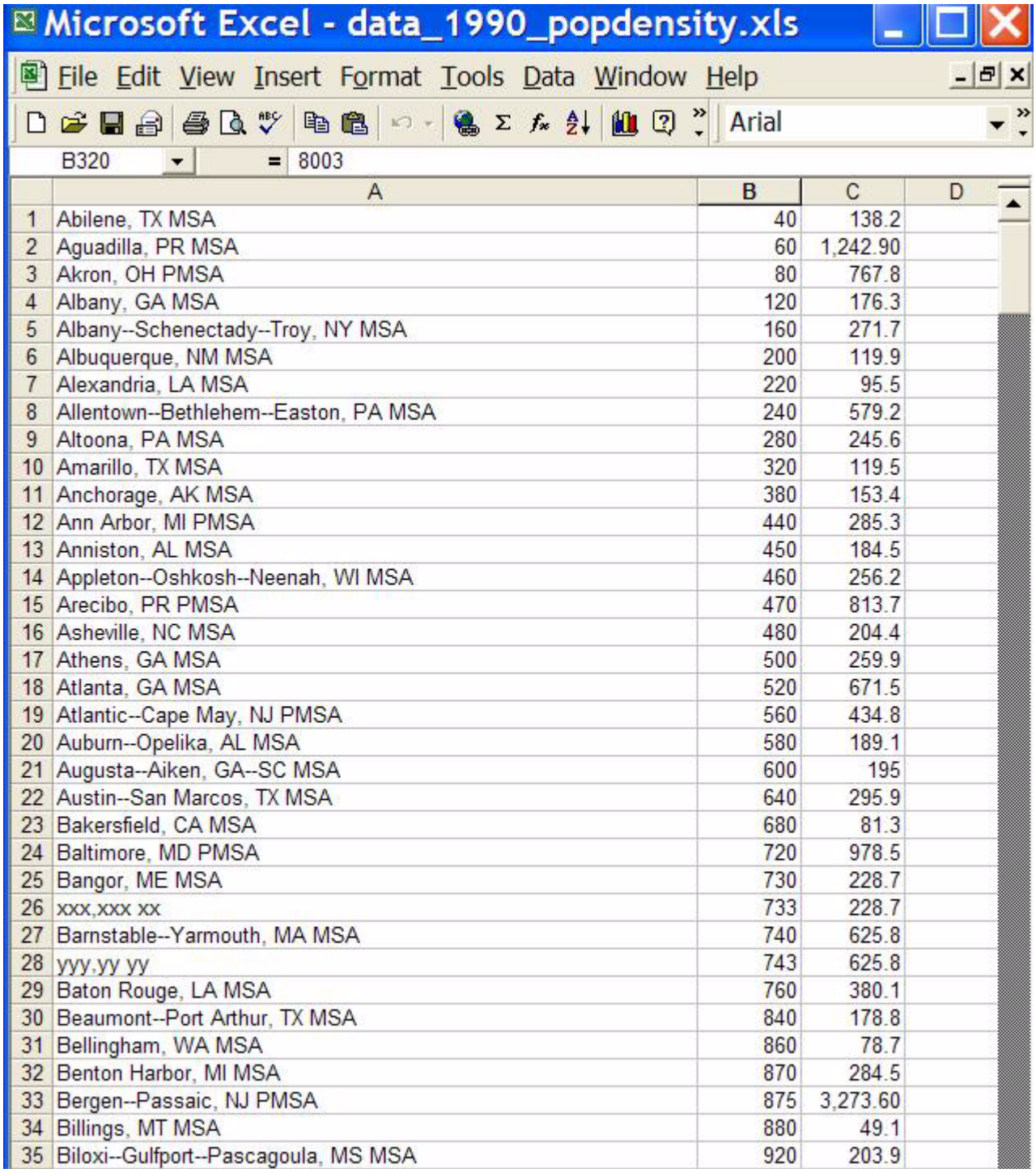
(v) Literacy rate among ethnic groups

In addition the Census Bureau produces two types of data- Microdata and Microdata.

1) Aggregate Data: There are four summary files in the aggregate type of data. Summary Files 1 and 2 contain 100-percent of the data (collected from all households) at various levels of geographical and racial/ethnic categories and include information on household relationship, sex, age, race, and housing tenure (owned or rented) and vacancy characteristics. Summary Files 3 and 4 contain sample data (collected from approximately 1-in-6 of total population and housing units who completed the long form) at varying levels of geographical and racial/ethnic categories.

2) Micro Data: The micro data, also known as the Public Use Microdata Sample (PUMS) files, includes the actual responses to the Census questionnaires at 1 and 5-percent levels of those who completed the long form. In other words, PUMS contains the complete structure of each household, including the number of people residing in a given household, the household income, number of workers, etc. Constructe to protect respondent confidentiality, these files allow users to perform customized data analyses on a wide range of population and housing characteristics.

The Census data was used to find out the population density of each of the 364 MSAs and PMSAs in the US, which is given the American factfinder in the Census data. The Census dataset was used to extract the MSA ID numbers. A snapshot of the Census data is given below.



	A	B	C	D
1	Abilene, TX MSA	40	138.2	
2	Aguadilla, PR MSA	60	1,242.90	
3	Akron, OH PMSA	80	767.8	
4	Albany, GA MSA	120	176.3	
5	Albany--Schenectady--Troy, NY MSA	160	271.7	
6	Albuquerque, NM MSA	200	119.9	
7	Alexandria, LA MSA	220	95.5	
8	Allentown--Bethlehem--Easton, PA MSA	240	579.2	
9	Altoona, PA MSA	280	245.6	
10	Amarillo, TX MSA	320	119.5	
11	Anchorage, AK MSA	380	153.4	
12	Ann Arbor, MI PMSA	440	285.3	
13	Anniston, AL MSA	450	184.5	
14	Appleton--Oshkosh--Neenah, WI MSA	460	256.2	
15	Arecibo, PR PMSA	470	813.7	
16	Asheville, NC MSA	480	204.4	
17	Athens, GA MSA	500	259.9	
18	Atlanta, GA MSA	520	671.5	
19	Atlantic--Cape May, NJ PMSA	560	434.8	
20	Auburn--Opelika, AL MSA	580	189.1	
21	Augusta--Aiken, GA--SC MSA	600	195	
22	Austin--San Marcos, TX MSA	640	295.9	
23	Bakersfield, CA MSA	680	81.3	
24	Baltimore, MD PMSA	720	978.5	
25	Bangor, ME MSA	730	228.7	
26	xxx,xxx xx	733	228.7	
27	Barnstable--Yarmouth, MA MSA	740	625.8	
28	yyy,yy yy	743	625.8	
29	Baton Rouge, LA MSA	760	380.1	
30	Beaumont--Port Arthur, TX MSA	840	178.8	
31	Bellingham, WA MSA	860	78.7	
32	Benton Harbor, MI MSA	870	284.5	
33	Bergen--Passaic, NJ PMSA	875	3,273.60	
34	Billings, MT MSA	880	49.1	
35	Biloxi--Gulfport--Pascagoula, MS MSA	920	203.9	

Figure 3.3 Sample of Census MSA Population Density Data (14).

3.2.4 Highway Performance Monitoring System (HPMS)

The HPMS (9) is a national level highway information system that includes data on the extent, condition, performance, use, and operating characteristics of the Nation's highways. In general, the HPMS contains administrative and extent of system information on all public roads, while information on other characteristics is represented in HPMS as a mix of universe and sample data for arterial and collector functional systems. Limited information on travel and paved miles is included in summary form for the lowest functional systems. The purpose of the HPMS is to assist in the analysis of the highway system condition,performance and investment that may be needed for improving any portion of the highway system. This information is included in a biennial report that the FHWA submits to the US Congress. The information in the HPMS database includes but not limited to

(i) AADT (Average Annual Daily Traffic)

(ii) Existence of High occupancy Vehicle (HOV) operations

(iii) Lane Miles of highway

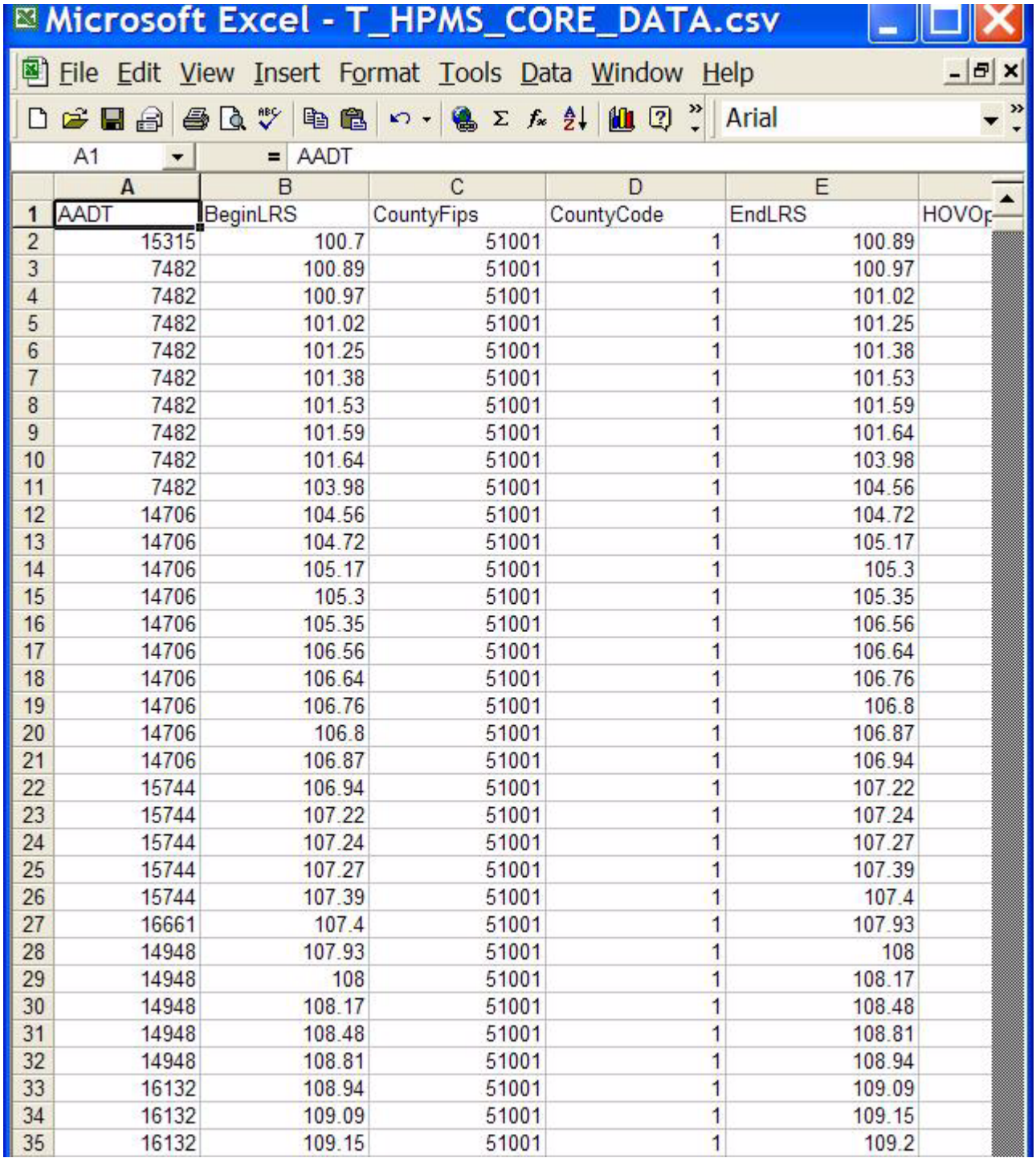
(iv) Miles of highway section

(v) County in which the highway section resides

(vi) Rural or Urban designation

(vii) Type of roadway section (One-way, Two-way, Bridge or Tunnel section).

The HPMS database was used to test the hypothesis that the population density and the road density(miles of road/square mile) are highly correlated. For testing the hypothesis the analysis was done on the county level. The miles of roadway section from the HPMS was divided by the area of the county from the Census data to get the road density for the county. This value was then plotted against the population density of the county to determine the validity of the hypothesis. A snapshot of the HPMS database is included below.



	A	B	C	D	E	
1	AADT	BeginLRS	CountyFips	CountyCode	EndLRS	HOVOp
2	15315	100.7	51001	1	100.89	
3	7482	100.89	51001	1	100.97	
4	7482	100.97	51001	1	101.02	
5	7482	101.02	51001	1	101.25	
6	7482	101.25	51001	1	101.38	
7	7482	101.38	51001	1	101.53	
8	7482	101.53	51001	1	101.59	
9	7482	101.59	51001	1	101.64	
10	7482	101.64	51001	1	103.98	
11	7482	103.98	51001	1	104.56	
12	14706	104.56	51001	1	104.72	
13	14706	104.72	51001	1	105.17	
14	14706	105.17	51001	1	105.3	
15	14706	105.3	51001	1	105.35	
16	14706	105.35	51001	1	106.56	
17	14706	106.56	51001	1	106.64	
18	14706	106.64	51001	1	106.76	
19	14706	106.76	51001	1	106.8	
20	14706	106.8	51001	1	106.87	
21	14706	106.87	51001	1	106.94	
22	15744	106.94	51001	1	107.22	
23	15744	107.22	51001	1	107.24	
24	15744	107.24	51001	1	107.27	
25	15744	107.27	51001	1	107.39	
26	15744	107.39	51001	1	107.4	
27	16661	107.4	51001	1	107.93	
28	14948	107.93	51001	1	108	
29	14948	108	51001	1	108.17	
30	14948	108.17	51001	1	108.48	
31	14948	108.48	51001	1	108.81	
32	14948	108.81	51001	1	108.94	
33	16132	108.94	51001	1	109.09	
34	16132	109.09	51001	1	109.15	
35	16132	109.15	51001	1	109.2	

Figure 3.4 Sample of HPMS Data (9).

3.2.5 Airline Origin and Destination Survey (DB1B).

Origin and Destination Survey (DB1B) (7), is a 10% sample of airline tickets from reporting carriers. Data includespassenger, freight, and mail transported. The DB1B gives the ticket specific information like

(i) Itinerary ID - Flight directional indicator

(ii) Market ID - Unique flight indicator

(iii) Fareclass- Indicates class of the ticket(first, business or economy class).

(iv) Origin Airport Code- Origin airport of the flight

(v) Destination Airport Code - Endpoint of the flight

(vi) Trip break code- Indicates whether the flight was direct or not.

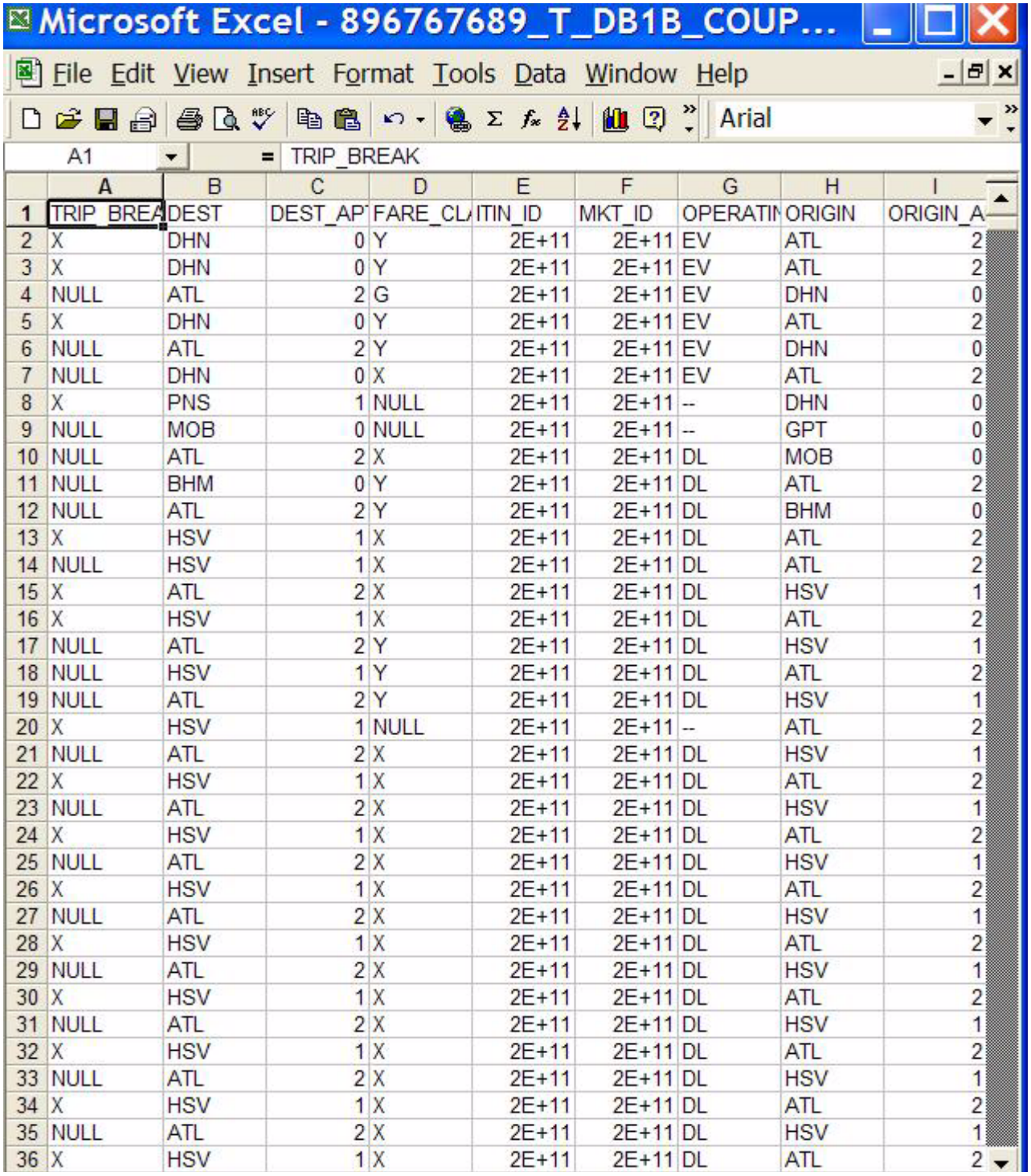
The DB1B has three types of databases- Market, Coupon and Ticket.

(i) DB1B Coupon-This table provides coupon-specific information for each domestic itinerary of the Origin and Destination Survey, such as the operating carrier, origin and destination airports, number of passengers, fare class, coupon type, trip break indicator, and distance.

(ii) DB1B Market- This table contains directional market characteristics of each domestic itinerary of the Origin and Destination Survey, such as the reporting carrier, origin and destination airport, prorated market fare, number of market coupons, market miles flown, and carrier change indicators.

(iii) DB1B Ticket-This table contains summary characteristics of each domestic itinerary on the Origin and Destination Survey, including the reporting carrier, itinerary fare, number of passengers, originating airport, roundtrip indicator, and miles flown.

The database used in the analysis was the DB1B Coupon. The main difference between Coupon and Market is that Coupon has the intermediate airport information, that is gives the leg information for the flight, which is necessary for determining the flight time, but the Market file has the origin and final destination but not the stopover airport. Also the DB1B has no information about international flights. The flights in DB1B are distinguished by Itinerary ID and Market ID. A flight between an origin (i) and destination (j) has the same Itinerary ID as the flight from j to i but the flights have different Market IDs. However a flight from an origin (i) to an intermediate airport (k) and a flight from k to i have the same Market IDs. A snapshot of the DB1B Coupon is included below.



	A	B	C	D	E	F	G	H	I
1	TRIP_BREA	DEST	DEST_AP	FARE_CL	ITIN_ID	MKT_ID	OPERATIN	ORIGIN	ORIGIN_A
2	X	DHN	0	Y	2E+11	2E+11	EV	ATL	2
3	X	DHN	0	Y	2E+11	2E+11	EV	ATL	2
4	NULL	ATL	2	G	2E+11	2E+11	EV	DHN	0
5	X	DHN	0	Y	2E+11	2E+11	EV	ATL	2
6	NULL	ATL	2	Y	2E+11	2E+11	EV	DHN	0
7	NULL	DHN	0	X	2E+11	2E+11	EV	ATL	2
8	X	PNS	1	NULL	2E+11	2E+11	--	DHN	0
9	NULL	MOB	0	NULL	2E+11	2E+11	--	GPT	0
10	NULL	ATL	2	X	2E+11	2E+11	DL	MOB	0
11	NULL	BHM	0	Y	2E+11	2E+11	DL	ATL	2
12	NULL	ATL	2	Y	2E+11	2E+11	DL	BHM	0
13	X	HSV	1	X	2E+11	2E+11	DL	ATL	2
14	NULL	HSV	1	X	2E+11	2E+11	DL	ATL	2
15	X	ATL	2	X	2E+11	2E+11	DL	HSV	1
16	X	HSV	1	X	2E+11	2E+11	DL	ATL	2
17	NULL	ATL	2	Y	2E+11	2E+11	DL	HSV	1
18	NULL	HSV	1	Y	2E+11	2E+11	DL	ATL	2
19	NULL	ATL	2	Y	2E+11	2E+11	DL	HSV	1
20	X	HSV	1	NULL	2E+11	2E+11	--	ATL	2
21	NULL	ATL	2	X	2E+11	2E+11	DL	HSV	1
22	X	HSV	1	X	2E+11	2E+11	DL	ATL	2
23	NULL	ATL	2	X	2E+11	2E+11	DL	HSV	1
24	X	HSV	1	X	2E+11	2E+11	DL	ATL	2
25	NULL	ATL	2	X	2E+11	2E+11	DL	HSV	1
26	X	HSV	1	X	2E+11	2E+11	DL	ATL	2
27	NULL	ATL	2	X	2E+11	2E+11	DL	HSV	1
28	X	HSV	1	X	2E+11	2E+11	DL	ATL	2
29	NULL	ATL	2	X	2E+11	2E+11	DL	HSV	1
30	X	HSV	1	X	2E+11	2E+11	DL	ATL	2
31	NULL	ATL	2	X	2E+11	2E+11	DL	HSV	1
32	X	HSV	1	X	2E+11	2E+11	DL	ATL	2
33	NULL	ATL	2	X	2E+11	2E+11	DL	HSV	1
34	X	HSV	1	X	2E+11	2E+11	DL	ATL	2
35	NULL	ATL	2	X	2E+11	2E+11	DL	HSV	1
36	X	HSV	1	X	2E+11	2E+11	DL	ATL	2

Figure 3.5 Sample of DB1B Coupon Data (7).

3.2.6 Official Airline Guide (OAG).

The OAG (8) contains information about flight schedules and frequencies for airlines throughout the world. The OAG includes information like

(i) Air Carrier- Indicates the airline that operates the service between the airport pairs

(ii) Origin Airport - The originating point of the flight

(iii) Destination Airport - The endpoint of the flight

(iv) Airtime - Time spent in air

(v) Ground time - Time spent on the ground including waiting and transfer times at airports

(vi) Seats - Number of available seats in the flight.

(vii) Distance - Flight distance in miles

(viii) Frequency - Number of flights between the origin and destination during the considered time intreval

(ix) Aircraft type - The name and model of aircraft that was flown between the origin and destination airports

The OAG was used along with the DB1B database to determine the the mobility for air travelers taking commercial flights. This is because only the OAG contains information about waiting and transfer time and the only DB1B contains the leg information. The combination of these two databases can also yield other important information about the structure of the airline network of the US such as location of hubs, percentage of direct flights and load factors for different markets.

All these data sources are used in conjunction with one another to give the final result which is the door-to-door travel time for air travelers. The model is explained in detail in the next chapter.

CHAPTER 4 Model Description

4.1 Introduction

In this chapter the door-to-door travel time model developed to determine the mobility of the air transportation network is described. The model has two parts-an airside part, which consists of the flight trajectory model, which will be described later in the section and a landside part which determines the access times and egress times for the baseline year (1995) and horizon year (2025). The total time is the sum of the two components. The total passenger time (in billions of passenger-hours) will give the modeler an idea of the present mobility and the reduction of mobility over the next 25 years, thus giving an indication of the importance of the new technogies NASA is proposing for achieving the goal of reducing intercity travel time by half by 2010 and by two-thirds over the next 25 years.

4.2 Variables Used in the Model

The variations in travel time both geographically and spatially can be explained by many factors like population, population density, road density, peak hour traffic,travel distance,local traffic congestions and whether the area under consideration is urban or rural.A few of these variables have been incorporated into the model. These are explained below.

4.2.1 Population

The driving time (driving speed) of an urban area was found to be dependent on the population of the urban area. This is because as the population increases, more number of people will be on the road, increasing the volume of the traffic and thus reducing the travel velocity and increasing the travel time The variation of travel speed with population in Figure 4.1.

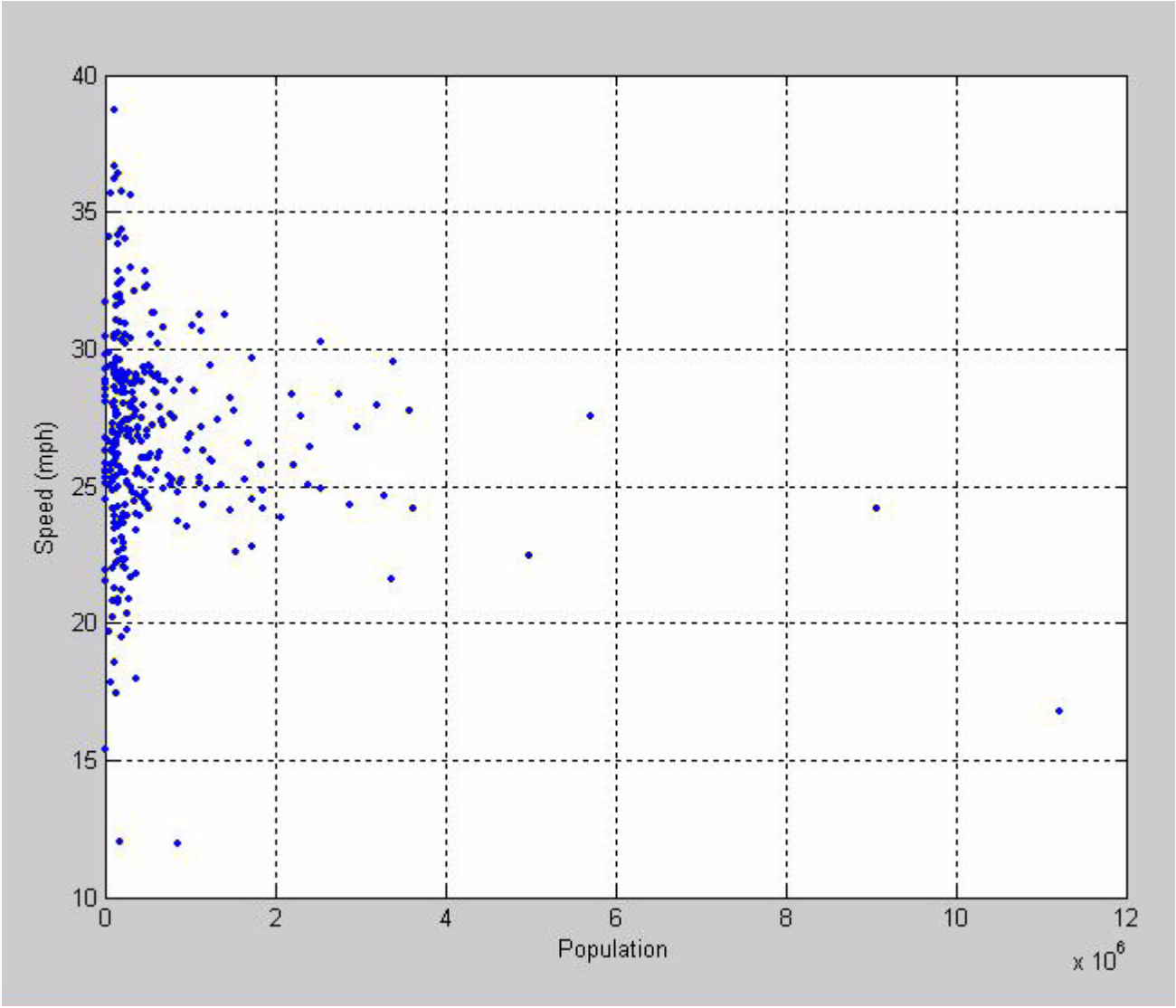


Figure 4.1 Relationship between Speed and MSA Population.

The correlation coefficient was found to be -0.139. The negative value is expected since the driving speed is reduced as MSA population increases.

4.2.2 Population Density

The driving speed was also dependent on the population density. It was postulated the speed would be more dependent on population density rather than population, since if two areas have some population and have different areas, the one with less area would have more congestion and less mobility.

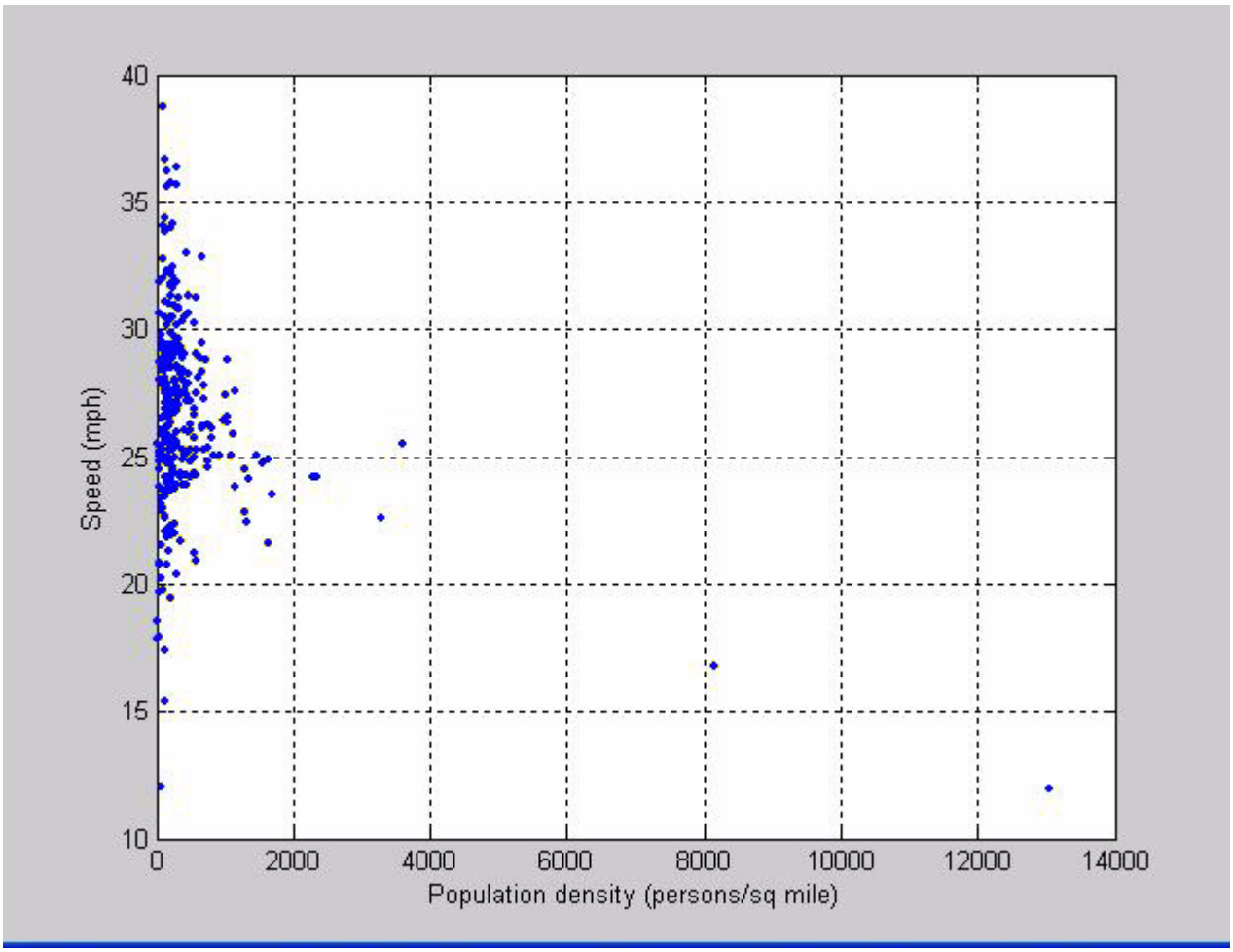


Figure 4.2 Variation of Travel Speed with Population Density.

The correlation coefficient was found to be -0.309 indicating that the relationship between speed and population density was stronger than the relationship between speed and population.

4.2.3 Travel Distance

It was found that travel speed is also dependent on travel distance,this is because longer trips will have greater average speed than very short ones. This variation is shown in Figure 4.3

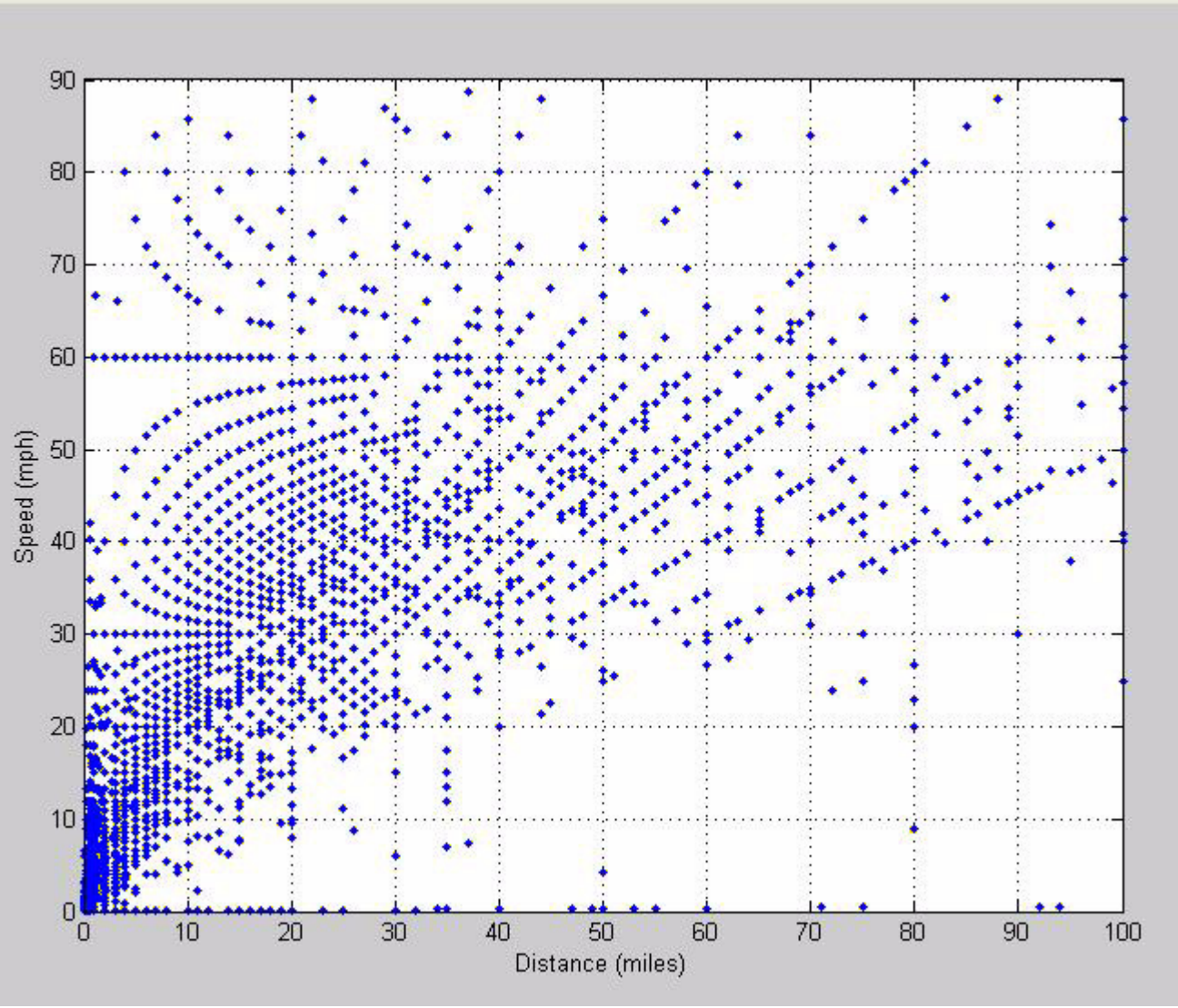


Figure 4.3 Variation of Travel Speed With Travel Distance.

Therefore an upper and a lower limit for the travel distance had to be found to make the model more accurate.

4.2.4 Urban/Rural Characteristic

The travel speed was found to be dependent on whether the area was urban or rural, since the urban areas would be more congested and have lesser travel speed.The variation is shown in Figure 4.4

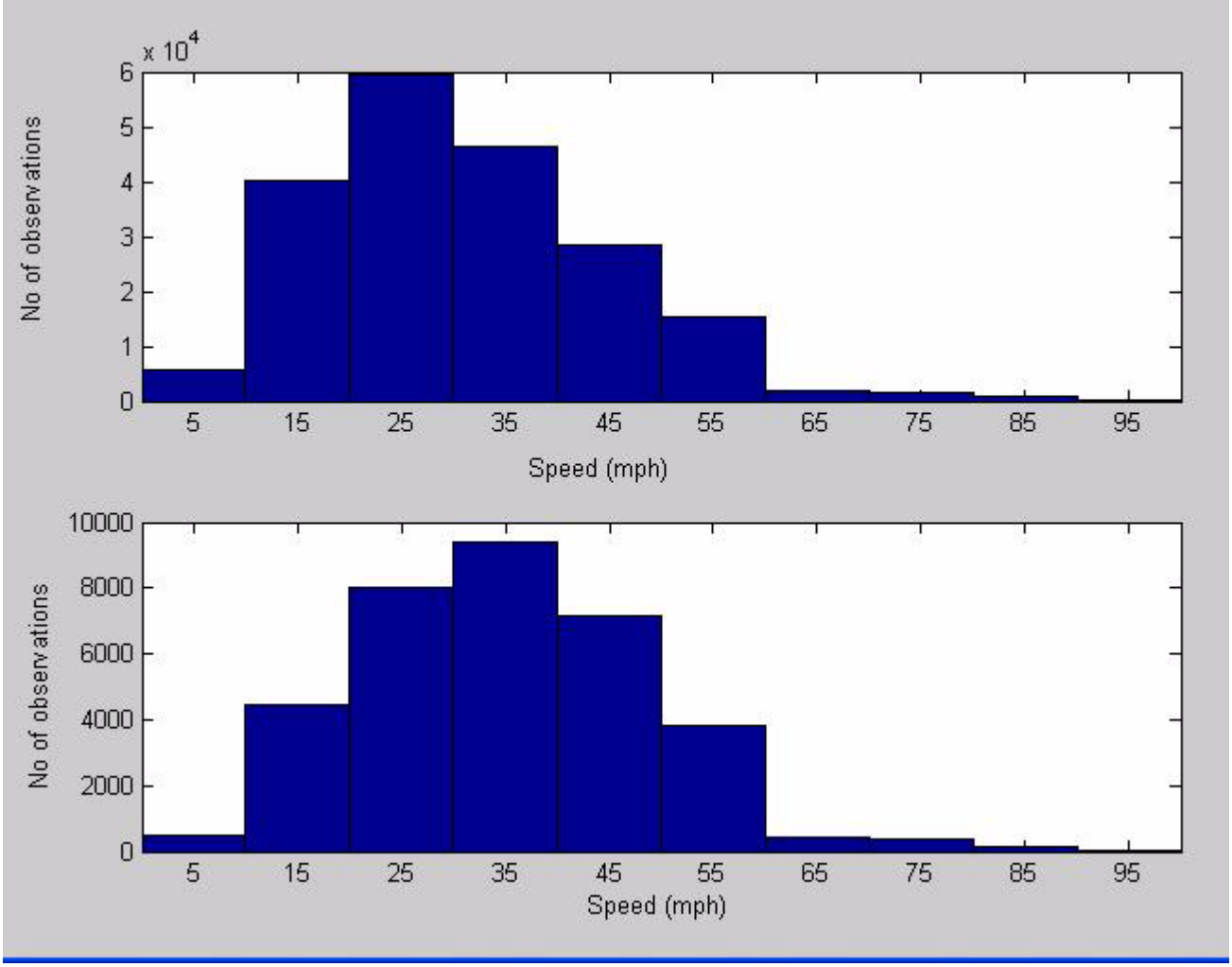


Figure 4.4 Histogram of MSA vs Non-MSA Speeds.

4.2.5 Peak vs Non-Peak Characteristics.

As mentioned in the second chapter, travel time increases with traffic volume and therefore peak and non-peak trips needed to be treated separately. The histogram of peak vs non-peak characteristics is shown in Figure 4.5.

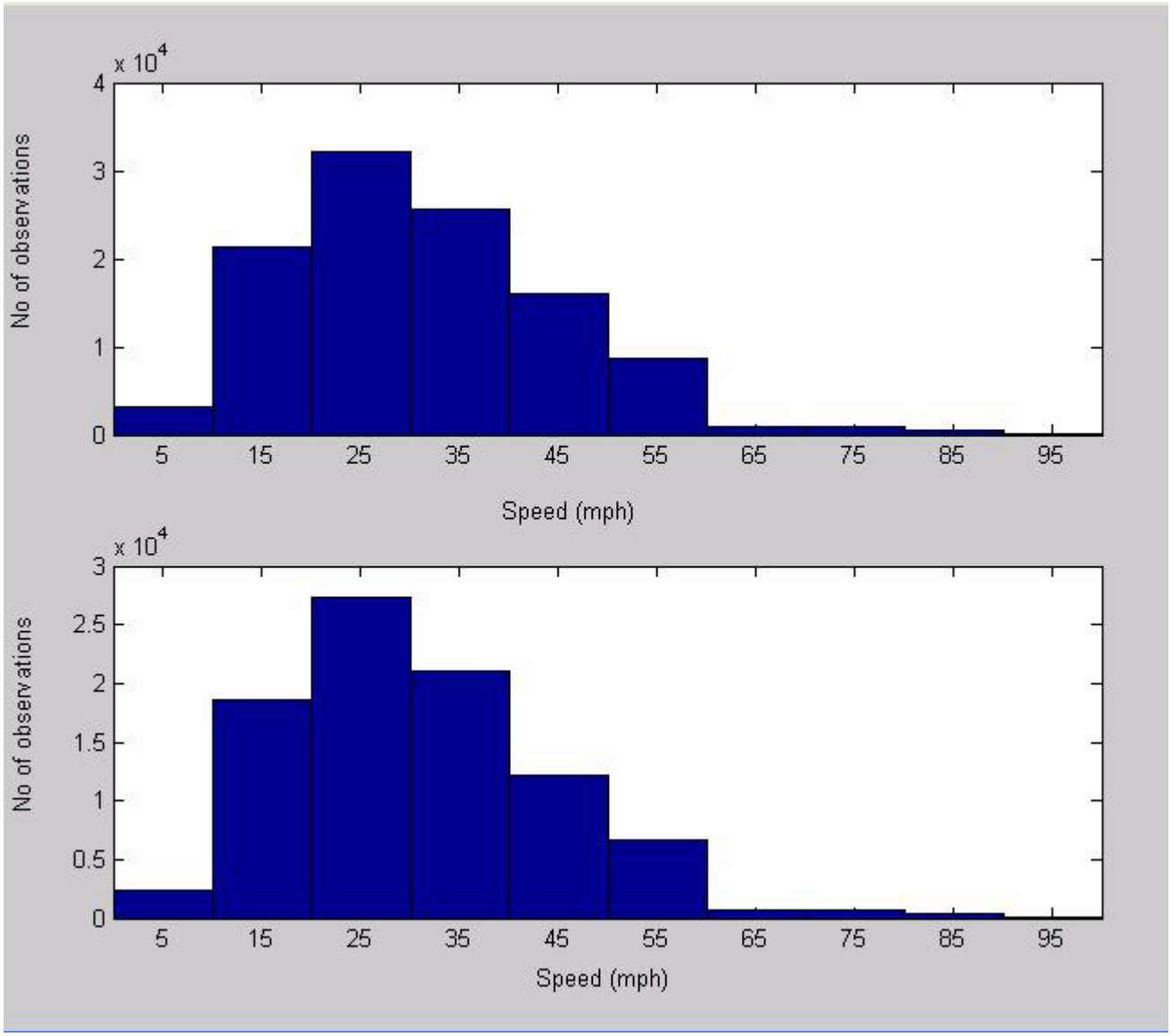


Figure 4.5 Peak and Non-Peak Characteristics For MSA Regions.

4.3 Model Assumptions

The model makes many assumptions, since all the issues could not be resolved due to lack of time or data. The assumptions in the airside part of the model are listed below.

In the flight trajectory model the selection of the intermediate airport, when the distance between the airports is greater than than the maximum range is based purely on distance, that is the airport nearest to the point where the aircraft runs out of fuel is chosen which is correct theoritically. But the above approach may not be correct practically since the nearest airport may be a very small airport which might not have landing or refueling facilities. Therefore the intermediate airport should be chosen based not only on the disatnce but also on the number of operations from the airport, since the number of operations from the airport is a direct indication of the infrastructure available at the airport. This correction was made in an updated version of the flight trajectory model written by Dr Hojong Baik.

The second assumption made in the flight trajectory model was that the great circle path is most fuel efficient path since it is the shortest path between any two points on a sphere. This assumption may also not be true since the most fuel efficient path is dependent on the local weather conditions like wind velocity. Therefore the most fuel efficient path should be calculated by obtaining the wind vectors and flying the trajectory which minimizes the opposition to the wind direction.

The third assumption was that the flight was made under normal operating conditions (ISA conditions), that is the temparature is 15 ^{o}C and the atmostpheric pressure is 1 atmostphere. This is not always true,for example if an aircraft took off from Mexico city,which is 7321 feet above sea level, the fuel required to reach cruise altitude would be much more than if the aircraft took off at sea level, due to decreased lift. The exact amount of fuel consumed can be got by solving the equations of motion of the aircraft, by using the aerodynamic coefficient supplied by BADA OPF file,which would be another improvement over the present version of the flight trajectory model.

The fourth approximation is that the air travel time was assumed to be same for the base and horizon years.This may not be true since as the demand function for air travel grows with time, the number of operations will grow as well. This will lead to conflicts in the airspace and an increase in the travel time over the years. An estimate of this delay could be made by directing the output of the flight tra-

jectory model, to AEM which is an in-house software developed by Virginia tech to calculate the delay due to conflicts in the airspace.

The assumptions in the landside part of the model are:

The first assumption is that general trips have the same characteristics as airport-specific trips. As was mentioned in the previous section the travel velocity is dependent on the travel distance since longer trips will have greater velocity than very short ones. Airport trips in general tend to be longer than say shopping or work trips so can expected to have higher velocity, and so for greater accuracy the analysis should be done with airport-specific trips from NPTS. Unfortunately the lack of airport-specific trips in NPTS precludes any type of airport-specific analysis. Out of 350,000 records in the 1995 NPTS there are only 500 airport trips from which it is statistically impossible to derive any inference about the difference in velocities between airport and general trips. Therefore it was assumed that the airport-specific trips and general trips have the same characteristics.

The second assumption was that all the non-MSA areas (non-urban) areas across the US have the same characteristic. This was done because the non-MSA areas have no specific boundaries or centriods unlike the MSAs due to which is rather difficult to classify the non-MSA regions. This assumption again may not be true because if an non-MSA area is very close to an MSA it may behave differently than a non-MSA region that is far away from any MSA.

The third assumption was that the areas of the MSA regions do not change significantly with time. This assumption was used to predict the population densities of the MSAs into the future. This assumption was made because no data about historical trends in the extent of the MSAs was available. This may not be true since many urban areas undego expansion and their areas change over time and this assumption may introduce appreciable errors in the prediction of the population density.

The fourth assumption is that all MSAs with similar population densities behave in a similar way. In other words, New York and Los Angeles have the same traffic and road characteristics and have the same average travel time. This assumption may also not be valid, if the travel time is strongly affected by local factors such as terrain, conditions of the road and driver characteristics.

4.4) Structure of the Model

As mentioned earlier, the model consists of two parts-an airside part and a landside part. These are descibed in detail below.

4.4.1 Flight Trajectory Model

The flight trajectory model uses the BADA model to calculate the fuel consumed and travel time for an aircraft flying a great circle route. The flight trajectory model uses the PTF file of the BADA model which gives rate of climb,true airspeed (TAS) and fuel consumption at various altitudes. As mentioned in previous section, the PTF file assumes that the flight operates under ISA conditions. A snapshot of the PTF file is included in the next page. The flight trajectory model consists of several separate MATLAB m-files. These are listed below. Each of these source codes is described in detail in the next section.

1) Climb.m- This computes the fuel consumed and the time taken for an aircraft to reach a particular altitude while in the climb phase.

2) Cruise.m - In this file the fuel and time taken for an aircraft to cruise a particular distance is computed.

3) Descent.m - This code gives the fuel and time consumed for an aircraft in the descent mode.

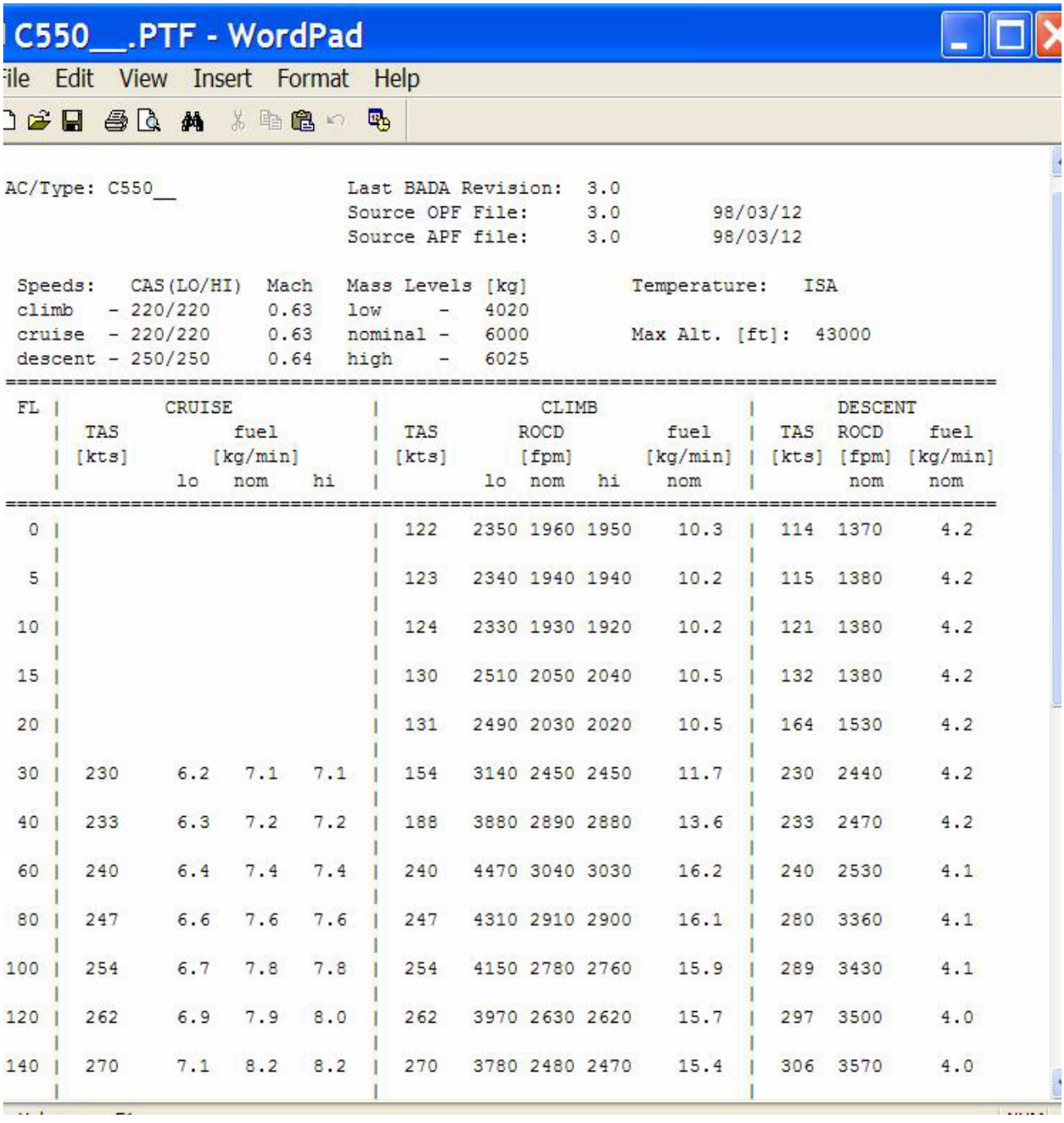
4) Optimize.m - This file gives the modeler the optimum height at which an aircraft should cruise so that the fuel consumed and the cruise time are minimized.

5) Randomize.m - This file takes the aircraft list from BADA and randomly assigns the aircraft to the O-D table

6) ParserforLMI.m - This reads the GA O-D table from LMI and loads it into the memory as a matrix.

7) ParserforPTF_all.m - This reads the PTF file and extracts the rate of climb, TAS and fuel consumed in the climb,cruise and descent phases

8) Sub_flight.m - In this code, the location of the intermediate airport is determined, when the distance between the airports is greater than the maximum range.



C550__.PTF - WordPad

File Edit View Insert Format Help

AC/Type: C550__ Last BADA Revision: 3.0
Source OPF File: 3.0 98/03/12
Source APF file: 3.0 98/03/12

Speeds: CAS(LO/HI) Mach Mass Levels [kg] Temperature: ISA
climb - 220/220 0.63 low - 4020
cruise - 220/220 0.63 nominal - 6000 Max Alt. [ft]: 43000
descent - 250/250 0.64 high - 6025

FL	CRUISE				CLIMB					DESCENT		
	TAS [kts]	fuel [kg/min]			TAS [kts]	ROCD [fpm]			fuel [kg/min]	TAS [kts]	ROCD [fpm]	fuel [kg/min]
		lo	nom	hi		lo	nom	hi	nom		nom	nom
0					122	2350	1960	1950	10.3	114	1370	4.2
5					123	2340	1940	1940	10.2	115	1380	4.2
10					124	2330	1930	1920	10.2	121	1380	4.2
15					130	2510	2050	2040	10.5	132	1380	4.2
20					131	2490	2030	2020	10.5	164	1530	4.2
30	230	6.2	7.1	7.1	154	3140	2450	2450	11.7	230	2440	4.2
40	233	6.3	7.2	7.2	188	3880	2890	2880	13.6	233	2470	4.2
60	240	6.4	7.4	7.4	240	4470	3040	3030	16.2	240	2530	4.1
80	247	6.6	7.6	7.6	247	4310	2910	2900	16.1	280	3360	4.1
100	254	6.7	7.8	7.8	254	4150	2780	2760	15.9	289	3430	4.1
120	262	6.9	7.9	8.0	262	3970	2630	2620	15.7	297	3500	4.0
140	270	7.1	8.2	8.2	270	3780	2480	2470	15.4	306	3570	4.0

Figure 4.6 Sample of the BADA Performance Table File (PTF) (11).

9) Sub_flight_expt.m - This serves the same function as the last program but does it much faster.

10) Max_range.m - This program gives the maximum range of the aircraft.

11) Main_flight.m - This is the main file of the model.

All these files function in conjuction with one another to give the final output of the model which is the fuel and time consumed for a aircraft flying a great circle route. First the file ParserforLMI reads the O-D file given by LMI and loads into the memory into the matrix. Then an aircraft is assigned randomly to complete the trip. Then the climb and descent routines are executed to see whether the distance between the origin and destination is enough to enable the aircraft to climb to its optimal height and descend. If the distance is too small then the height to which the aircraft can climb is dependent on the distance. In this case the height is taken as the minimum of the optimal cruising height and the maximum height that the aircraft can climb due to the distance constraint. After determing the cruise height the fuel consumed and time taken are calculated using the cruising height as an input. If the distance between the two airports is greater than the distance required to climb and descend but less than the maximum range of the aircraft, then the cruise height is taken as the optimal cruising height and the cruise distance is calculated as the difference between the distance from origin to destination and the sum of the distances taken to climb to the optimal height and descend from the optimal height to the ground. If the distance between the origin and destination is greater than the maximum range of the aircraft, then an intermediate airport is selected. The airport nearest to the point where the aircraft runs out of fuel is selected. Also the distance between the starting point and the selected airport has to be less than the maximum range. The search process has to be made efficient since searching for a candidate airport among the 3346 airports in the US every time the great circle distance exceeds the maximum range of the aircraft is extremely time-consuming. The selection is further complicated by the fact that if the great circle distance is a little greater than an integral multiple of the aircraft range, then one has to design a suitable heuristic to divide the distance into an equal number of legs, since the aircraft cannot fly a very small distance. The entire process is shown as a flowchart in the Figure 4.7. A detailed description of some of the source codes is included in the next section.

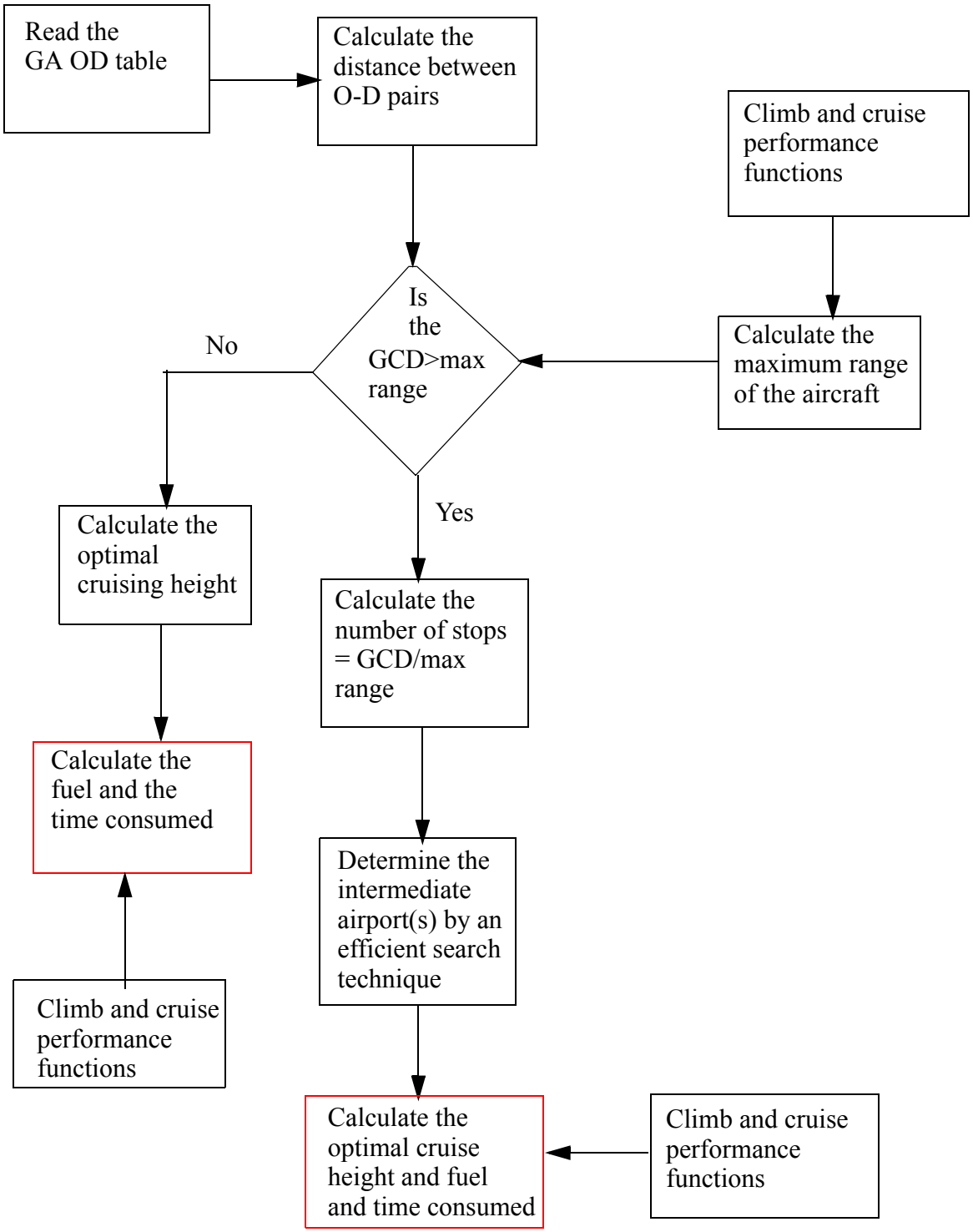


Figure 4.7 General Aviation Travel Time Model.

4.4.1.1 Climb Performance Function

The climb performance function uses the rate of climb and fuel consumption given in the PTF file to calculate the time and consumed to reach a particular height. The actual rate of climb varies continously but for simplicity, the rate of climb for any given height intreval is taken as the arithmetic mean of the rates of climb of the end-points of the height intreval and the rate of climb at any intemediate height is got by interpolation.The same procedure is adopted for calculating the fuel consumption and the true airspeed. Therefore given any height this function will give the time, horizontal distance and the fuel consumed to reach that height. These relationships are shown in Figure 4.8.

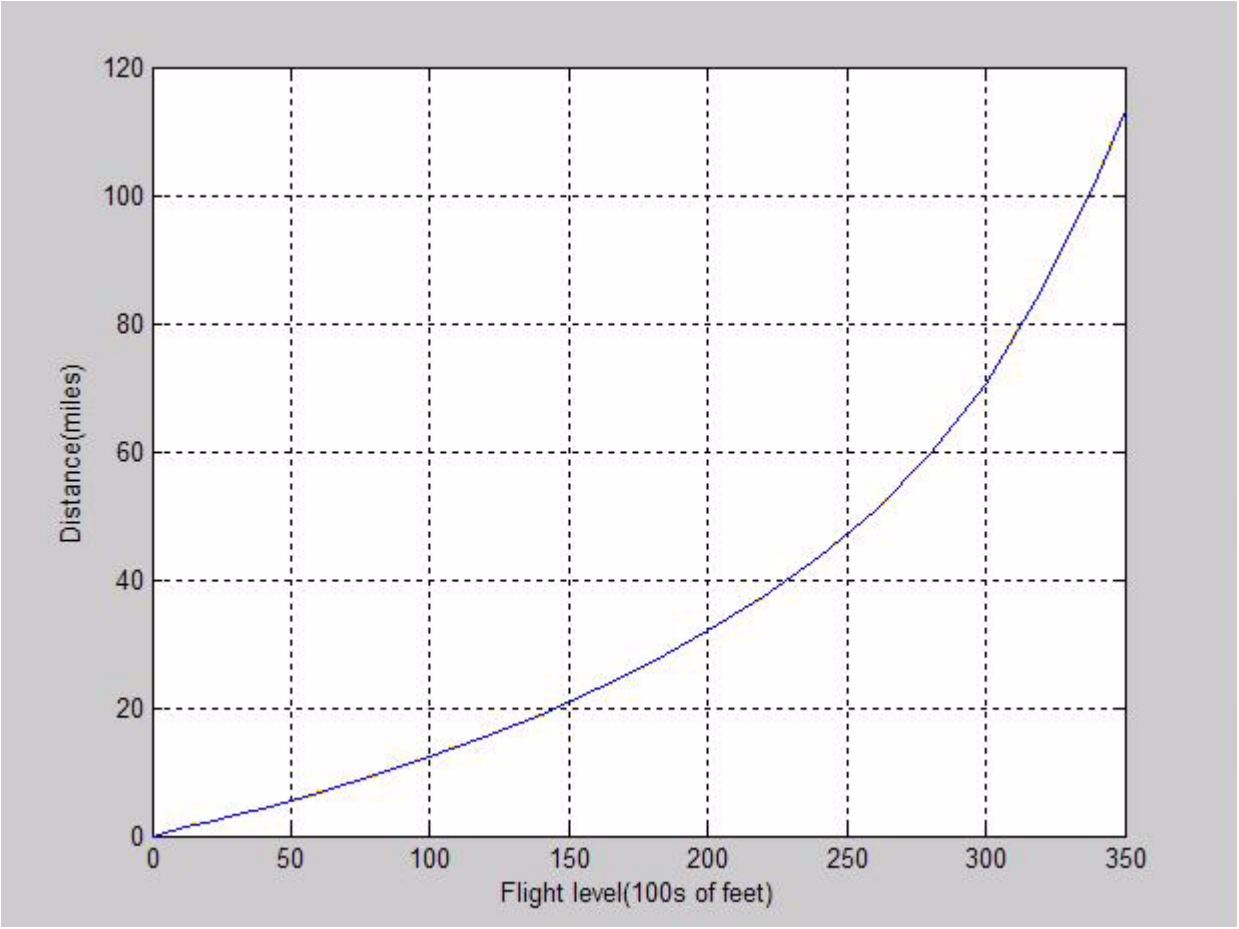


Figure 4.8 Distance to Reach a Particular Altitude for the Cessna Citation.

4.4.1.2 Cruise Performance Function

The cruise performance function uses the fuel consumption and TAS at various altitudes to determine the amount of fuel consumed with distance for an aircraft in the cruise mode. The cruise fuel consumptin from the BADA model is compared with the Bruguet range equation and a plot of the models is given in Figure 4.9.

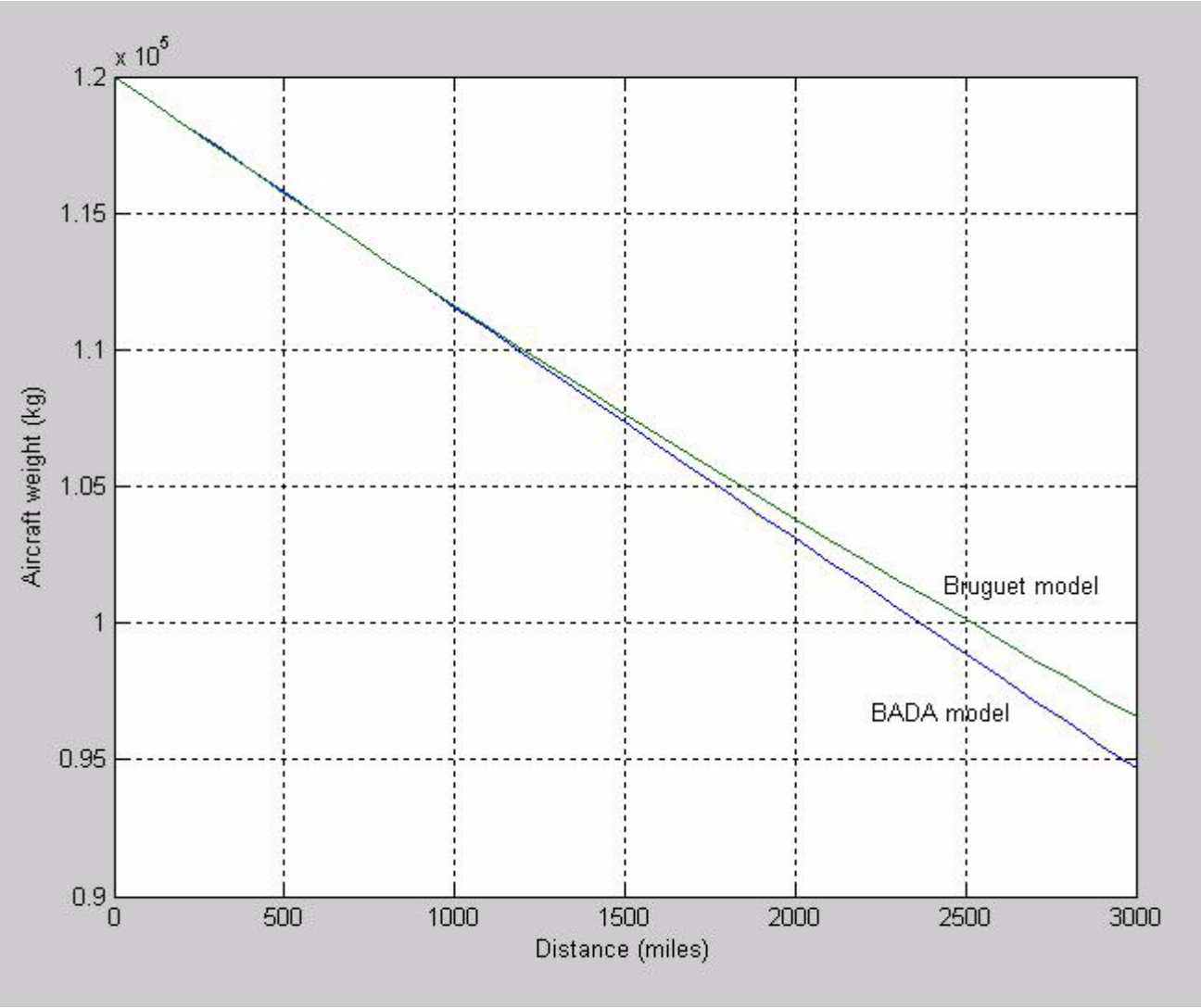


Figure 4.9 Comparison of the BADA and Bruguet Range Equations for an Airbus A-310 in Cruise-Mode.

4.4.1.3 Optimal Height Generator

This program finds the optimal cruising height so that the fuel consumed and flying time are minimized. Since the minimum fuel consumption and minimum flying time occur at different heights, the program minimized a composite index which is a combination of fuel and time to obtain cruising height. The variation of the composite index with altitude for a specific mach number is shown in Figure 4.10

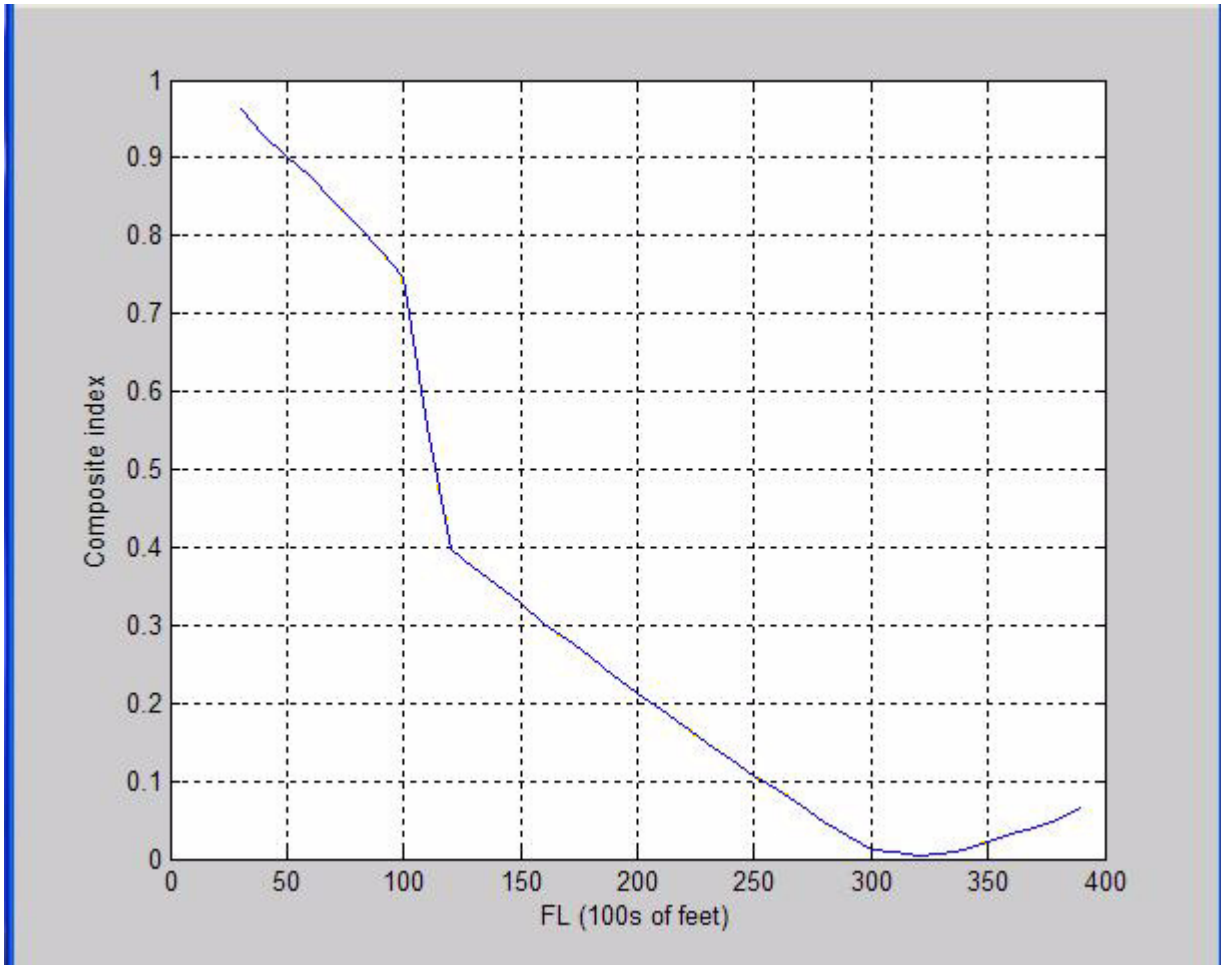


Figure 4.10 Variation of the Composite Index with Height.

4.4.1.4) Descent Performance Function.

The logic behind the descent performance function is the same as in the climb and cruise performance functions, that is the true airspeeds and the fuel consumption rates at different altitudes are used to calculate the horizontal distance and fuel consumed for an aircraft to descent from a particulat flight level to the ground. The variation of horizontal distance required to reach the ground for with flight level is shown in Figure 4.11.

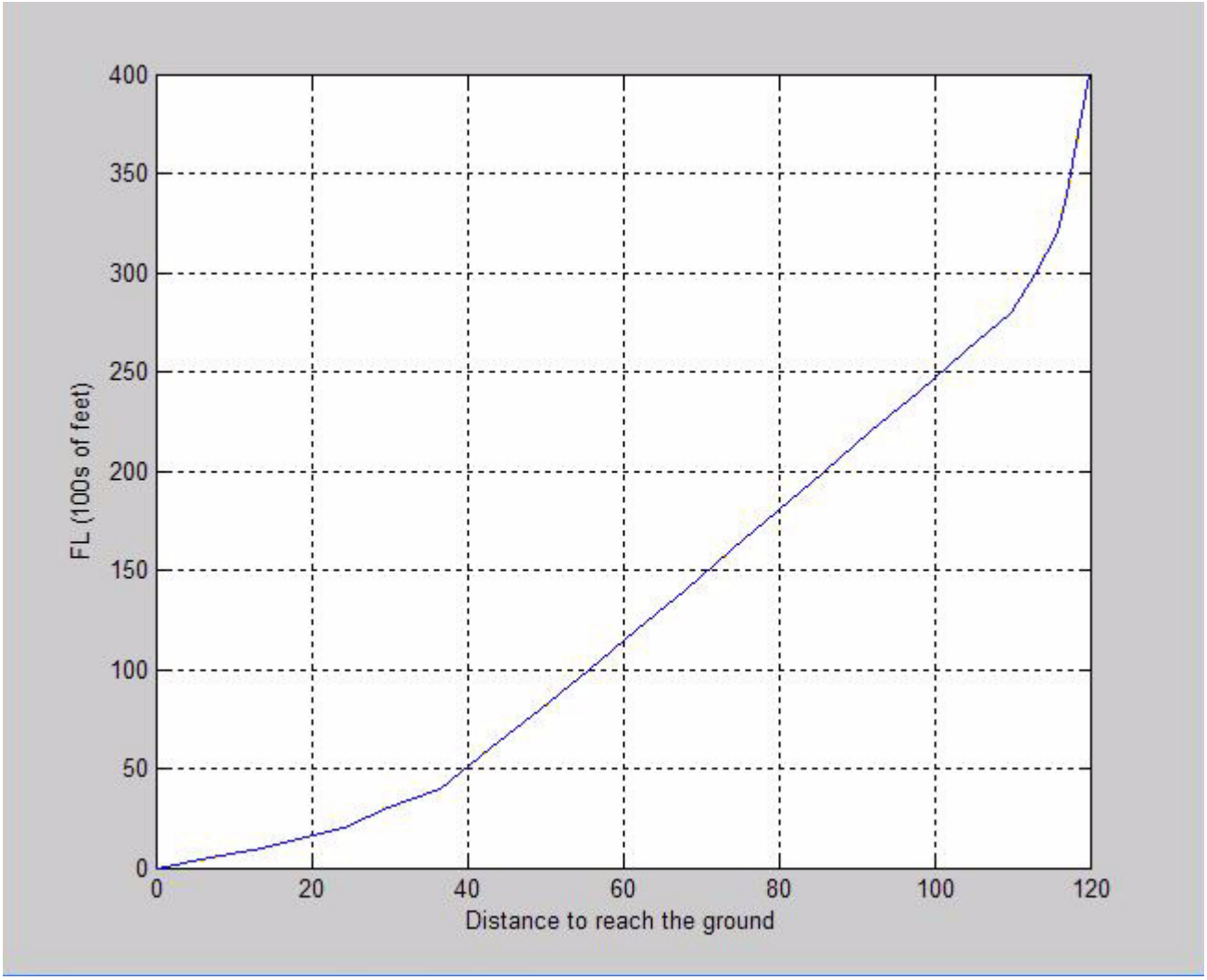


Figure 4.11 Distance to Descend from a Particular Altitude to Ground For the Airbus A 310.

4.4.2 Commercial Aviation Mobility Model

The flight trajectory model measures the travel time for air passengers taking the GA mode. For commercial aviation the flight trajectory model would not give a good assesment of mobility, since commercial flights from origin to destination rarely follow a great circle path, but are routed through a hub, thus increasing the travel time. Therefore the mobility for commercial aviation for the baseline year was measured with the help of DB1B and OAG. The DB1B was used to obtain the network structure and the OAG was used to determine the time spent for each flight including stopover time at intermediate airports.

First a function was written to determine the network structure of the NAS. For this the Coupon file of the DB1B was used, since it gives the intermediate airports. The output of this program is a 419 by 419 MATLAB struct array where each cell(i,j) contains the number of passengers from from airport *i* to airport *j* and all the intermediate airports that the passengers are routed through. The next step was to determine the total travel time for passengers travelling from airport *i* to airport *j*. The travel time was obtained from OAG. The OAG has the travel time information for all direct flights. This information was combined with the network topology derived from DB1B to calculate the total travel time(in passenger hours) for all travelers across NAS. This is explained more clearly with an example.

Consider for example 1000 passengers traveling from Roanoke, VA (ROA) to San Fransisco (SFO). From the DB1B we know that 700 passenegers are routed thro Chicago O' Hare (ORD) and 300 through Baltimore-Washington international (BWI). The OAG database is searched for flights from ROA-BWI and BWI-SFO and their travel times are obtained. But neither DB1B nor OAG give information about the connecting flights, therefore it was assumed that the pair of flights with minimum time differential were the two flights that the passenger took for going from ROA to SFO. For example if a flight took off from ROA at 8:30 am and reached ORD at 10:30 am and the next flight after 10:30 am from ORD-SFO took off from ORD at 11:45 am and reached SFO at 3:00 pm then total travel time for all passengers on the ROA-BWI-SFO route was 700*6.5 = 4550 passenger hours. The same procedure was repeated for all the O-D pairs across NAS and mobility for commercial air travelers across NAS was calculated.

4.5 Network Analysis Model.

The network analysis model calculates the mobility of the ground network. This was done in two parts. First the CDF's for travel speed were derived from NPTS. The second part consisted of deriving the travel distance CDF's from ATS data. This was done by Mr Lee at the NVGC of Virginia Tech. Finally the two CDF's were convoluted to give the travel time CDF.

Initially multiple linear regression was used to model the variation of travel velocity. The independent variables used were population density, peak/non-peak characteristic, MSA/non-MSA characteristic, type of transit and travel distance. However the R-square values for MSA and non-MSA areas were 0.3 and 0.13 respectively and subsequent addition or removal of independent variables did not cause any significant improvement in the R-square value. Therefore the multiple regression approach was abandoned since the variation in travel velocity was too large.Thus travel velocity was modeled as a random variable and the distribution was derived directly from NPTS data.

The next step was to find explanatory variables to bin the raw data into different categories. The variables considered were population,population density, road density, travel distance, MSA/non-MSA characteristics and peak/non-peak characteritics. However some of these variables were eliminated after performing statistical analysis on the raw data. As mentioned earlier in the chapter the correlation coefficient of speed vs population density was much higher than the correlation coefficient for speed vs population. Therefore population density was taken as an explanatory variable and population was dropped. It was hypothesized that road density could also be dropped from the list of independent variables since in a developed country like the US population density and road density would be highly correlated. To test this hypothesis the road density for each county was extracted from HPMS and compared against the population density obtained from the Census data. The plot is shown in Figure 4.12. It was found that the correlation coefficient was 0.97 thus proving the hypothesis correct.

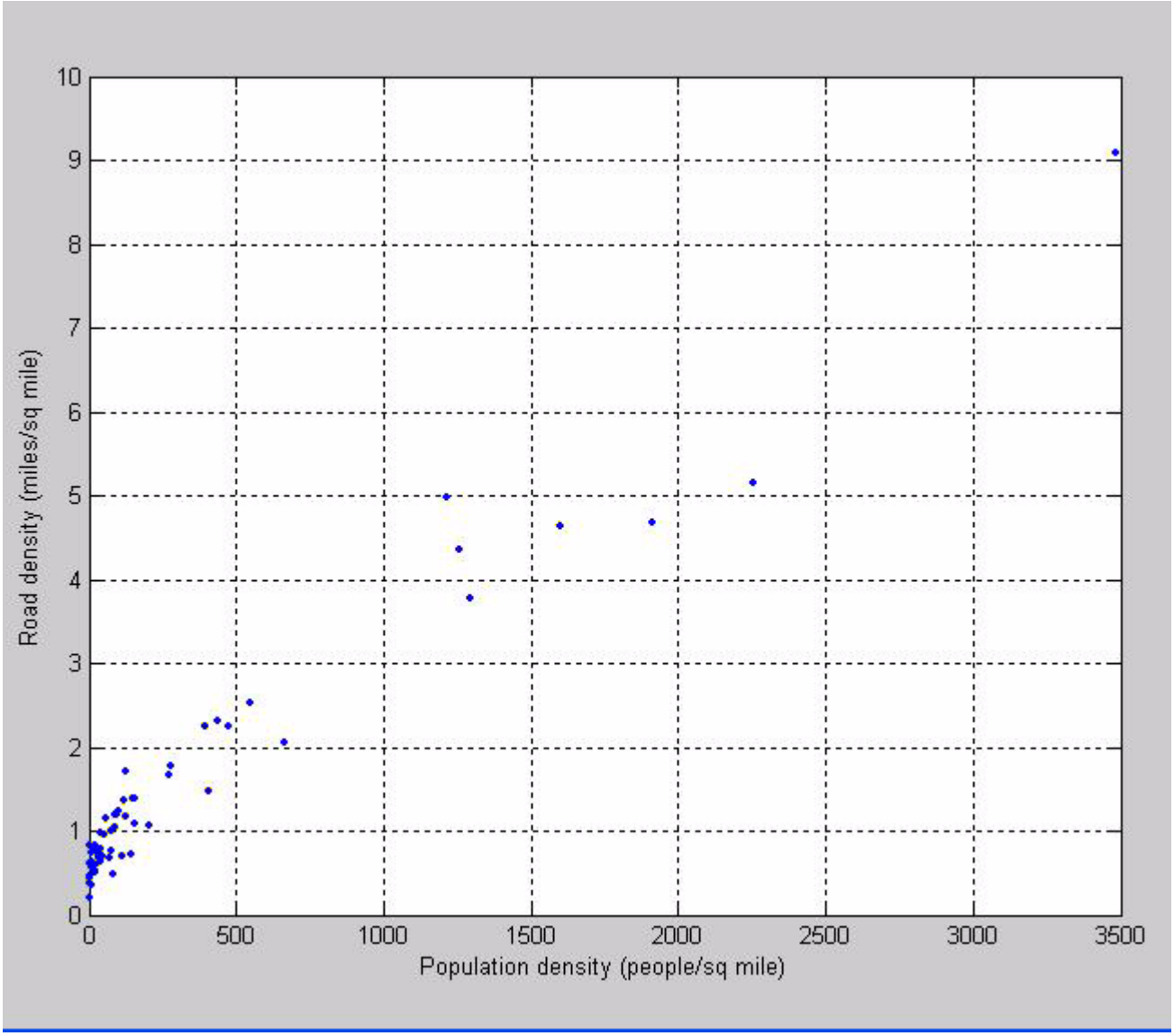


Figure 4.12 Population Density vs Road Density Plot for The State of California.

Therefore the three explanatory variables taken were population density, peak/non-peak characteristic and MSA/non-MSA characteristic. The raw data had to be seperated into bins using some statistical test of significance. The various tests that are commonly used are the T-test and F-test. However in this case the raw speed data was found to be non-normal at a 95% level of significance. Therefore a non-

parametric test called the Wilcoxon rank sum test was used to separate the raw data into bins. The MSA data was first divided to two groups based on peak and non-peak characteristic. Then each of these groups were divided into five subgroups based on population density. All the non-MSA data was grouped into a single CDF since the peak and non-peak characteristics for non-MSA areas were found to be the same within a 95% level of significance and no demographic or geographic boundaries could be found for non-MSAs.

Another factor which had to be considered was travel distance. This is because travel distance and travel velocity were found to be positively correlated, that is longer trips were faster than short ones. Another reason was that people cannot live very close to an airport because of noise and other concerns. Therefore a lower limit for travel distance had to be found. This was done by collecting all the airport-specific trips from NPTS and determining the lower limit at 95% significance. This value was found to be 2 miles. The upper limit was set at 100 miles since it was highly likely that trips greater than 100 miles were intercity trips. Thus all trips less than 2 miles and greater than 100 miles were eliminated. The final output of the model were 11 CDF's- 5 for peak MSA characteristics, 5 for non-peak characteristics and 1 for non-MSA areas. The CDF's are given as table function since no suitable interpolating function was found.

The final step was the convolution of travel speed CDF's with the travel distance CDF's to get travel time CDF's. For non-MSAs, the travel distance CDF was extracted from the ATS data. For MSAs two independent approaches were adopted. The first method was macroscopic and involved generating a single travel distance CDF from airport-specific NPTS data and convoluting the travel speed CDF's with the single travel distance CDF obtained above to get travel time CDF's.

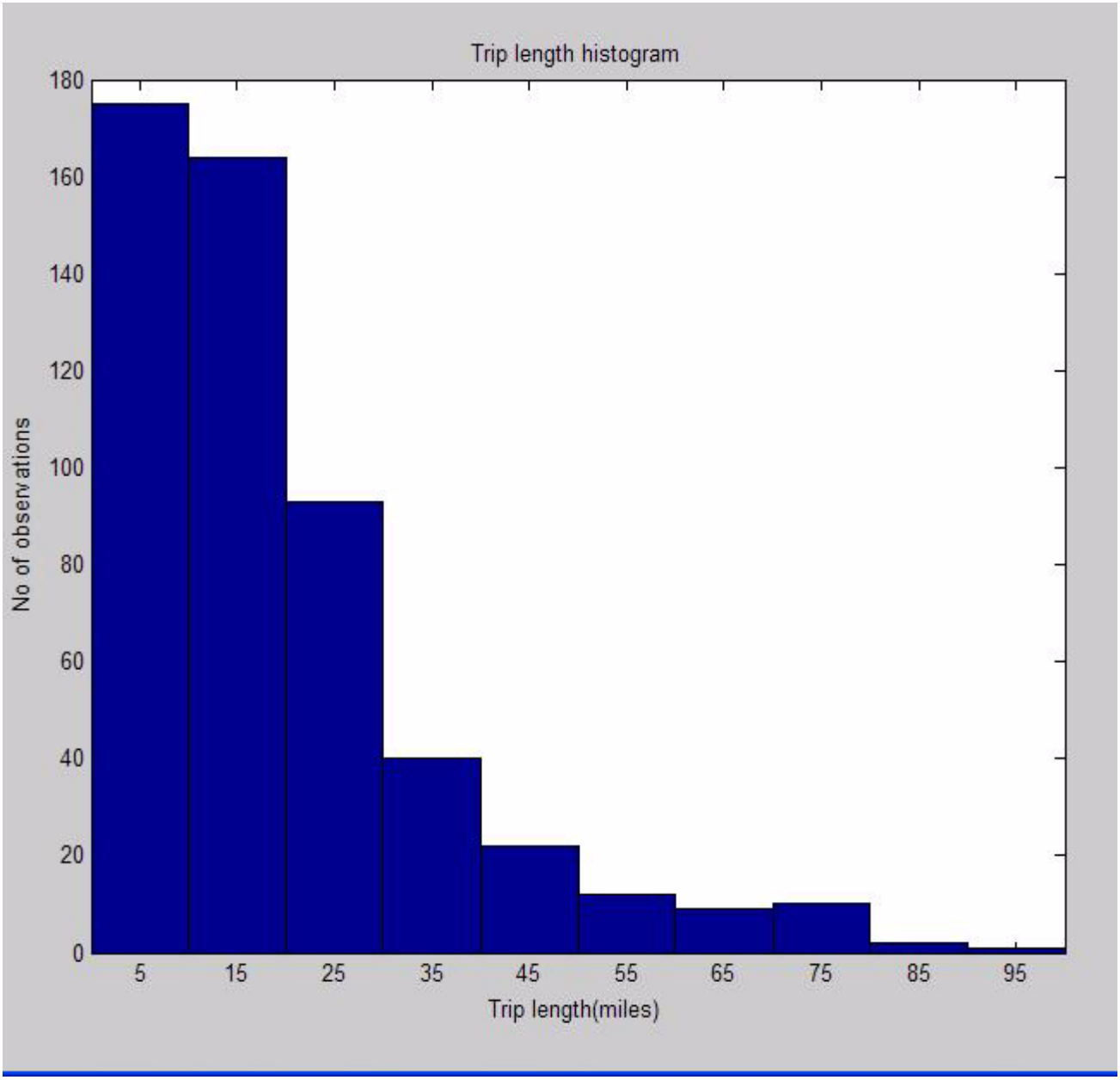


Figure 4.13 Trip Length Histogram Derived from NPTS.

The second approach was more microscopic and involved extracting travel distance CDF's from ATS data. The explanatory variable was the number of airports within a 40 mile radius, since if there are more airports within a certain distance of the MSA, the travel distance to the airports would also be less. In this case three travel distance CDF's were derived and the convolution was done for each MSA separately depending on its population density and the number of airports within a 40 mile radius.

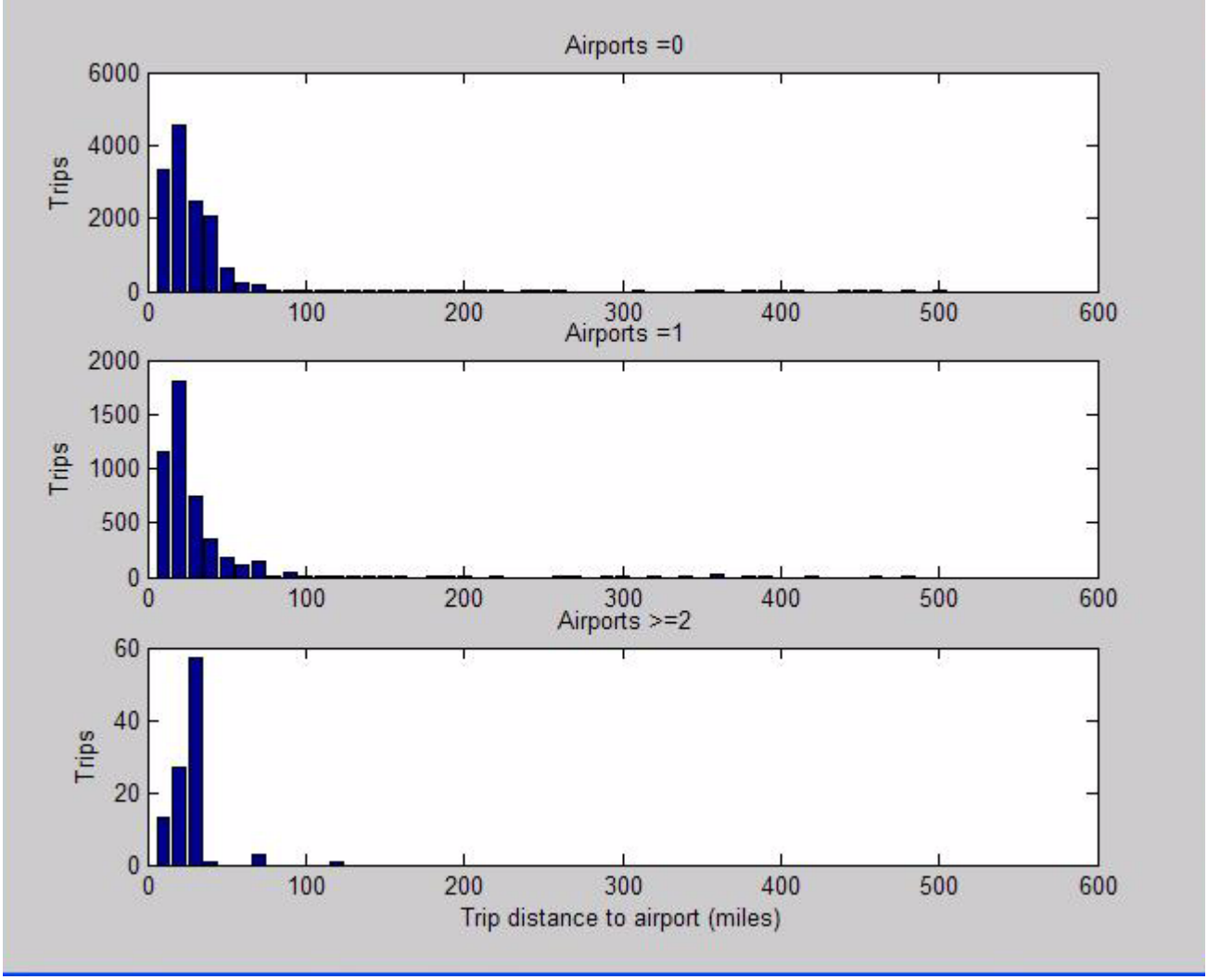


Figure 4.14 Trip Length Distribution for MSA Airport Trips (Lee Myunghung, 2002).

The entire process can be viewed as a flowchart as shown in Figure 4.15.

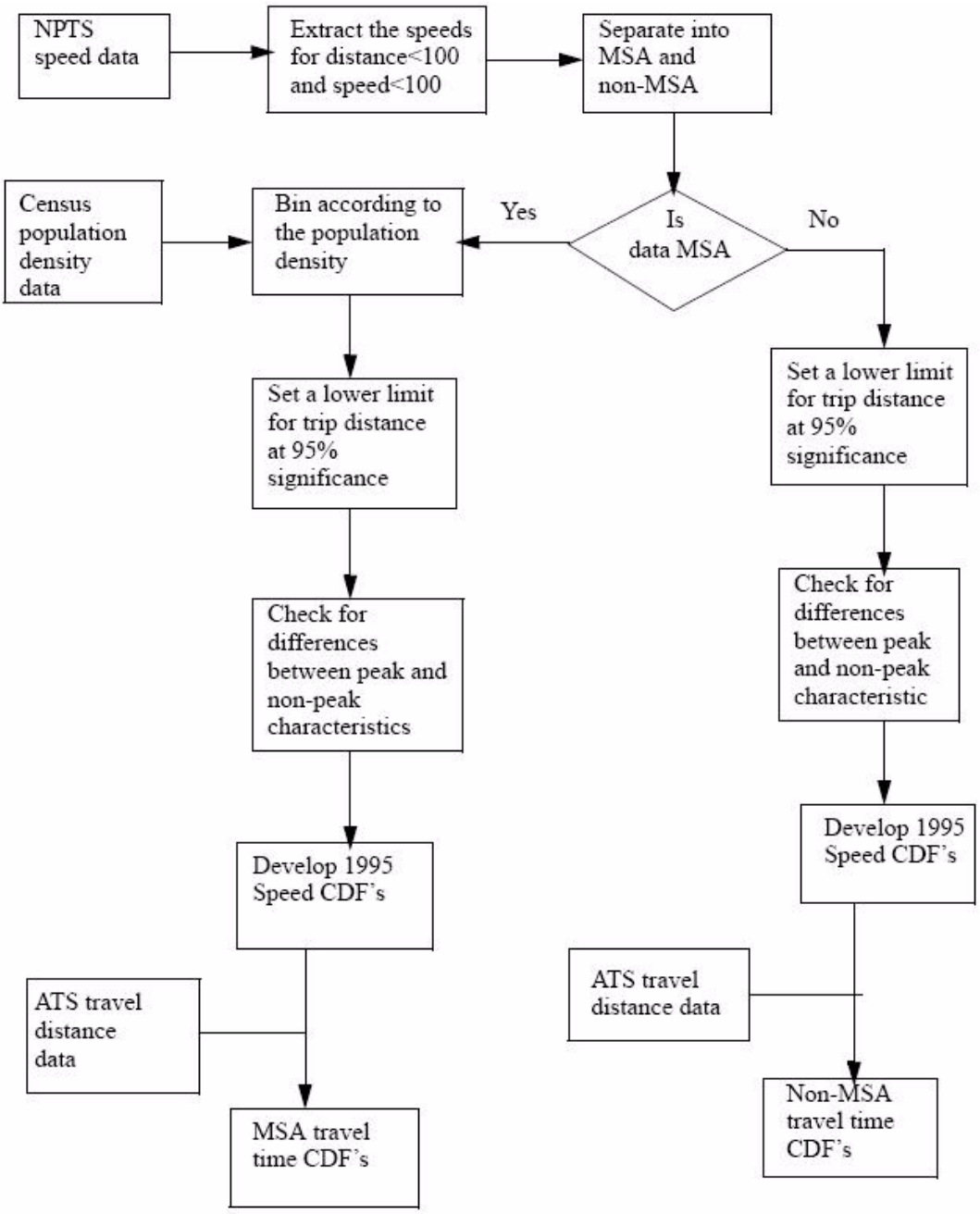


Figure 4.15 Network Analysis Model.

4.6 Horizon Year Modeling

The final step in the model was to measure mobility metrics in the year 2020. The mobility of the air-side part of the network was assumed to be the same. The reduction in travel speed over the years was assumed to be caused by an increase in the population density and all the factors such as differences in peak and non-peak characteristic and distribution of travel distance were assumed to remain the same. Historical trends in travel speeds were obtained from the Urban Mobility study performed by the Texas Transportation Institute which gives the mean value of travel speeds for 75 MSAs from 1980 to 2000. The trend in population of these MSAs was obtained from Woods and Poole which also predicts the population of each MSA, County and state till 2025. The population densities of the 75 MSAs from 1980 to 2025 was found by dividing the population of each MSA by the area of the MSA in the year 2000. As already mentioned in the beginning of the chapter, the assumption was that the areas of MSAs do not significantly change with time. Next the speeds of each of the 75 MSAs from 1980 to 2000 were regressed against the corresponding population densities. The regression equations whose R-square value fell below 0.6 were rejected. The remaining equations were used to obtain the travel velocities in the year 2020 and the travel velocities in the year 2020 were divided by the travel velocities in the year 1995 to get the correction factors for the horizon year. Then the correction factors were binned depending on the population density of the corresponding MSA in the year 2000 and average correction factors for each of the population density categories were obtained. For the non-MSA regions the correction factor was assumed to be the same as the correction factor for the MSAs having the lowest population density since no historical trends in travel speeds for non-MSA areas was available. Finally the 1995 speed CDF's were multiplied by the corresponding correction factors to get travel speed distributions in the year 2020. The variation of travel speed with population density and time is shown graphically in Figure 4.16.

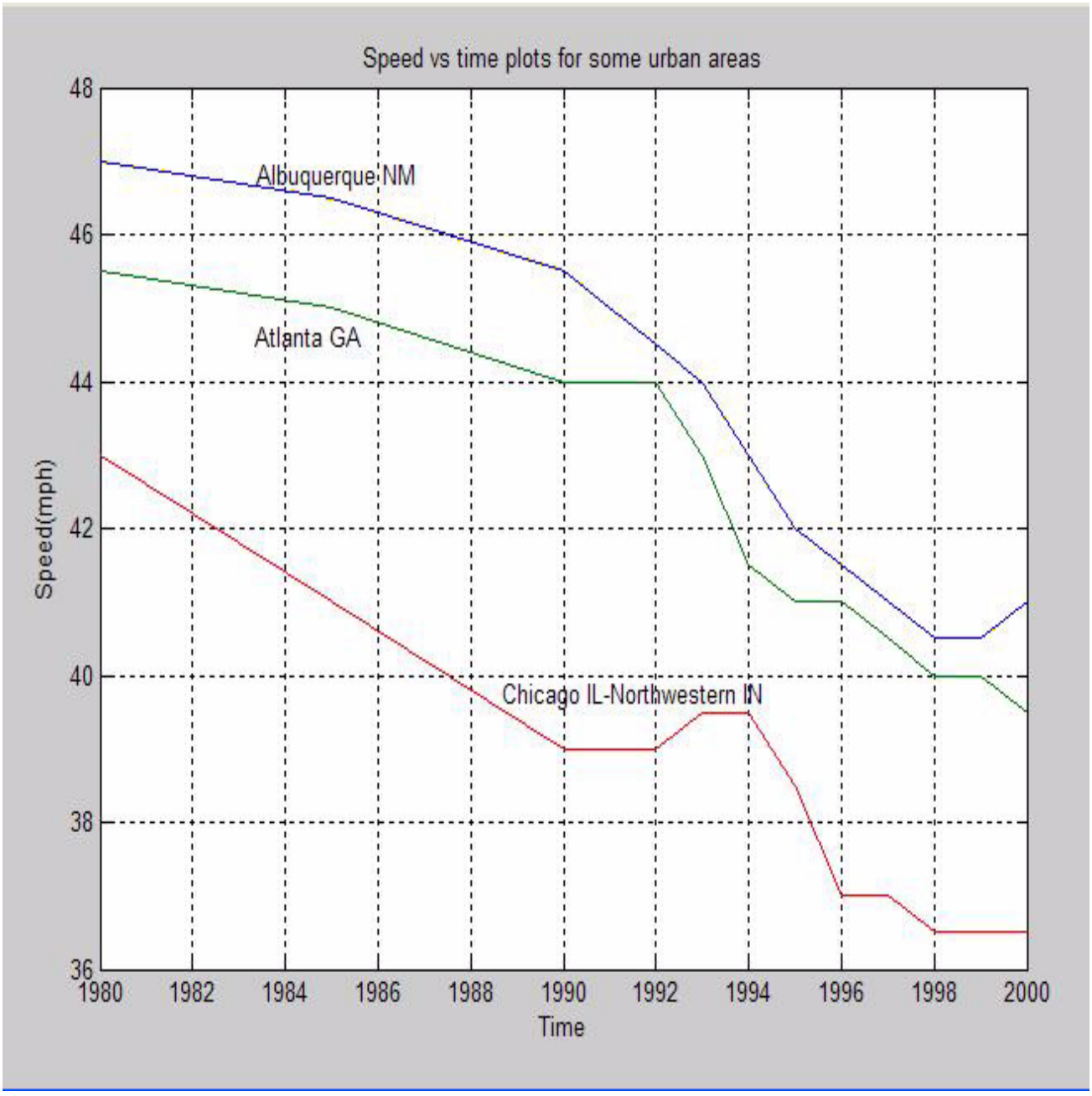


Figure 4.16 Variation of Speed With Time For Some MSAs.

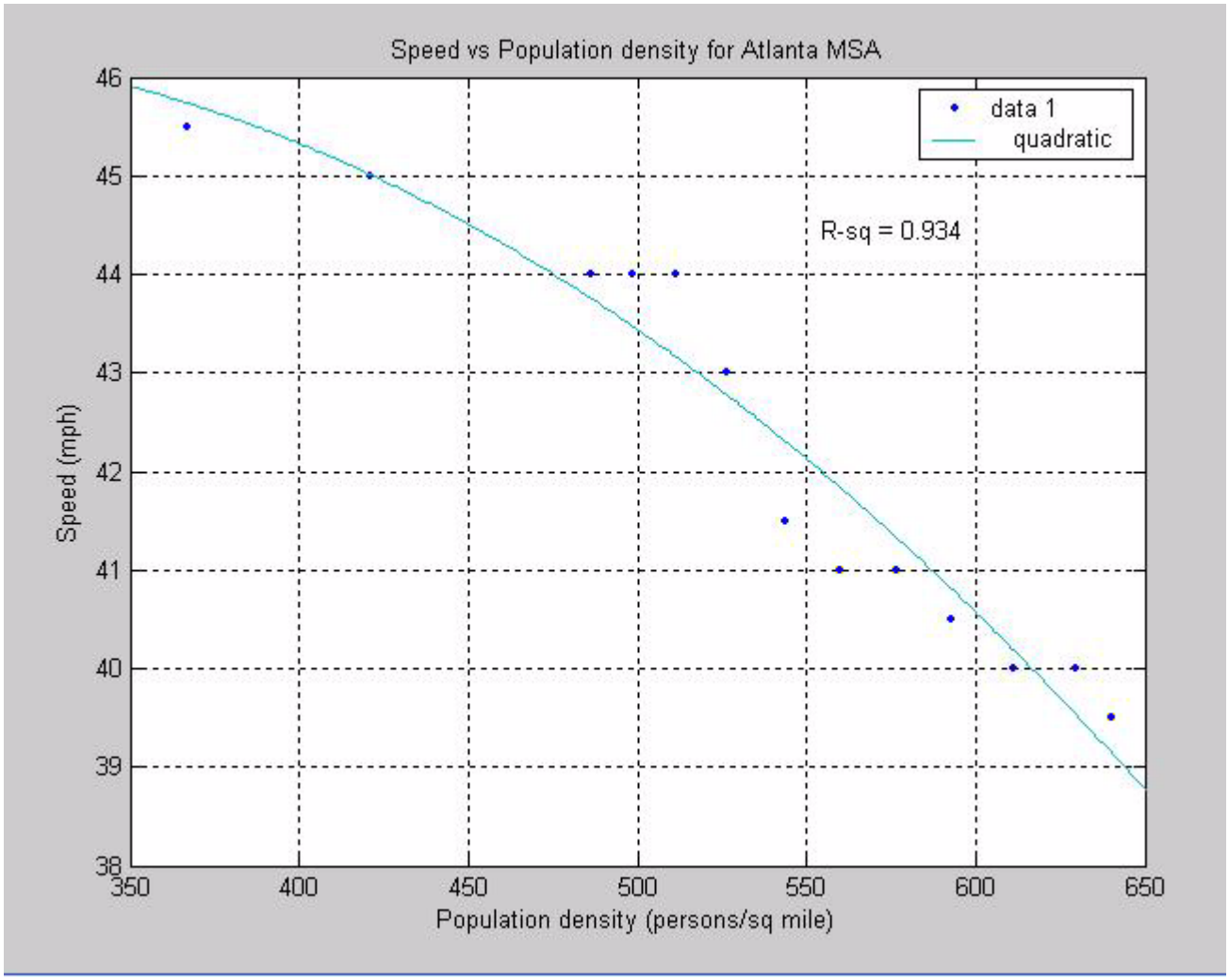


Figure 4.17 Variation of Speed With Population Density and the Corresponding Curve Fit.

CHAPTER 5 Model Results

5.1 Model Results

The network analysis model was run on 1995 NPTS data and the speed and the speed CDF's were obtained.This was convoluted with the travel distance obtained from NPTS for the macroscopic approach and from ATS for the microscopic approach and the travel time CDF's were obtained. The average value of travel time for each MSA was then multiplied by the number of air travelers passing through each MSA to get the number of passenger-hours that the air travelers spend on the ground. The passenger-hours spent in flying and waiting at airports is obtained by multiplying the air travel time from each origin to destination by the number of passengers from the same origin to destination. The results are summarized below.

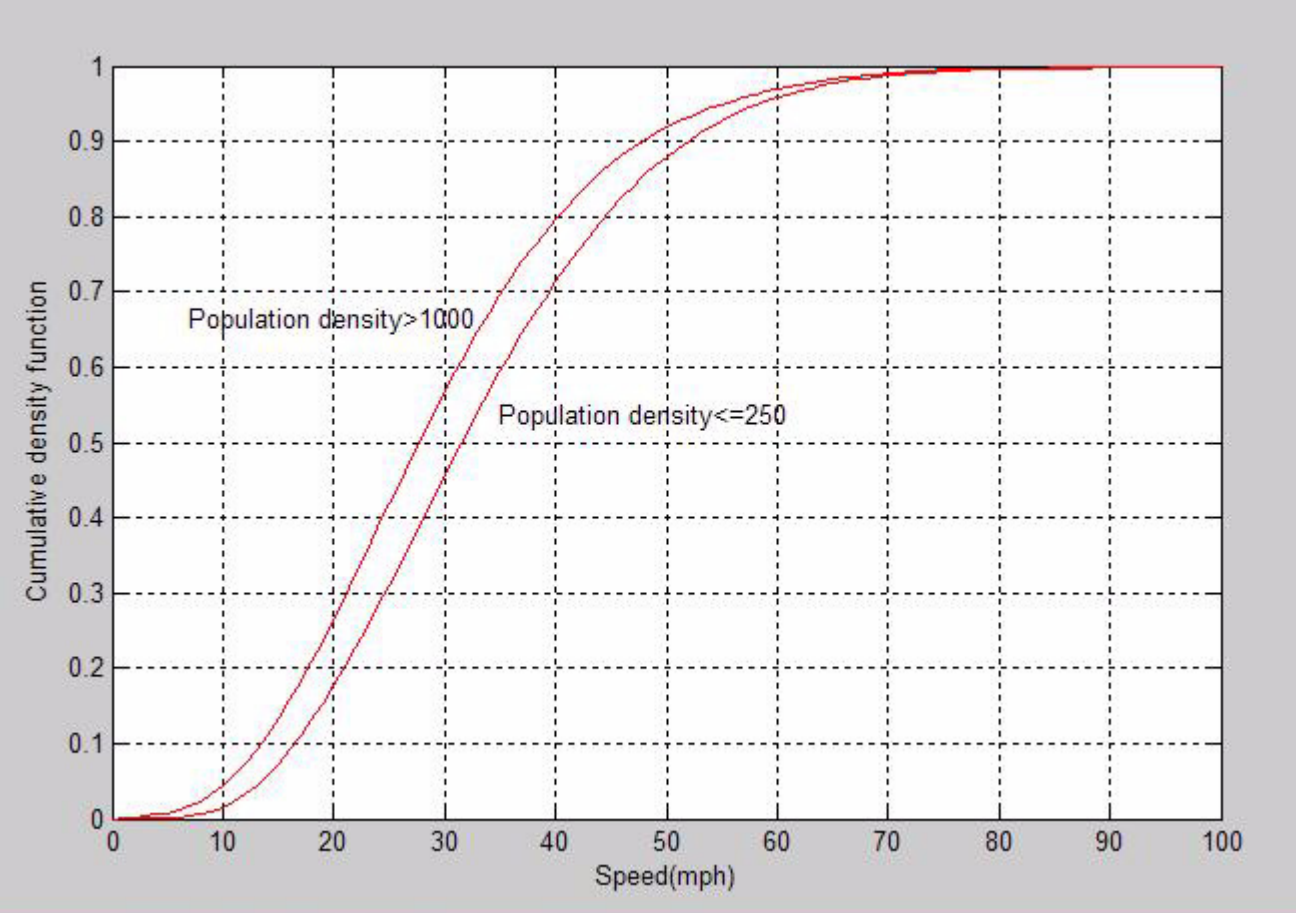


Figure 5.1 Cumulative Density Function for 1995 MSA Peak Characteristic.

Figure 5.1 gives the plot of the CDF's for the 1995 peak speeds for Metropolitan areas. Peak trips are defined as any trip made from 9 AM to 7 PM. These speeds were extracted from the Nationwide Personal Transportation Survey. These were used for determining the travel time CDF's.

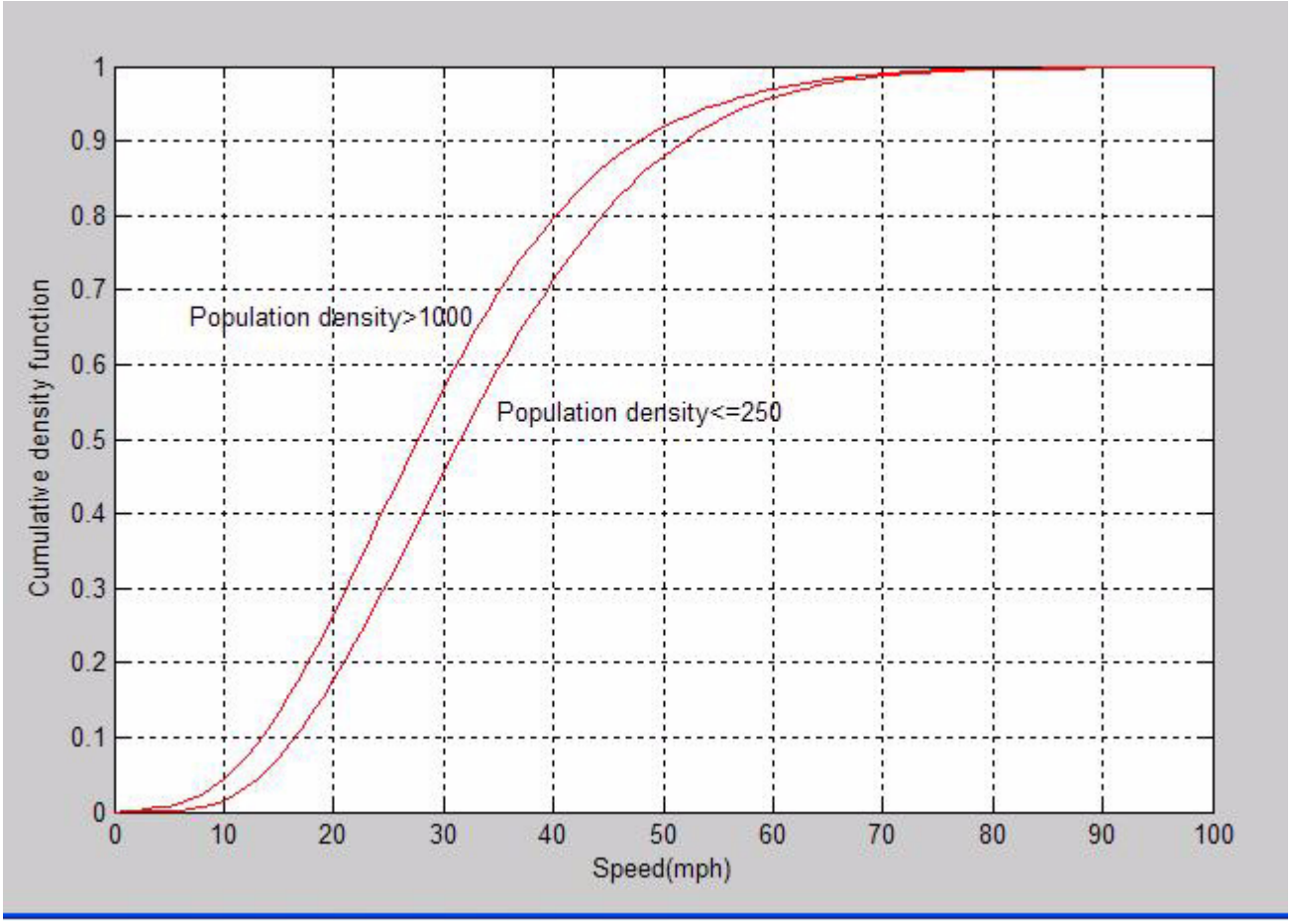


Figure 5.2 Cumulative Density Function for 1995 MSA Non-Peak Characteristic.

Figure 5.2 gives the plot for the travel speeds for Metropolitan areas during peak hours. Off-peak trips are taken as trips made from 7 PM to 9 AM. The average speed of the non-peak trips are more than the peak trips which is expected.

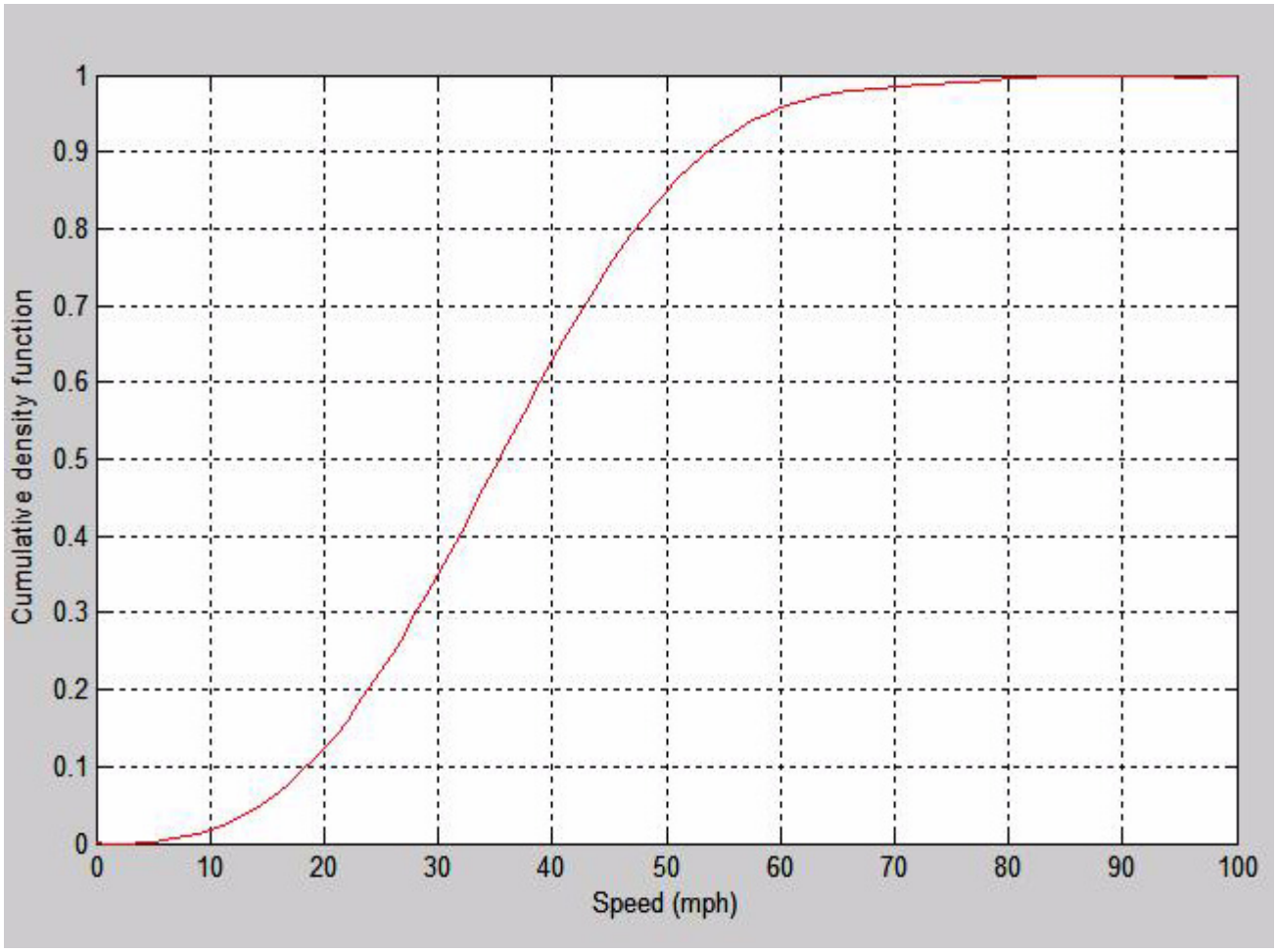


Figure 5.3 Cumulative Density Function for 1995 Non-MSA Areas.

Figure 5.3 gives the plot of speeds for the non-Metropolitan areas. All the non-MSA data was developed into a single CDF since peak and non-peak speeds were found to be the same. The average speed was higher than both the peak and non-peak MSA average speeds.

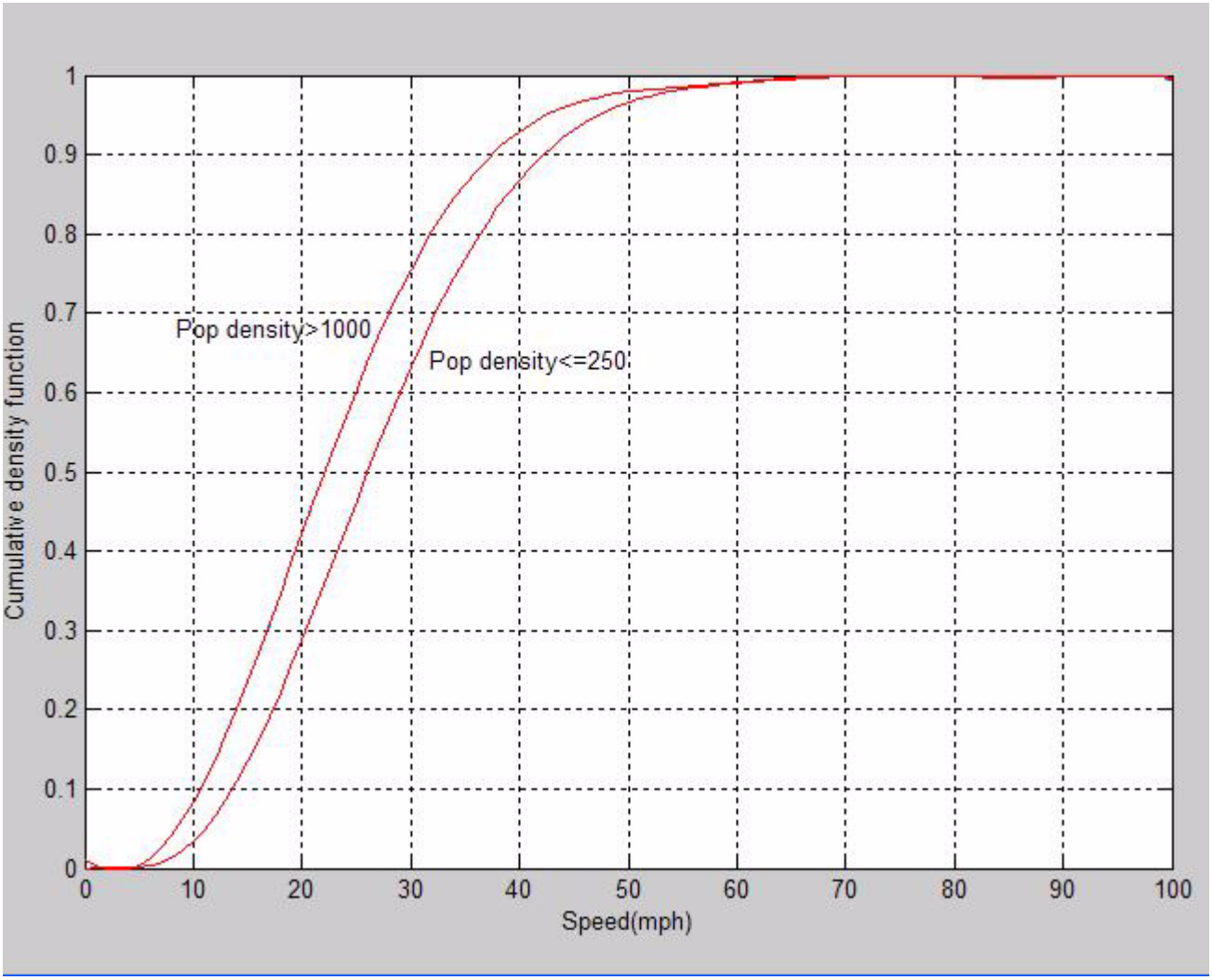


Figure 5.4 Cumulative Density Function for 2020 MSA Peak Characteristic.

Figure 5.4 gives the plot for MSA peak speeds for the horizon year. It is derived by extrapolating the historical trends in speed with population trends over the years from 1975 to 2000 which is derived from the Woods and Poole Data (13).

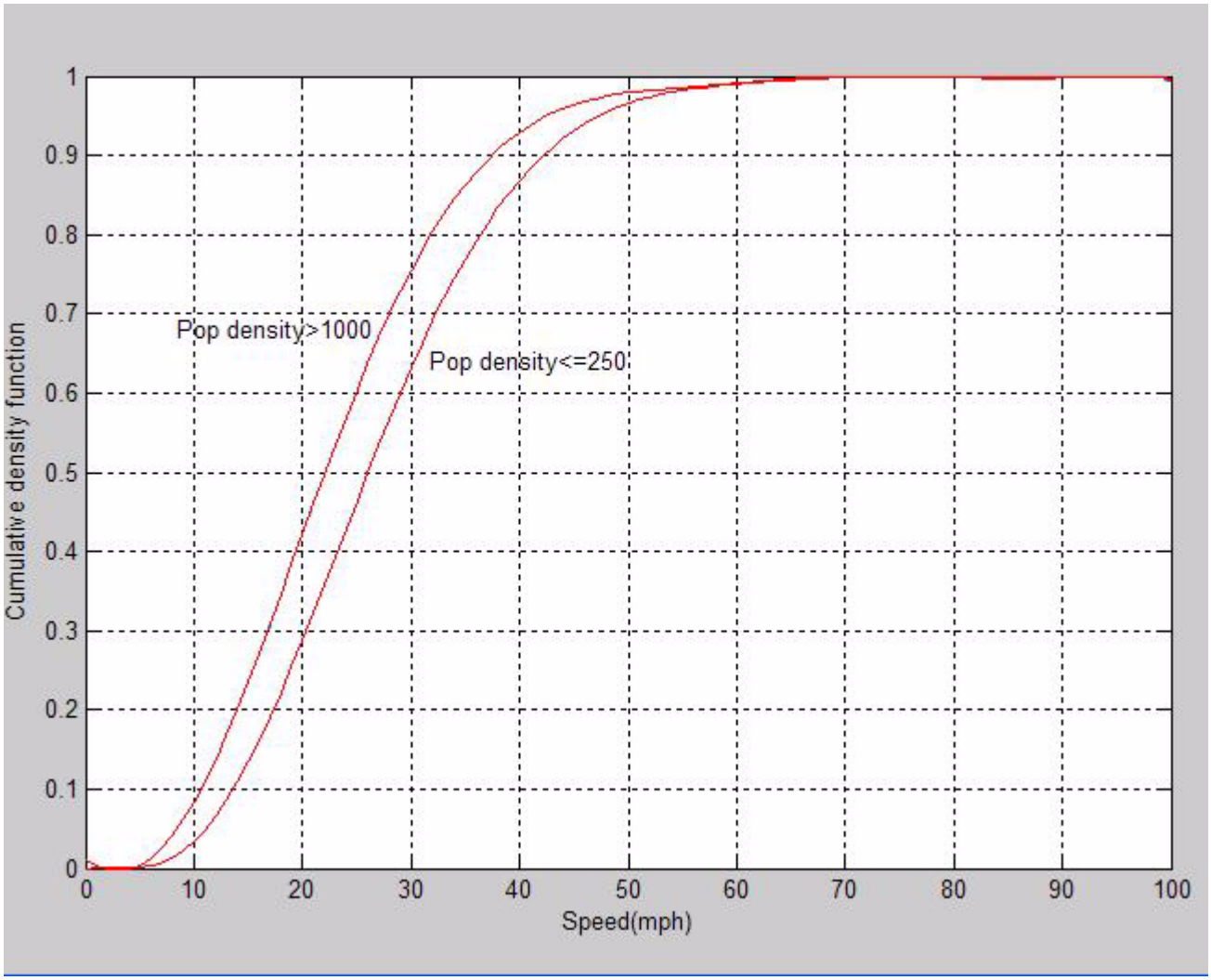


Figure 5.5 Cumulative Density Function for 2020 MSA Non-Peak Characteristic.

Figure 5.5 gives the CDF's for MSA areas at non-peak hours for the horizon year. They are derived by extrapolating the historical trends in speed with population trends over the years from 1975 to 2000 which is derived from the Woods and Poole Data (13).

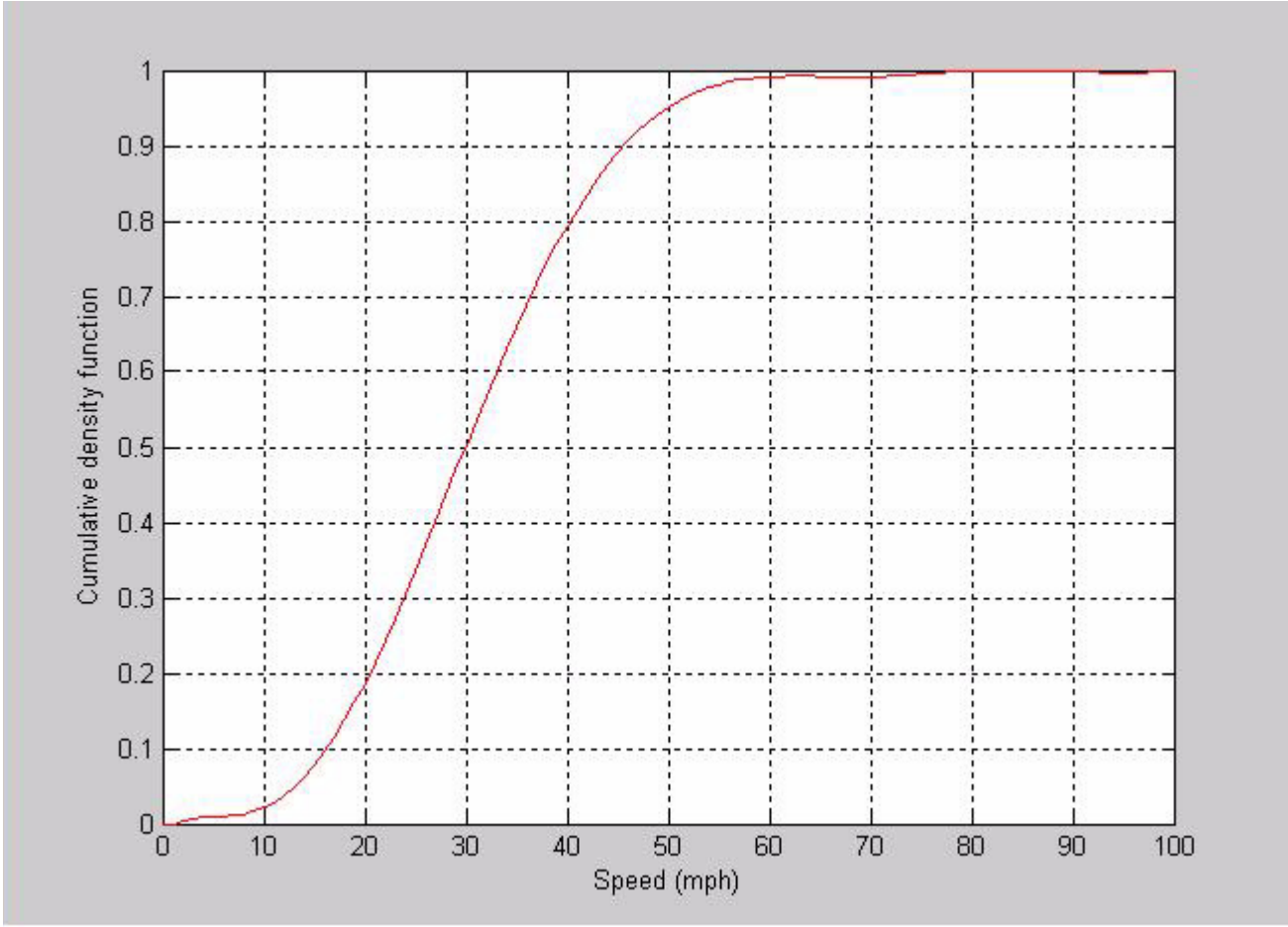


Figure 5.6 Cumulative Density Function for 2020 Non-MSA Areas.

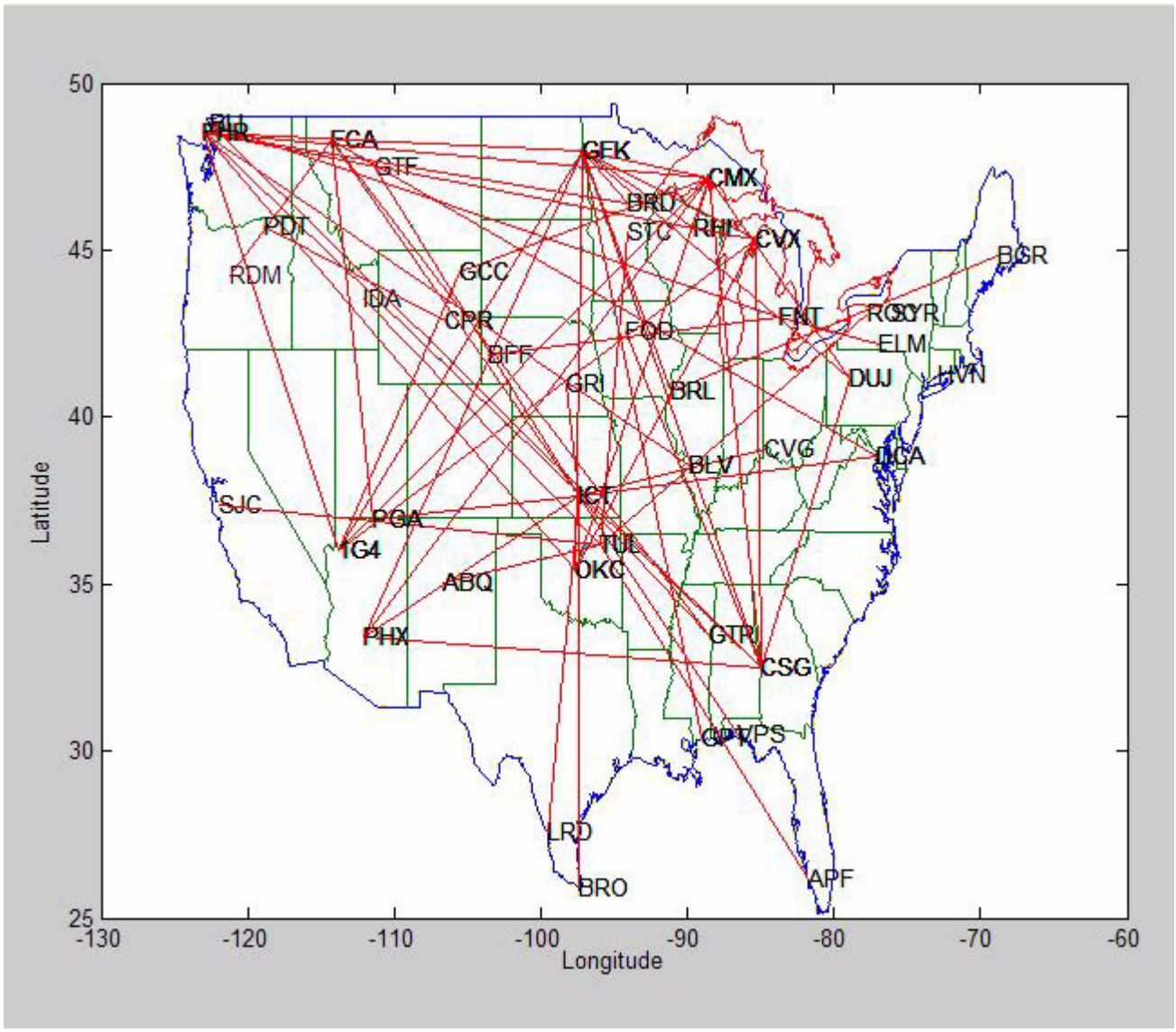


Figure 5.7 Passenger Flows for the Top 100 Business Markets in the United States.

Figure 5.7 gives the passenger flows for the business travelers in the US. This was extracted from the Coupon file of the Airline Origin and Destination Survey (DB1B). From Figure 5.7 it is seen that the main airports like ATL, DFW are not in the top 100 business markets even though they are the top airports in terms of traffic. This could be due to two reasons,one being that a large proportion of the traffic

is itinerant and therefore dont show up in the above graph since the above graph depicts Origin-Destination flows. The other reason could be that the percentage of business flights in the major airports is low.

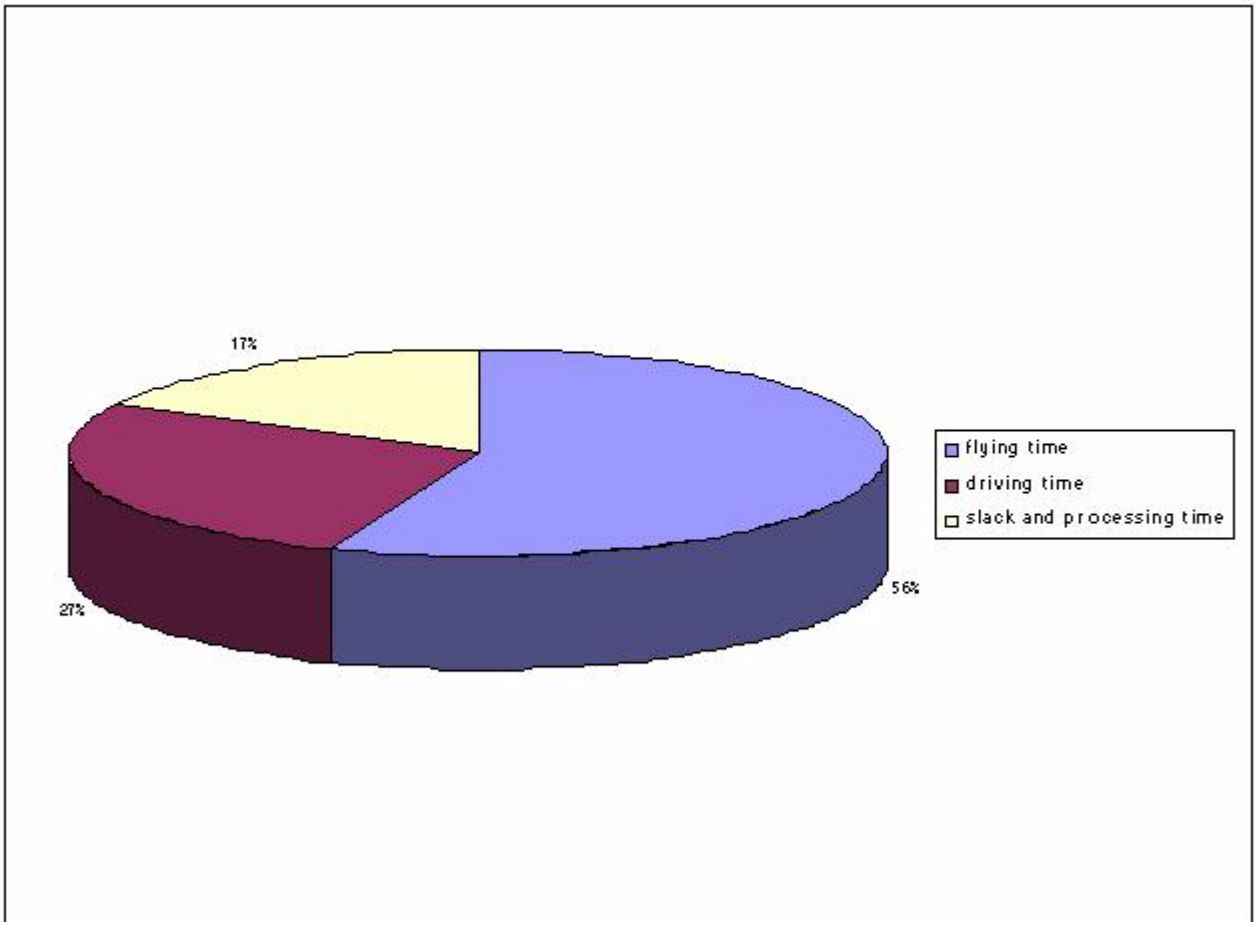


Figure 5.8 Pie Chart of Travel Times Across NAS.

Table 5.1. (1995 MSA Non-Peak Speed CDFs)

	D<250 (persons/sq mile)	250<=D<500 5(persons/sq mile)	500<=D<750 (persons/sq mile)	750<D<1000 (persons/sq mile)	D>=1000 (persons/sq mile)
Velocity (mph)	Cumulative%	Cumulative%	Cumulative%	Cumulative%	Cumulative%
0	0	0	0	0	0
10	2.14	1.92	2.35	2.59	4.59
20	20.42	18.99	22.34	22.62	26.96
30	48.42	47.72	51.53	54.84	57.17
40	71.79	72.65	76.38	78.04	78.47
50	88.39	88.75	90.45	90.87	91.34
60	97.42	97.73	97.92	97.73	98.02
70	98.43	98.68	98.66	98.66	98.84
80	99.51	99.61	99.56	99.58	99.54
90	100	99.99	100	99.99	99.99
100	100	100	100	100	100

Table 5.2. (1995 MSA Peak Speed CDF)

	D<250 (persons/sq mile)	250<=D<500 (persons/ sqmile)	500<=D<750 (persons/sq mile)	750<D<1000 (persons/sq mile)	D>=1000 (persons/sq mile)
Velocity (mph)	Cumulative%	Cumulative%	Cumulative%	Cumulative%	Cumulative%
0	0	0	0	0	0
10	1.71	1.63	2.09	2.75	4.64
20	20.20	19.80	22.46	25.25	28.99
30	48.91	49.91	53.1	57.34	60.13
40	73.05	74.54	77.05	80.98	81.33
50	88.82	89.62	90.34	92.62	92.53
60	97.69	97.86	98.00	97.88	98.30
70	98.61	98.74	98.86	98.85	99.04
80	99.65	99.60	99.61	99.54	99.63
90	100	100	99.99	100	99.99
100	100	100	100	100	100

Table 5.3. (2020 MSA Peak Speeds)

	D<250 (persons/sq mile)	250<=D<500 (persons/sq mile)	500<=D<750 (persons/sq mile)	750<D<1000 (persons/sq mile)	D>=1000 (persons/sq mile)
Velocity (mph)	Cumulative%	Cumulative%	Cumulative%	Cumulative%	Cumulative%
0	0	0	0	0	0
10	1.83	4.51	5.27	3.02	5.35
20	23.62	32.76	34.44	30.49	35.25
30	55.03	67.74	69.38	64.39	68.74
40	82.60	89.61	88.85	88.73	89.43
50	92.75	97.93	98.00	94.65	95.83
60	98.45	99.31	99.11	98.88	99.05
70	99.64	99.80	99.72	99.53	99.64
80	100	100	99.99	100	99.99
90	100	100	100	100	100
100	100	100	100	100	100

Table 5.4. (2020 MSA Non-Peak Speeds)

	D<250 (persons/sq mile)	250<=D<500 (persons/sq mile)	500<=D<750 (persons/sq mile)	750<D<1000 (persons/sq mile)	D>=1000 (persons/sq mile)
Velocity (mph)	Cumulative%	Cumulative%	Cumulative%	Cumulative%	Cumulative%
0	0	0	0	0	0
10	2.31	4.66	5.07	3.14	5.10
20	23.67	30.75	33.77	27.05	31.05
30	54.24	65.88	68.44	62.26	63.85
40	81.58	88.74	89.07	86.05	86.92
50	92.40	97.80	97.92	93.64	94.25
60	98.34	99.32	99.11	98.67	98.86
70	99.51	99.76	99.72	99.65	99.58
80	100	100	100	99.98	100
90	100	100	100	100	100
100	100	100	100	100	100

Table 5.5. (1995 and 2020 Non-MSA speeds

Velocity (mph)	Cumulative%	Cumulative%
0	0	0
10	1.43	1.55
20	14.42	16.74
30	37.88	44.37
40	65.30	77.27
50	86.15	91.64
60	97.38	98.44
70	98.58	99.61
80	99.59	99.99
90	99.99	100
100	100	100

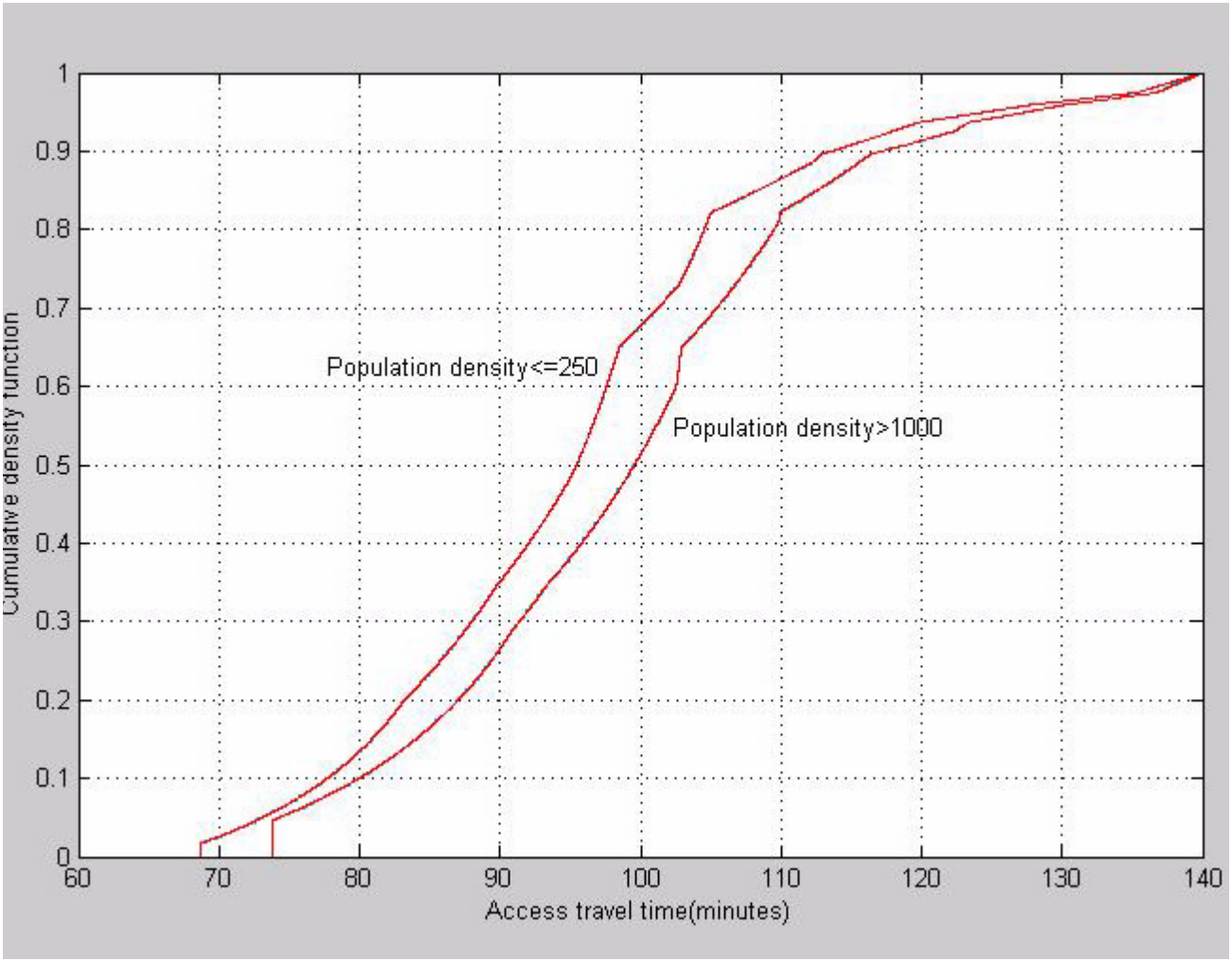


Figure 5.9 MSA Peak Access Time (Macroscopic Approach).

Figure 5.9 gives the plot for Access time CDF's to airports in MSA areas during peak hours. The CDF was obtained by convoluting the travel velocities with travel time CDFs obtained from the American Travel Survey.

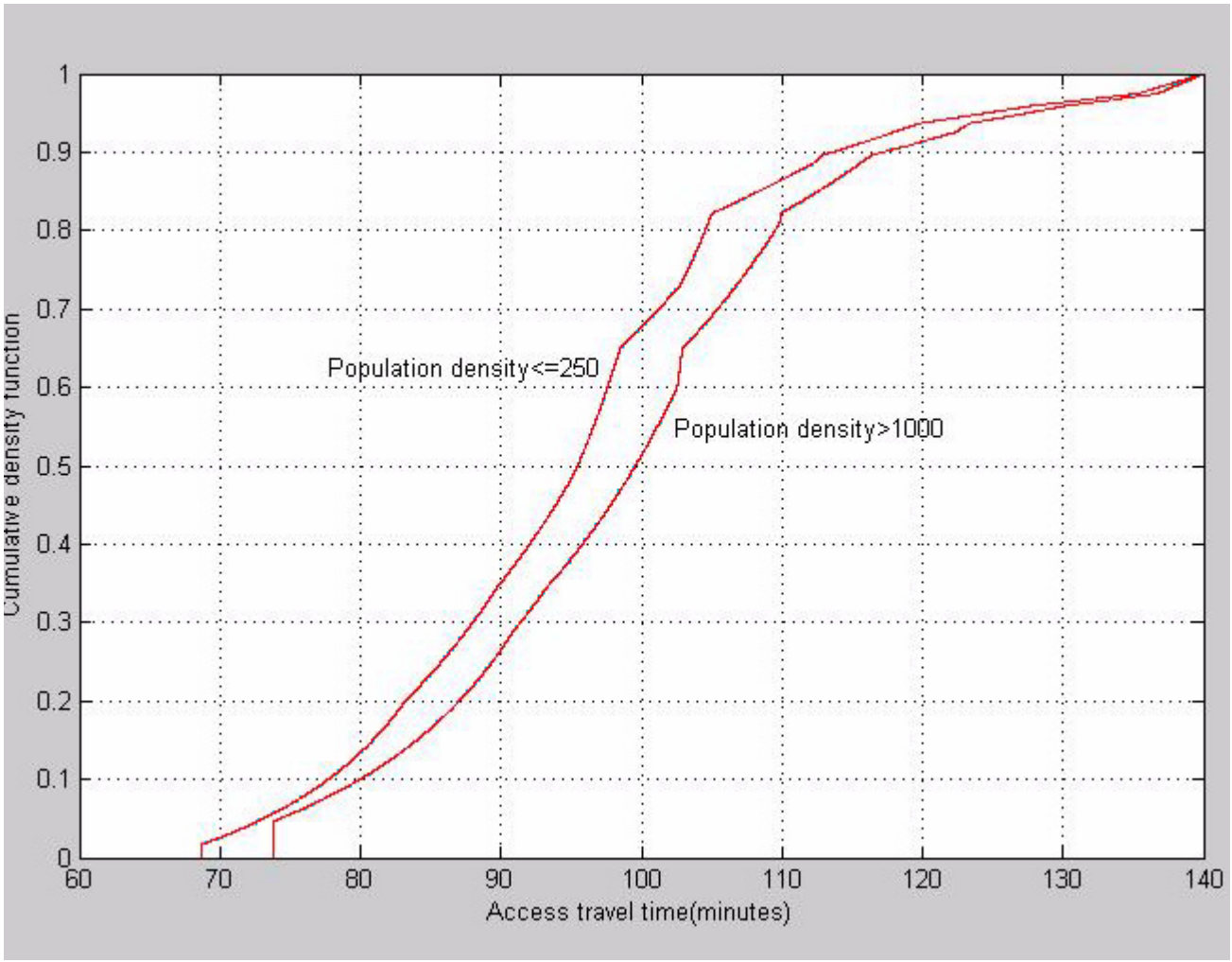


Figure 5.10 MSA Non-Peak Access Time (Macroscopic Approach).

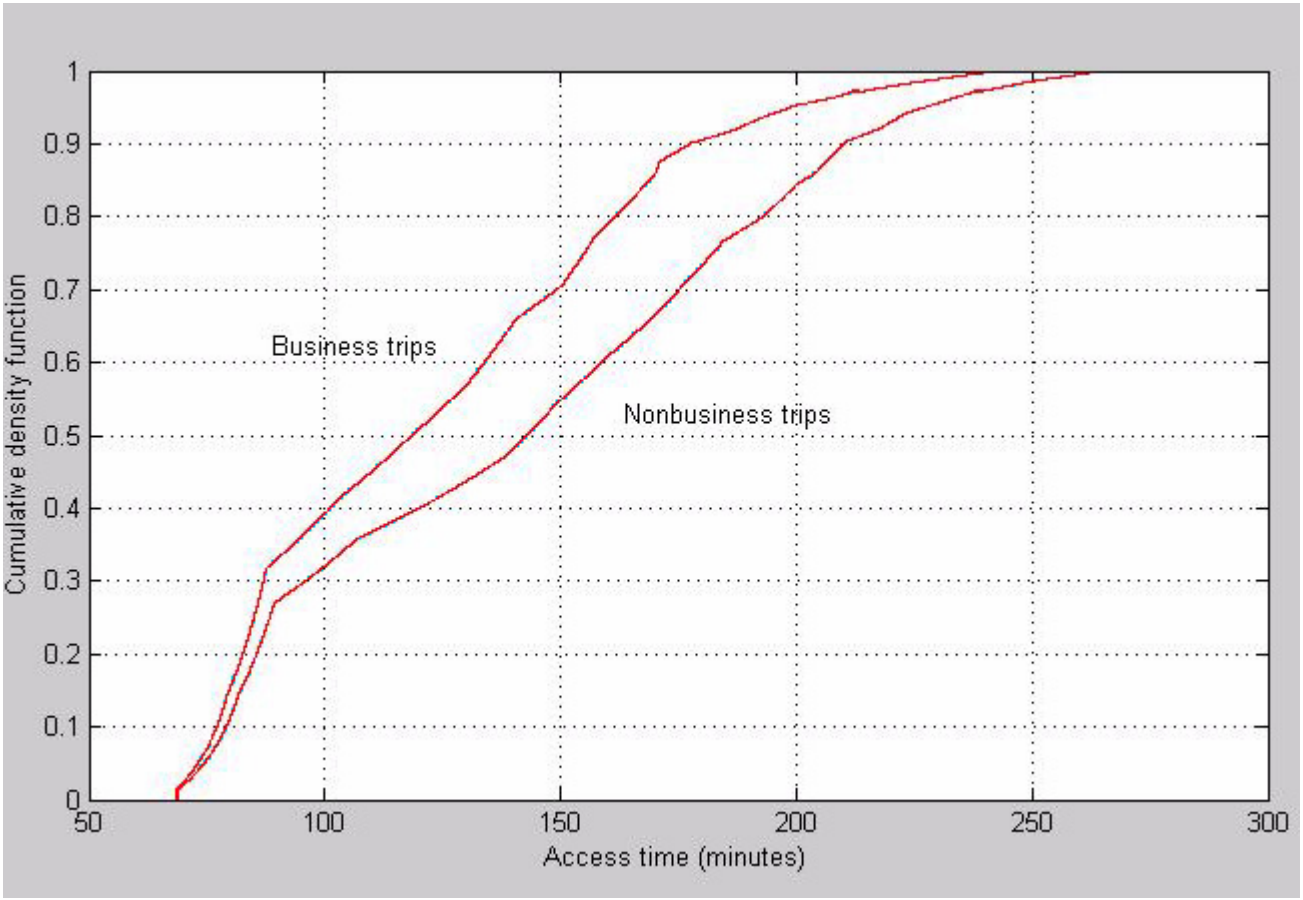


Figure 5.11 Non-MSA Access Time (Macroscopic Approach).

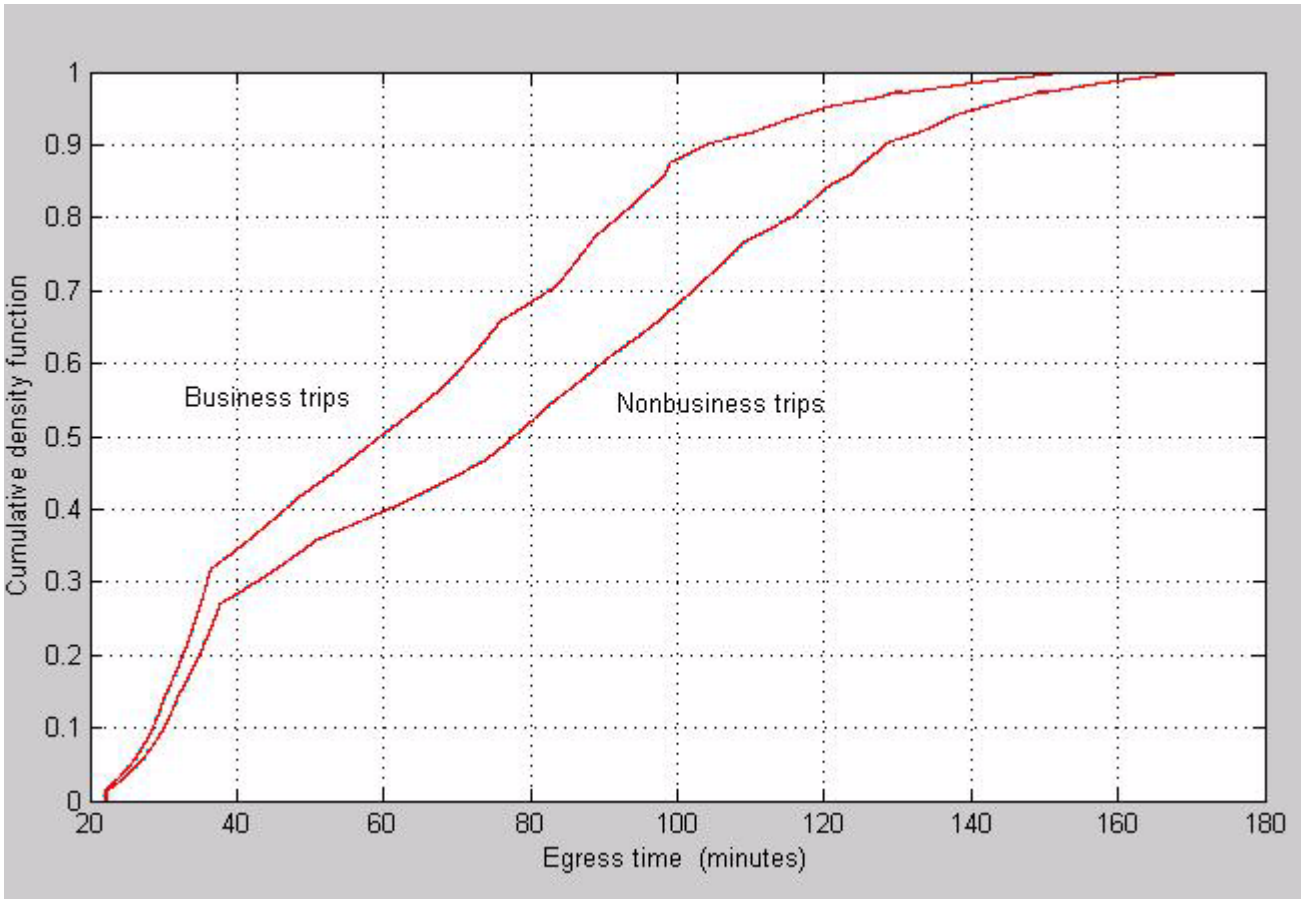


Figure 5.12 Non-MSA Egress Time(Macroscopic Approach).

Table 5.6. (MSA access times)

Time (minutes)	Access time	Access time
	Peak Cumulative%	Non-peak Cumulative%
70	2.57	2.48
80	12.33	13.42
90	32.54	33.08
100	64.58	65.58
110	85.19	85.32
120	93.06	93.02
130	96.21	96.23
140	99.21	99.21
150	100	100

Table 5.7. (MSA egress times)

Time (minutes)	Egress time	Egress time
	Peak Cumulative%	Non-peak Cumulative%
20	0	0
30	11.04	10.97
40	37.94	38.71
50	75.87	76.67
60	92.39	92.34
70	96.71	96.74
80	100	100

Table 5.8. (Non-MSA access time)

Time (minutes)	Business trips	Non-business trips
	Cumulative%	Cumulative%
65	0	0
80	15.47	11.36
100	39.21	32.06
120	50.87	40.06
140	65.64	48.26
160	79.04	60.86
180	90.56	73.38
200	95.31	84.44
230	100	95.59
250	100	100

Table 5.9. (Non-MSA egress time)

Time (minutes)	Non-Business trips	business trips
	Cumulative%	Cumulative%
20	0	0
40	28.39	34.5
60	39.75	50.33
80	52	68.4
100	68.2	88.1
120	83.85	95.1
140	94.45	100
160	100	100

CHAPTER 6

Conclusions and Recommendations

In this thesis document an approach for evaluating the mobility of the entire Air Transporation System is outlined. The purpose of this study was two-fold. One purpose was to determine the travel time of the Air Transportation System in view of the NASA plan to reduce the Air travel time across the United States by half in twenty years and to identify potential strategies to achieve this objective if it is poss ble.

The second purpose was to derive a generic value for travel time for input into another model which determines the level of utility of each travel mode.

The model is not exhaustive since it does not take into account new modes that might become available or predominant in the future. Therefore the projections for travel times may not be accurate if new modes are introduced. However the approach presented can be used to determine the utlility of the new mode. This model relies heavily on travel survey data, so inaccurate or faulty survey data may lead to wrong or inaccurate results, also sufficient survey data for any new mode developed is essential for the model to produce good results.

6.1 Conclusions

There are several conclusions that can be derived from this analysis

1) Out of the total travel time (1.6 billion hours) that are spent by air travelers about 50% of it is consumed on the ground (access and egress time, slack and processing times).

2) Commercial Aviation flights are generally longer than General Aviation flights for the same distance, on account on connecting times and schedule delay. Therefore SATS could be an attractive alternative alternative for remote airports

3) The decrease in mobility on the ground is mainly due to increase in population, as evident from the strong relationship given in Figure 4.17. and proper planning of any large city is essential for avoiding chronic congestion

4) The average speed for Non-MSA areas is found to be about 40 mph which seems to be rather low since in rural areas, traffic moves at or above the speed limit, which is 55 mph or greater for the highways, therefore for rural areas the speed limit could be taken as the travel speed

5) For MSA areas the speed limit is a bad representative for travel velocity, since the average speed sometimes is reduced to 10 mph, so for urban areas the above analysis is fairly accurate

6) The results presented assumes that no new mode would be available in the next 20 years, if a new mode comes into operation, the above results could be in error.

7) The Airline Origin and Destination Survey (DB1B) does not adequately sample small markets and the travel time by Commercial Aviation for small markets had to be derived artificially from the Official Airline Guide

8) The average travel speeds do not show much variation between various MSAs, therefore all the MSA speed data could be developed into a single CDF instead of using different CDFs for different population densities.

9) Airport Access and Egress times are much greater for Non-MSA areas than MSA areas since people from rural areas, on an average have to travel a longer distance to reach an airport.

6.2 Recommendations

1) From the magnitude of the total travel time (1.6 billion passenger hours), it is clear that NASA goal of reducing travel time in half over the next 20 years is impractical and it would an effort even to stem the deterioration in travel time over the period.

2) The mobility study for Non-MSA areas could be made more accurate by performing the analysis at the county level, rather than grouping the entire non-urban areas into a single category

3) This model assumes that no new MSA areas would come into existence over the next 20 years, the analysis could be made more refined by modeling the creation of new MSAs.

Bibiliography

1) Kisgyorgy R L, Rilett L R, “Travel time prediction by advanced neural network”, *Periodica Polytechnica Ser Civil engineering,*1,15-32 (2002).

2) Al-Deek H M, Angelo P D, Wang M C, “Travel time prediction by non-linear time series”, *ASCE Journal of Transportation engineering* Volume 122,6,440-446 (1998).

3) Van lint J W C, Hoogendoorn S P, “ Robust and Adaptive travel time prediction with neural networks, *Trail Research School Delft,* (2000).

4) Zhang X, Rice J A, “Short-term travel time prediction using a time-varying coefficient linear model”, *Journal of the American Statistical Association,*829-836, (1979).

5) _____, Bureau of Transportation Statistics, “Nationwide Personal Transporation Survey”, *http://transtats.bts.gov,* (1995).

6) _____, Bureau of Transportation Statistics, “American Travel Survey”, *http://transtats.bts.gov,* (1995).

7) _____, Bureau of Transportation Statistics, “Airline Origin and Destination Survey”, *http://transtats.bts.gov,* (2000).

8) _____, Bureau of Transportation Statistics, “Official Airline Guide”, *http://transtats.bts.gov,* (2000).

9) _____, Bureau of Transportation Statistics, “Highway Performance Monitoring System” , *http://transtats.bts.gov,* (1995).

10) _____, Texas Transportation Institute, “Urban Mobility Study” , *http://mobility.tamu.edu,* (2002).

11) _____, Eurocontrol Experimental Center R and D, “Base of Aircraft Data v 3.0”, *http://www.eurocontrol.fr/projects/bada* , (1998).

12) _____, MITRE Center for Advanced Aviation System Design, “User Request Evaluation Tool”, *http://www.mitrecaasd.org/proj/uret*

13) _____, Woods and Poole Inc, “Woods and Poole Economics” , *http://www.woodsandpoole.com/index,php,* (2000).

14) _____, Census Bureau , “American Factfinder” , *http://factfinder.census.gov,* (2000).

15) _____, Federal Aviation Administration, “Controller Pilot Datalink Communication” , *http://ffp1.faa.gov/public/tools/tools_cpdlc.asp*

16) _____, Air Transportation System Laboratory, “Aircraft performance” , *http://128.173.204.63/courses/cee5614/cee5614_pub.*

17) Ashiabor Senanu, Baik Hojong, *Transportation Systems Baseline Assessment,* (2000)

18) Garber Nicholas J, Hoel Lester A, “Traffic and Highway Engineering”, *Brooks and Cole Publishers*, (2002).

Appendix A

A.1 Matlab source code

In this section the Matlab source codes for the flight trajectory and network analysis model is presented. The flight trajectory model calculates the travel time and fuel conusmed for any aircraft flying a great circle route. The inputs required are the BADA PTF (11) files and the output is an array containinf the fuel consumed and time taken. The flight trajectory has about 15 files. however all the m-files are not presented, since some of them are only modifications of others. The input for the source code is an O-D file containing the origin and destination airport and the number of operations between each O-D pair. The output can be either a binary file containing the travel time and fuel consumed for each O-D pair or a single number for the entire O-D matrix. The function calls are described in detail below.

A.1.1 main_flight.m- This is the main m-file of the flight trajectory model which calls all the other m-files and calculates the fuel and time consumed for the case when the great circle distance between them less than the maximum range capability of the aircraft.

% Coded by Anand Seshadri

```
% Calls
% ParserforPTF_all1
% max_range
% Calculates the time and fuel for any O-D pair and stores it in a Fp array
clear all;
ParserforLMI;
max_range;
[row1,col1]=size(arr_lon1);

i=1;
var=1;
begin=1;
initialize=1;
for initialize=1:5000          % Fp array is initialized to manage memory more efficiently
   Fp(initialize).fname='';
   Fp(initialize).fmodel='';
   Fp(initialize).origin='';
   Fp(initialize).destination='';
   Fp(initialize).n='';
   Fp(initialize).wp='';
   Fp(initialize).twp='';
end
for var=1:size(arr_lon1,1)
    clear b e* h* i* j* k* work_lat work_lon work_dest lat lon loop lat2 lon2 n* p* q* s* C* D* distance1 distance2 distance3 distance5 distance6 distance7 distgc diststp fuel_consumed1 fuel_consumed2;
  clear fuel_consumed5 fuel_consumed6 fuel_consumed7 fuel_and_time fuel_optimize fn fn1 fn2 fp1 climb_velocity climb_velocity1 col count cruise_distance cruise_time cruise_velocity cruisefuel;
  clear FL FL1 FL2 FL3 FL4 FL5 FL6 FL7 time time1 time2 time5 time6 time7 time_optimize timestp maxalt mat maxheight m req_distance row rate_of_climb1 timestp distgc1;
   % clears all the variables used in each iteration
lat1 = arr_lat1(var);
lon1 = arr_lon1(var);
```

```
lat2 = arr_lat2(var);
lon2 = arr_lon2(var);
distgc = deg2nm(distance(lat1,lon1,lat2,lon2));
if(distgc>=2000) % separates medium and long-range aircrafts to aviod problems when the flight takes the place over the ocean
  main_aircraft=aircraft2;
  maximum_range=maximum_range2;
else
   main_aircraft=aircraft1;
   maximum_range=maximum_range1;
end
ParserforPTF_all1;
distance1 = zeros(1,1);
diststp = zeros(1,1);
timestp = zeros(1,1);
fuel_consumed = zeros(1,1);
time(1)=0;
rate_of_climb(1)=0;
climb_velocity(1)=0;
n=0;
[a,b]=size(FL);
for c=1:b;
   if(CLIMB_ROC_HI(c)==0) n=c;
    break
 else
    n=b;
   end
end
[lat,lon]=gcwaypts(lat1,lon1,lat2,lon2,40);
mat = gcwaypts(lat1,lon1,lat2,lon2,40);
```

```
maxheight = min(FL(n),maxht/100);
maxalt = [0:10:maxheight];
[e,f] = size(maxalt);
distance7=zeros(1,1);
FL5 = zeros(1,1);
req_distance = zeros(1,1);
for x=1:f
  h=maxalt(x);
   climb1;
   descent1;
  req_distance(x)=max(distance7);
end
if (rand_no==0)
   rand_no=1;
end
if (distgc<=maximum_range(rand_no)) % Checking if the great-circle distance is less than maximum range of the aircraft
%plot(maxalt,req_distance);xlabel('Altitude in 100s of feet');ylabel('Minimum distance required to to reach the height');
%pause;
if (distgc<=max(req_distance))
   h = interp1(req_distance,maxalt,distgc);
if (h>=30)
      optimize;
 %     plot(fuel_optimize,hvary);xlabel('Fuel consumption');ylabel('Cruising height');
 %     pause;
 %     plot(time_optimize,hvary);xlabel('Time(mins)');ylabel('Cruising height');
 %     pause;
 %     plot(fuel_optimize,hvary);xlabel('Fuel consumption');ylabel('Cruising height');
      h=zeros(1,1);
```

```
      h=hmin;

distance3=zeros(1,1);distance2=zeros(1,1);distance1=zeros(1,1);fuel_consumed=zeros(1,1);fuel_con
sumed1=zeros(1,1);fuel_consumed2=zeros(1,1);
      time2=zeros(1,1);time1=zeros(1,1);time=zeros(1,1);
      FL3=zeros(1,1);FL2=zeros(1,1);FL1=zeros(1,1);
    if(h<maxheight)
      h=hmin;
      climb;
      cruise;
      descent;
    elseif(h==maxheight)
      climb;
      descent;
   end
elseif(h<300)
   climb;
   descent;
end
%plot(distance3,fuel_consumed2);xlabel('Distance(nautical miles)');ylabel('Fuel Consumed(kgs)');
%pause;
%plot(distance3,time2);xlabel('Distance(nautical miles)');ylabel('Time(Minutes)');
%pause;
%plot(distance3,FL3);xlabel('Distance(nautical miles)');ylabel('Altitude(100s feet)');
%pause;
elseif(distgc>max(req_distance))
      h = maxheight;
      optimize;
 %      plot(fuel_optimize,hvary);xlabel('Fuel consumption');ylabel('Cruising height');
 %      pause;
```

```
%      plot(time_optimize,hvary);xlabel('Time(mins)');ylabel('Cruising height');
%      pause;
      h=zeros(1,1);
      h=hmin;

distance3=zeros(1,1);distance2=zeros(1,1);distance1=zeros(1,1);fuel_consumed=zeros(1,1);fuel_con
sumed1=zeros(1,1);fuel_consumed2=zeros(1,1);
      time2=zeros(1,1);time1=zeros(1,1);time=zeros(1,1);
      FL3=zeros(1,1);FL2=zeros(1,1);FL1=zeros(1,1);
   climb;
   cruise;
   descent;
%   plot(distance3,fuel_consumed2);xlabel('Distance(nautical miles)');ylabel('Fuel Consumed(kgs)');
%pause;
%plot(distance3,time2);xlabel('Distance(nautical miles)');ylabel('Time(Minutes)');
%pause;
%plot(distance3,FL3);xlabel('Distance(nautical miles)');ylabel('Altitude(100s feet)');
%pause;
end
[m,n]=size(mat);
i=1;
for i=1:m
   diststp(i)=deg2nm(distance(lat1,lon1,lat(i),lon(i)));
        timestp(i)=time_initial_hour(var)*60+time_initial_minutes(var)+interp1(distance3,time2,dist-
stp(i));
end
timestp(m)=time_initial_hour(var)*60+time_initial_minutes(var)+max(time2);
[row,col]=size(mat);
loop=1;
for loop=1:row
   tempdistance=deg2nm(distance(mat(1,1),mat(1,2),mat(loop,1),mat(loop,2)));
```

```
    mat(loop,3)=interp1(distance3,FL3,tempdistance);
end
mat(row,3)=0;
Fp(begin).fname = 'SATS';
Fp(begin).fmodel = fn1;
Fp(begin).origin = (origin(:,var))';
Fp(begin).destination = (destination(:,var))';
Fp(begin).n=m;
Fp(begin).wp = mat;
Fp(begin).twp = timestp';
var
begin=begin+1;
else
    prev_timestp=time_initial_hour(var)*60+time_initial_minutes(var);
    sub_flight_expt;
    var
    begin=begin+1;
end
end
```

A.1.2) sub_flight.m

This program calculates the fuel consumed and time taken when the great circle distance between origin and destination is greater than the maximum range capability of the aircraft. The intermediate airport is found based on the nearest distance criterion.

```
% Coded by Anand seshadri
% Determines the intermediate airport when the distance is greater than the
% maximum range.
% Uses a quasi-binary search to perform a more efficient search
max_distance = maximum_range(rand_no);
    no_of_trips = (distgc/max_distance);
   if (no_of_trips-round(no_of_trips)<0.4) %Calculates the no of trips required to complete the journey
```

```
    final=round(no_of_trips)-1;
  else
    final = round(no_of_trips);
  end
  loop=1;
  for loop=1:41
    distgc1(loop)=deg2nm(distance(lat1,lon1,lat(loop),lon(loop)));
  end
  loop=1;
  origin1 = origin(:,var);
  for subloop=1:(final)

distance3=zeros(1,1);distance2=zeros(1,1);distance1=zeros(1,1);fuel_consumed=zeros(1,1);fuel_con
sumed1=zeros(1,1);fuel_consumed2=zeros(1,1);
     time2=zeros(1,1);time1=zeros(1,1);time=zeros(1,1);
     FL3=zeros(1,1);FL2=zeros(1,1);FL1=zeros(1,1);diststp = zeros(1,1);timestp = zeros(1,1);
    temp_lat = interp1(distgc1,lat,subloop*max_distance);
    temp_lon = interp1(distgc1,lon,subloop*max_distance);
    mid=floor(dim2/2);
    start=1;
    fin=dim2;
   if ((abs(arr_lat3(mid)-temp_lat)>2)&(start>fin)) % this part has the quasi-binary search to find the
intermediate airport
      mid=(start+fin)/2;
      if (arr_lat3(mid)>temp_lat)
        fin=mid-1;
      elseif(arr_lat3(mid)<=temp_lat)
        start=mid+1;
      end
    end
    fix=mid;
```

```
    fix1=max(mid-100,1);
    fix2=min(mid+100,dim2);
    work_lat=arr_lat3(fix1:fix2);
    work_lon=arr_lon3(fix1:fix2);
    work_dest=way_destination(:,fix1:fix2);
    for check=1:200
       if (work_lat(check)==lat1)&(work_lon(check)==lon1)
          work_lat(check)='';
          work_lon(check)='';
          work_dest(:,check)='';
       end
    end
    for loop=1:(fix2-fix1);
       diff_dist(loop)=deg2nm(distance(temp_lat,temp_lon,work_lat(loop),work_lon(loop)));
       if (deg2nm(distance(lat1,lon1,work_lat(loop),work_lon(loop)))>max_distance)
          diff_dist(loop)=10000;
       end
    end
    [min_dist,min_index]= min(diff_dist);
    detour = work_dest(:,min_index);
    lat2 = work_lat(min_index);
    lon2 = work_lon(min_index);
    mat = gcwaypts(lat1,lon1,lat2,lon2,40);
    distgc = deg2nm(distance(lat1,lon1,lat2,lon2,40));
h = interp1(req_distance,maxalt,distgc);
optimize;
     h=zeros(1,1);
     h=hmin;
climb;
cruise;
```

```
   descent;
[m,n]=size(mat);
i=1;
for i=1:m
   diststp(i)=deg2nm(distance(lat1,lon1,mat(i,1),mat(i,2)));
   timestp(i)=max(prev_timestp)+interp1(distance3,time2,diststp(i));
end
timestp(m)=max(prev_timestp)+max(time2);
[row,col]=size(mat);
loop=1;
prev_timestp=timestp;
for loop=1:row
   tempdistance=deg2nm(distance(mat(1,1),mat(1,2),mat(loop,1),mat(loop,2)));
   mat(loop,3)=interp1(distance3,FL3,tempdistance);
end
mat(row,3)=0;
Fp(begin).fname = 'SATS';
Fp(begin).fmodel = fn1;
Fp(begin).origin = (origin1)';
Fp(begin).destination = detour';
Fp(begin).n=m;
Fp(begin).wp = mat;
Fp(begin).twp = timestp';
origin1=detour;
lat1 = lat2;
lon1 = lon2;
begin=begin+1;
end
%end
%end
```

```
if(final>0)
lat1=lat2;
lon1=lon2;
end
if(deg2nm(distance(lat1,lon1,arr_lat2(var),arr_lon2(var)))<=max_distance)
destination1 = origin(:,var);
lat2 = arr_lat2(var);
lon2 = arr_lon2(var);
[mat]=gcwaypts(lat1,lon1,lat2,lon2);
distgc = deg2nm(distance(lat1,lon1,lat2,lon2));
 h = interp1(req_distance,maxalt,distgc);
 optimize;
 h=zeros(1,1);
       h=hmin;

distance3=zeros(1,1);distance2=zeros(1,1);distance1=zeros(1,1);fuel_consumed=zeros(1,1);fuel_con
sumed1=zeros(1,1);fuel_consumed2=zeros(1,1);
       time2=zeros(1,1);time1=zeros(1,1);time=zeros(1,1);
       FL3=zeros(1,1);FL2=zeros(1,1);FL1=zeros(1,1);diststp=zeros(1,1);timestp = zeros(1,1);


    climb;
    cruise;
    descent;
[m,n]=size(mat);
i=1;
for i=1:m
    diststp(i)=deg2nm(distance(lat1,lon1,mat(i,1),mat(i,2)));
   timestp(i)=max(prev_timestp)+interp1(distance3,time2,diststp(i));
end
initial_distance = max(distance3);
```

```
timestp(m)=max(prev_timestp)+max(time2);
prev_timestp = timestp;
[row,col]=size(mat);
loop=1;
for loop=1:row
   tempdistance=deg2nm(distance(mat(1,1),mat(1,2),mat(loop,1),mat(loop,2)));
   mat(loop,3)=interp1(distance3,FL3,tempdistance);
end
mat(row,3)=0;
Fp(begin).fname = 'SATS';
Fp(begin).fmodel = fn1;
Fp(begin).origin = (origin1)';
Fp(begin).destination = destination(:,var)';
Fp(begin).n=m;
Fp(begin).wp = mat;
Fp(begin).twp = timestp';
else
                max_distance = min(deg2nm(distance(lat1,lon1,arr_lat2(var),arr_lon2(var)))/
1.3,maximum_range(rand_no));
   subloop=final+1;
while (deg2nm(distance(lat1,lon1,arr_lat2(var),arr_lon2(var)))>max_distance)

distance3=zeros(1,1);distance2=zeros(1,1);distance1=zeros(1,1);fuel_consumed=zeros(1,1);fuel_con
sumed1=zeros(1,1);fuel_consumed2=zeros(1,1);
      time2=zeros(1,1);time1=zeros(1,1);time=zeros(1,1);
      FL3=zeros(1,1);FL2=zeros(1,1);FL1=zeros(1,1);diststp = zeros(1,1);timestp = zeros(1,1);
      [mat]=gcwaypts(lat1,lon1,arr_lat2(var),arr_lon2(var),40);
      [lat,lon]=gcwaypts(lat1,lon1,arr_lat2(var),arr_lon2(var),40);
      loop=1;
      for loop=1:21;
         distgc1(loop)=deg2nm(distance(mat(1,1),mat(1,2),mat(loop,1),mat(loop,2)));
```

```
 end
temp_lat = interp1(distgc1,lat,max_distance);
temp_lon = interp1(distgc1,lon,max_distance);
mid=floor(dim2/2);
start=1;
fin=dim2;
while((abs(arr_lat3(mid)-temp_lat)>2)&(start>fin))
   mid=(start+fin)/2;
   if (arr_lat3(mid)>temp_lat)
      fin=mid-1;
   elseif(arr_lat3(mid)<=temp_lat)
      start=mid+1;
   end
end
fix1=max(mid-100,1);
fix2=min(mid+100,dim2);
work_lat=arr_lat3(fix1:fix2);
work_lon=arr_lon3(fix1:fix2);
work_dest=way_destination(:,fix1:fix2);
for check=1:200
   if (work_lat(check)==lat1)&(work_lon(check)==lon1)
      work_lat(check)='';
      work_lon(check)='';
      work_dest(:,check)='';
   end
end
 for loop=1:(fix2-fix1);
 diff_dist(loop)=deg2nm(distance(temp_lat,temp_lon,work_lat(loop),work_lon(loop)));
   if (deg2nm(distance(lat1,lon1,work_lat(loop),work_lon(loop)))>max_distance)
      diff_dist(loop)=10000;
```

```
        end
    end
    [min_dist,min_index]= min(diff_dist);
    detour = work_dest(:,min_index);
    lat2 = work_lat(min_index);
    lon2 = work_lon(min_index);
    mat = gcwaypts(lat1,lon1,lat2,lon2,40);
    distgc = deg2nm(distance(lat1,lon1,lat2,lon2));
    h = interp1(req_distance,maxalt,distgc);
    optimize;
     h=zeros(1,1);
     h=hmin;
  climb;
  cruise;
  descent;
[m,n]=size(mat);
i=1;
for i=1:m
  diststp(i)=deg2nm(distance(lat1,lon1,mat(i,1),mat(i,2)));
  timestp(i)=max(prev_timestp)+interp1(distance3,time2,diststp(i));
end
timestp(m)=max(prev_timestp)+max(time2);
[row,col]=size(mat);
loop=1;
prev_timestp=timestp;
for loop=1:row
  tempdistance=deg2nm(distance(mat(1,1),mat(1,2),mat(loop,1),mat(loop,2)));
  mat(loop,3)=interp1(distance3,FL3,tempdistance);
end
mat(row,3)=0;
```

```
Fp(begin).fname = 'SATS';
Fp(begin).fmodel = fn1;
Fp(begin).origin = (origin1)';
Fp(begin).destination = detour';
Fp(begin).n=m;
Fp(begin).wp = mat;
Fp(begin).twp = timestp';
origin1=detour;
lat1 = lat2;
lon1 = lon2;
begin=begin+1;
end
destination1 = origin(:,var);
lat2 = arr_lat2(var);
lon2 = arr_lon2(var);
[mat]=gcwaypts(lat1,lon1,lat2,lon2);
distgc = deg2nm(distance(lat1,lon1,lat2,lon2));
 h = interp1(req_distance,maxalt,distgc);
 optimize;
 h=zeros(1,1);
       h=hmin;

distance3=zeros(1,1);distance2=zeros(1,1);distance1=zeros(1,1);fuel_consumed=zeros(1,1);fuel_con
sumed1=zeros(1,1);fuel_consumed2=zeros(1,1);
       time2=zeros(1,1);time1=zeros(1,1);time=zeros(1,1);
       FL3=zeros(1,1);FL2=zeros(1,1);FL1=zeros(1,1);diststp=zeros(1,1);timestp = zeros(1,1);


    climb;
    cruise;
    descent;
```

```
[m,n]=size(mat);
i=1;
for i=1:m
    diststp(i)=deg2nm(distance(lat1,lon1,mat(i,1),mat(i,2)));
   timestp(i)=max(prev_timestp)+interp1(distance3,time2,diststp(i));
end
initial_distance = max(distance3);
timestp(m)=max(prev_timestp)+max(time2);
prev_timestp = timestp;
[row,col]=size(mat);
loop=1;
for loop=1:row
    tempdistance=deg2nm(distance(mat(1,1),mat(1,2),mat(loop,1),mat(loop,2)));
    mat(loop,3)=interp1(distance3,FL3,tempdistance);
end
mat(row,3)=0;
Fp(begin).fname = 'SATS';
Fp(begin).fmodel = fn1;
Fp(begin).origin = (origin1)';
Fp(begin).destination = destination(:,var)';
Fp(begin).n=m;
Fp(begin).wp = mat;
Fp(begin).twp = timestp';
end
```

A.1.3) climb.m

This program calculates the fuel consumed and time for an aircraft to reach a particular altitude in the climb mode. As mentioned this program assumes that the flight operates under ISA conditions.

```
% Coded by Anand seshadri
% calculates fuel consumed and time consumed to reach a particular height
% while in climb mode
```

```
% Assumes flight takesoff under ISA conditions
[col,row] = size(CLIMB_ROC_HI);
i=1;flag=-1;
fuel_consumed = zeros(1,1); % variable initialization.
time = zeros(1,1);
rate_of_climb = zeros(1,1);
climb_velocity = zeros(1,1);
 CLIMB_ROC_HI1(i)=zeros(1,1);
 CLIMB_TAS1(i)=zeros(1,1);
 CLIMB_FUEL_NOM1(i)=zeros(1,1);
for i=1:row
   if (FL(i)==h)
      n1=i;
      flag=0;
end
end
i=1;
if (flag==0)
   while (i<=n1);
         CLIMB_ROC_HI1(i)=CLIMB_ROC_HI(i);
         CLIMB_TAS1(i)=CLIMB_TAS(i);
         CLIMB_FUEL_NOM1(i)=CLIMB_FUEL_NOM(i);
         FL1(i)=FL(i);
         i=i+1;
      end
   else
    while(FL(i)<h)
         FL1(i)=FL(i);
         CLIMB_ROC_HI1(i)=CLIMB_ROC_HI(i);
         CLIMB_TAS1(i)=CLIMB_TAS(i);
```

```
        CLIMB_FUEL_NOM1(i)=CLIMB_FUEL_NOM(i);
        i=i+1;
   end
   FL1(i)=h;
TAS_climb1 = interp1(FL,CLIMB_TAS,h); % interpolates for intermediate altitudes
rate_climb1= interp1(FL,CLIMB_ROC_HI,h);
CLIMBFUEL1 = interp1(FL,CLIMB_FUEL_NOM,h);
rate_climb1 = interp1(FL,CLIMB_ROC_HI,h);
CLIMB_ROC_HI1(i)=rate_climb1;
CLIMB_TAS1(i)=TAS_climb1;
CLIMB_FUEL_NOM1(i)=CLIMBFUEL1;
[p,q] = size(CLIMB_ROC_HI1);
end
[p,q]=size(FL1);
i=2;
  for i=2:q
   climb_velocity = (CLIMB_TAS1(i-1)+CLIMB_TAS1(i))/2;
   rate_of_climb(i) = (CLIMB_ROC_HI1(i-1)+CLIMB_ROC_HI1(i))/2;
   time(i) = time(i-1)+(FL1(i)-FL1(i-1))/(rate_of_climb(i))*100;
   distance1(i) = distance1(i-1)+(time(i)-time(i-1))/60*climb_velocity;
    fuel_consumed(i)=fuel_consumed(i-1)+(CLIMB_FUEL_NOM1(i)+CLIMB_FUEL_NOM1(i-1))/
2*(time(i)-time(i-1));
end
```

A.1.4) descent.m

```
% Coded by Anand Seshadri
% calculates the fuel consumed and time for an aircraft to descend
[col,row] = size(CLIMB_ROC_HI);
fuel_consumed1 = zeros(1,1);% initialized all the variables
rate_of_descent= zeros(1,1);
descent_velocity = zeros(1,1);
```

```
DESCENT_FUEL1(i)=zeros(1,1);
DESCENT_ROD1(i)=zeros(1,1);
DESCENT_TAS1(i)=zeros(1,1);
time1 = zeros(1,2);
time2 = zeros(1,1);
distance3 = zeros(1,1);
fuel_consumed2 = zeros(1,1);
FL3 = zeros(1,1);
i=1;flag=-1;
FL2 = zeros(1,2);
for i=1:row
   if (FL(i)==h)
      flag=0;
   end
end
i=1;
if (flag==0)
   while(FL(i)<h)
         DESCENT_FUEL1(i)=DESCENT_FUEL(i);
         DESCENT_TAS1(i)=DESCENT_TAS(i);
         DESCENT_ROD1(i)=DESCENT_ROD(i);
      i=i+1;
   end
else
   while(FL(i)<h)
         DESCENT_FUEL1(i)=DESCENT_FUEL(i);
         DESCENT_TAS1(i)=DESCENT_TAS(i);
         DESCENT_ROD1(i)=DESCENT_ROD(i);
      i=i+1;
   end
```

```
end
rate_descent = interp1(FL,DESCENT_ROD,h); % interpolates for intermediate heights
TAS_descent = interp1(FL,DESCENT_TAS,h);
DESCENTFUEL = interp1(FL,DESCENT_FUEL,h);
FL1(i)=h;
[p,q]=size(FL1);
for k=1:q
FL2(k)= FL1(q-k+1)+0.001; % to avoid singularity in interpolation
end
DESCENT_FUEL1(i)=DESCENTFUEL;
DESCENT_TAS1(i)=TAS_descent;
DESCENT_ROD1(i)=rate_descent;
if (distgc<=max(req_distance))
time1(1)=time(q);
distance2(1)=distance1(q);
fuel_consumed1(1)=fuel_consumed(q);
i=2;
size = q;
for i=1:size-1; %calculates all the required quantities
   descent_velocity = (DESCENT_TAS1(size-i+1)+DESCENT_TAS1(size-i))/2;
   rate_of_descent(i) = (DESCENT_ROD1(size-i+1)+DESCENT_ROD1(size-i))/2;
   time1(i+1) = time1(i)+(FL1(size-i+1)-FL1(size-i))/(rate_of_descent(i))*100;
   distance2(i+1) = distance2(i)+(time1(i+1)-time1(i))/60*descent_velocity;
        fuel_consumed1(i+1)=fuel_consumed1(i)+(DESCENT_FUEL1(i+1)+DESCENT_FUEL1(i))/
2*(time1(i+1)-time1(i));
end
distance3 = [distance1 distance2]; % combines the arrays from the climb,cruise and descent phases.
time2 = [time time1];
fuel_consumed2 = [fuel_consumed fuel_consumed1];
FL3 = [FL1 FL2];
```

```
elseif(distgc>max(req_distance))
time1(1)=time(q)+cruise_time*60;
distance2(1)=distance1(q)+cruise_distance;
fuel_consumed1(1)=fuel_consumed(q)+cruisefuel;
i=2;
size = q;
for i=1:size-1;
   descent_velocity = (DESCENT_TAS1(size-i+1)+DESCENT_TAS1(size-i))/2;
   rate_of_descent(i) = (DESCENT_ROD1(size-i+1)+DESCENT_ROD1(size-i))/2;
   time1(i+1) = time1(i)+(FL1(size-i+1)-FL1(size-i))/(rate_of_descent(i))*100;
   distance2(i+1) = distance2(i)+(time1(i+1)-time1(i))/60*descent_velocity;
        fuel_consumed1(i+1)=fuel_consumed1(i)+(DESCENT_FUEL1(i+1)+DESCENT_FUEL1(i))/
2*(time1(i+1)-time1(i));
end
distance3 = [distance1 distance2];
time2 = [time  time1];
fuel_consumed2 = [fuel_consumed fuel_consumed1];
FL3 = [FL1 FL2];
end
```

A.1.5) maximum_range.m

This program calculates the maximum range of an aircraft. The code calculates the fuel required to travel a range of distances and interpolates these values with the maximum fuel that the aircraft can carry to find out the maximum range.

```
% Coded by Anand seshadri
% Calculates the maximum range of an aircraft
% Assumes ISA conditions
distance1 = zeros(1,1); % initializes the variables
time5 = zeros(1,1);
distance5 = zeros(1,1);
fuel_consumed5 = zeros(1,1);
rate_of_climb1 = zeros(1,1);
```

```
climb_velocity1 = zeros(1,1);
 CLIMB_ROC_HI1=zeros(1,1);
 CLIMB_TAS1=zeros(1,1);
 CLIMB_FUEL_NOM1=zeros(1,1);
randomize;
[size1,size2]=size(main_aircraft);
no_of_runs=1;
runtime=1;
distgc1= [0:1000:10000];
for no_of_runs=1:size2
    ParserforPTF_all2;
    n=0;
[a,b]=size(FL);
for c=1:b;
    if(CLIMB_ROC_HI(c)==0) n=c;
     break
 else
     n=b;
    end
end
maxheight = min(FL(n-1),maxht/100);
maxalt = [0:10:maxheight];
[e,f] = size(maxalt);
distance7=zeros(1,10);
FL5 = zeros(1,1);
req_distance = zeros(1,1);
for x=1:f
   h=maxalt(x);
    climb1;
    descent1;
```

```
  req_distance(x)=max(distance7);
end
   for runtime=1:8
      distgc = distgc1(runtime);
      FL1 = zeros(1,1);
      distance1 = zeros(1,1);
      distance2 = zeros(1,1);
    if (distgc<=max(req_distance))
         h = interp1(maxalt,req_distance,distgc);
         climb;
         descent2;
      else
         optimize;
         h=hmin;
         climb;
         cruise;
         descent;
      end
   maxfuel(runtime)=max(fuel_consumed2);
   maxdistance(runtime) = max(distance3);
   end
   fuelmax = 1000*(maximum_weight-0.7*maximum_payload-empty_weight);
   maximum_range(no_of_runs)=interp1(maxfuel,maxdistance,fuelmax);
end
runtime=1;
i=1;j=1;n=1;
for runtime=1:size2
   if (maximum_range(runtime)<=2500) % separates medium and long range aircraft
      maximum_range1(j)=maximum_range(runtime);
      if (main_aircraft(4,runtime)==')
```

```
        for n=1:3
            aircraft1(n,j)=main_aircraft(n,runtime);
        end
    else
        for n=1:4
            aircraft1(n,j)=main_aircraft(n,runtime);
        end
    end
    j=j+1;
  else
    maximum_range2(i)=maximum_range(runtime);
    if (main_aircraft(4,runtime)==[])
        for n=1:3
            aircraft2(n,i)=main_aircraft(n,runtime);
        end
    else
        for n=1:4
            aircraft2(n,i)=main_aircraft(n,runtime);
        end
    end
    i=i+1;
  end
end
```

A.1.6) Optimize.m

This program calculates the optimum height at which a composite index consisting of a combination of fuel and time is minimized.

```
% Coded by Anand Seshadri
% Finds out the optimum height for cruise
hvary = [30:10:maxheight-10];
[r,c] = size(hvary);
```

```
time_optimize = zeros(1,1);% initialized the variables
fuel_optimize = zeros(1,1);
fuel_and_time = zeros(1,1);
loop=1;
high = c;
for loop=1:high
   h = hvary(loop);
   climb;
   cruise;
   descent;
   fuel_optimize(loop)=max(fuel_consumed2);
   time_optimize(loop)=max(time2);
end
h=hvary(c);
climb;
descent;
fuel_optimize(c)=max(fuel_consumed2);
time_optimize(c)=max(time2);
loop=1;
min_fuel = min(fuel_optimize);
min_time = min(time_optimize);
for loop=1:c
                fuel_and_time(loop)=(fuel_optimize(loop)-min_fuel)/(min_fuel)+(time_optimize(loop)-
min_time)/(min_time);
   % calculates the composite index
end
minfuel_and_time = min(fuel_and_time);
hmin=interp1(fuel_and_time,hvary,minfuel_and_time); % searches for the optimal height.
```

A.1.7) Randomize.m

This program selects flights at random from the BADA database and injects them into the main model

```
for flight trajectory simulation.
% Coded by Anand seshadri
% Randomly selects flights for trajectory generation
filename = 'random.txt';% reads the file containing aircraft names
fp1 = fopen(filename,'r');
arr=1;
i=1;
j=1;
while(feof(fp1)==0)
   line = fgetl(fp1);
   strlength = length (line);
   for i=1:4
   main_aircraft(i,j) = (line(i:i));
   end
   i=1;
   j=j+1;
end
fclose(fp1);
[m,n]=size(main_aircraft);
j=round(rand*n);
rand_no=j;
if(j==0)
   j=1;
end
i=1;
fn1 =[];
if (main_aircraft(4,j)==' ')
   for i=1:3
   fn1 = [fn1 main_aircraft(i,j)];
   end
```

```
   fn2 = '___.PTF';
else
   for i=1:4
   fn1 = [fn1 main_aircraft(i,j)];
   end
   fn2 = '__.PTF';
end
fn = [fn1 fn2];
```

A.1.8) ParserforPTF_all.m

This code reads all the all the various climb and fuel consumption rates from the BADA PTF file and loads it into the memory.

```
% parser for *.ptf file (BADA)
% reading for different Flight Levels (FL) values TAS [knots], and Rate of Climb/Descent [fpm] for
% two profiles CLIMB and DESCENT
% input data are file name (extension is ptf)
% output data are FL, CLIMB_TAS, CLIMB_ROC (nominal value), DESCENT_TAS, DESCENT_ROD

% Coded by  Panta Lucic (December 2000)
% Modified by A.Trani (March 2001) to extract all values of the BADA PFT file
%Added an If statement to extract climb values
fn1 = [];
 fn = [];
 if (main_aircraft(4,no_of_runs)==' ')
   for i=1:3
   fn1 = [fn1 main_aircraft(i,no_of_runs)];
   end
   fn2 = '___.PTF';
else
   for i=1:4
```

```
    fn1 = [fn1 main_aircraft(i,no_of_runs)];
    end
    fn2 = '__.PTF';
end
fn = [fn1 fn2];
%fn = input('Enter the aircraft to be flown followed by __.PTF');
fp1 = fopen(fn,'r');

count = 0;

while(feof(fp1)==0)
  line = fgetl(fp1);
  strlength = length (line);
  if (strlength==77)
        maxht = str2num(line(72:77));  %extracts the maximum height
  end
  if strlength > 3
    inNum = str2num(line(1:3));
    inNum1 = str2num(line(36:38));
  else
    inNum = [];
    inNum1= [];
  end
  if isempty(inNum1) == 0% Checks for an empty value (1 = empty)]
     count = count + 1;         % counter to increment value of the output array
    FL(count) = inNum;% Extracts flight level
 CLIMB_TAS(count) = str2num(line(36:38));% True Airspeed in climb (knots)
    CLIMB_ROC_LO(count) = str2num(line(42:45));
    CLIMB_ROC_NOM(count) = str2num(line(47:50));
 CLIMB_ROC_HI(count) = str2num(line(52:55));
```

```
  CLIMB_FUEL_NOM(count) = str2num(line(59:63));
     DESCENT_TAS(count) = str2num(line(69:71));
     DESCENT_ROD(count) = str2num(line(74:77));
  DESCENT_FUEL(count) = str2num(line(82:85));

     % This part reads the bada code for cruise which starts above FL 20

  if FL(count) > 20
  CRUISE_TAS(count) = str2num(line(8:10));
  CRUISE_FUEL_LO(count) = str2num(line(14:18));
  CRUISE_FUEL_NOM(count) = str2num(line(20:24));
  CRUISE_FUEL_HI(count) = str2num(line(26:30));
  end
   end
end %while(feof(fp1)==0)
fclose(fp1);
clear fn1,fn2;
     fn1 = [];
     fn2 =[];
if (main_aircraft(4,no_of_runs)==' ')
    for i=1:3
    fn1 = [fn1 main_aircraft(i,no_of_runs)];
    end
    fn2 = '___.OPF';
    else
    for i=1:4
    fn1 = [fn1 main_aircraft(i,no_of_runs)];
    end
    fn2 = '__.OPF';
    end
```

```
    fnptf = [fn1 fn2];
     fp4 = fopen(fnptf,'r');
while(feof(fp4)==0)
    line = fgetl(fp4);
    if (line(47:57)=='max payload')
        nextline = fgetl(fp4);
     maximum_payload = str2num(nextline(47:56));
     maximum_weight = str2num(nextline(34:43));
     empty_weight = str2num(nextline(21:30));
    end
end
fclose(fp4);
```

A.1.9) ParserforLMI.m

This file reads the GA OD table from LMI and loads it into the memory.

```
% Coded by Anand Seshadri
% reads the LMI O-D Table
fn1 = 'sorted_jet.dat'
fp2 = fopen(fn1,'r');
i=0;
while(feof(fp2)==0)% Checks for end-of-file
    i=i+1;
    line = fgetl(fp2);
    strlength = length (line);
    if (strlength==0) break
    end
    arr_lat1(i)= str2num(line(17:23));
    arr_lon1(i)= str2num(line(27:34));
    arr_lat2(i)= str2num(line(54:60));
    arr_lon2(i)= str2num(line(64:71));
    dist_between(i)=str2num(line(77:81));
```

```
    j=1;
    for j=1:3
    origin(j,i)=line(10+j:10+j);
    end
    j=1;
    for j=1:3
        destination(j,i)=line(47+j:47+j);
    end
   time_initial_hour(i)=str2num(line(87:90));
   time_initial_minutes(i)=str2num(line(92:93));
end
fclose(fp2);
way_destination=destination;
arr_lat3=arr_lat2;
arr_lon3=arr_lon2;
[dim1,dim2]=size(arr_lat2);
lop=1;
j=1;
while (lop<=dim2)
    j=lop+1;
    while(j<=dim2)
        if (way_destination(:,lop)==way_destination(:,j))
            way_destination(:,j)='';
            arr_lat3(j)='';
            arr_lon3(j)='';
        end
        [dim1,dim2]=size(way_destination);
        j=j+1;
    end
    lop=lop+1;
```

```
end
lop=1;
while (lop<=dim2)
   j=lop+1;
   while(j<=dim2)
      if (way_destination(:,lop)==way_destination(:,j))
         way_destination(:,j)='';
         arr_lat3(j)='';
         arr_lon3(j)='';
      end
      [dim1,dim2]=size(way_destination);
      j=j+1;
   end
   lop=lop+1;
end
lop=1;
run=1;
lop=2;
val=0;
str='XXX';
incr=floor(dim2/2);
while(incr>0)
    for lop=(incr+1):dim2
       j=lop-incr;
       while(j>0)
         if (arr_lat3(j)>arr_lon3(j))
            str=way_destination(:,j);
            way_destination(:,j)=way_destination(:,j+incr);
            way_destination(:,j+incr)=str;
            temp=arr_lat3(j);
```

```
            arr_lat3(j)=arr_lat3(j+incr);
            arr_lat3(j+incr)=temp;
            temp=arr_lon3(j);
            arr_lon3(j)=arr_lon3(j+incr);
            arr_lon3(j+incr)=temp;
            j=j-incr;
          else
            j=0;
          end
        end
    end
incr=floor(incr/2);
end
```

A.2) Network analysis codes

A.2.1) traveltime.m

```
% Coded by Anand Seshadri

% This code calculates the travel time CDF's for the MSA and the non-MSA

% regions using both the macroscopic and microscopic analysis using the

% NPTS and ATS data

% this part is for the MSA data

 load data_1995_msa.mat;

j=1;l=1;m=1;n=1;u=1;

 for i=1:347958

                                                                                if
(data_1995_msa.speed(i)<=100)&(data_1995_msa.tripmiles(i)<=100)&(data_1995_msa.tripmiles(i)
~=99997)&(data_1995_msa.tripmiles(i)~=99998)&(data_1995_msa.tripmiles(i)~=99999)&(data_19
95_msa.travelminutes(i)~=9996)&(data_1995_msa.travelminutes(i)~=9997)&(data_1995_msa.trave
```

```
lminutes(i)~=9998)&(data_1995_msa.travelminutes(i)~=9999)&(data_1995_msa.tripmiles(i)>2)
    if data_1995_msa.peaktrip(i)==0
    if data_1995_msa.popdensity(i)<=250
        speed_nonpeak_1995_1(j)=data_1995_msa.speed(i);
        j=j+1;
  elseif (250<data_1995_msa.popdensity(i))&(data_1995_msa.popdensity(i)<=500)
        speed_nonpeak_1995_3(l)=data_1995_msa.speed(i);
        l=l+1;
   elseif (500<data_1995_msa.popdensity(i))&(data_1995_msa.popdensity(i)<=750)
        speed_nonpeak_1995_4(m)=data_1995_msa.speed(i);
        m=m+1;
   elseif (750<data_1995_msa.popdensity(i))&(data_1995_msa.popdensity(i)<=1000)
        speed_nonpeak_1995_5(n)=data_1995_msa.speed(i);
        n=n+1;
    elseif (data_1995_msa.popdensity(i)>=1000)
        speed_nonpeak_1995_6(u)=data_1995_msa.speed(i);
        u=u+1;
    end
 end
end
end
x=[5:10:95];
```

```
N=hist(speed_nonpeak_1995_4,x);
x1=[0:10:100];
N=N/sum(N);
cdf_speed=zeros(1,11);
for i=1:10
cdf_speed(i+1)=cdf_speed(i)+N(i);
end
z=polyfit(x1,cdf_speed,9);
y=polyval(z,x1);
for i=1:101
   if y(i)<0
      y(i)=0;
   end
   if y(i)>1
      y(i)=1;
   end
end
plot(x1,y,'r');xlabel('Speed (mph)');ylabel('Cumulative density function');hold on;grid;
if cdf_speed(10)>=0.999
   cdf_speed(10)=0.99991;
end
driving_distance=0;
```

```
for i=1:size(data(:,2),1);
   if data(i,2)<=100
      driving_distance=[driving_distance data(i,2)];
   end
end
x2=[5:10:95];
N=hist(driving_distance,x2);
N=N/sum(N);
cdf_distance=zeros(1,11);
for i=1:10
   cdf_distance(i+1)=cdf_distance(i)+N(i);
end
time=zeros(1,10000);
pos=zeros(1,10000);
rand('state',sum(100*clock))
for i=1:10000
   monte_carlo=0.975*rand;
   speed=interp1(cdf_speed,x1,monte_carlo);
   distance=interp1(cdf_distance,x1,monte_carlo);
   time(i)=0.75*distance/speed*60+20;
   if time(i)==NaN
      time(i)=0
```

```
    end
    pos(i)=monte_carlo;
end
z=polyfit(pos,time,8);
pp=polyval(z,[0.975:0.00001:1]);
pos=[pos [0.975:0.00001:1]];
time=[time pp];
plot(time,pos,'rd');grid;xlabel('Egress time (minutes)');ylabel('Cumulative density function');hold on;grid;

this part is for the non-MSA analysis
j=1;
for i=1:54391
    if data_1995_nonmsa.tripmiles(i)>2
        speed_nonmsa(j)=data_1995_nonmsa.speed(i);
        j=j+1;
    end
end
cdf_speed=zeros(1,11);
x=[5:10:95];
N=hist(speed_nonmsa,x);
x1=[0:10:100];
```

```
N=N/sum(N);
for i=1:10
cdf_speed(i+1)=cdf_speed(i)+N(i);
end
if cdf_speed(10)>=0.999
   cdf_speed(10)=0.99991;
end
z=polyfit(x1,cdf_speed,8);
y=polyval(z,x1);
for i=1:101
   if y(i)<0
      y(i)=0;
   end
   if y(i)>1
      y(i)=1;
   end
end
plot(x1,y,'r');xlabel('Speed (mph)');ylabel('Cumulative density function');hold on;grid
business_distance=0;
nonbusiness_distance=0;
for i=1:21153
  if data(i,2)>=500
```

```
    if data(i,5)==1
        business_distance=[business_distance data(i,4)];
    else
        nonbusiness_distance=[nonbusiness_distance data(i,4)];
     end
 end
%  end
x2=[5:10:195];
N=hist(business_distance,x2);
N=N/sum(N);
cdf_distance=zeros(1,21);
for i=1:20
    cdf_distance(i+1)=cdf_distance(i)+N(i);
end
x3=[0:10:200];
time=zeros(1,10000);
pos=zeros(1,10000);
rand('state',sum(100*clock))
for i=1:10000
    monte_carlo=0.975*rand;
    speed=interp1(cdf_speed,x1,monte_carlo);
    distance=interp1(cdf_distance,x3,monte_carlo);
```

```
    time(i)=0.75*distance/speed*60+20;
    if time(i)==NaN
        time(i)=0
    end
    pos(i)=monte_carlo;
end
z=polyfit(pos,time,8);
pp=polyval(z,[0.97:0.00001:1]);
pos=[pos [0.97:0.00001:1]];
time=[time pp];
plot(time,pos,'rs');xlabel('Access time (minutes)');ylabel('Cumulative
density function');hold on;grid;\
cdf_speed_1=zeros(1,11);
x=[5:10:95];
N=hist(speed_peak_1995_6,x);
x1=[0:10:100];
N=N/sum(N);
for i=1:10
cdf_speed_1(i+1)=cdf_speed_1(i)+N(i);
end
cdf_speed_3=zeros(1,11);
x=[5:10:95];
```

```
N=hist(speed_peak_1995_3,x);
x1=[0:10:100];
N=N/sum(N);
for i=1:10
cdf_speed_3(i+1)=cdf_speed_3(i)+N(i);
end
cdf_speed_4=zeros(1,11);
x=[5:10:95];
N=hist(speed_peak_1995_4,x);
x1=[0:10:100];
N=N/sum(N);
for i=1:10
cdf_speed_4(i+1)=cdf_speed_4(i)+N(i);
end
cdf_speed_5=zeros(1,11);
x=[5:10:95];
N=hist(speed_peak_1995_5,x);
x1=[0:10:100];
N=N/sum(N);
for i=1:10
cdf_speed_5(i+1)=cdf_speed_5(i)+N(i);
end
```

```
cdf_speed_6=zeros(1,11);
x=[5:10:95];
N=hist(speed_peak_1995_6,x);
x1=[0:10:100];
N=N/sum(N);
for i=1:10
cdf_speed_6(i+1)=cdf_speed_6(i)+N(i);
end
x11=[5:10:195];
x21=[5:10:155];
N1=distintreval(1,1:20);
N2=distintreval(2,1:16);
cdf_distance_1=zeros(1,21);
cdf_distance_2=zeros(1,17);
N1=N1/sum(N1);
N2=N2/sum(N2);
for i=1:20
   cdf_distance_1(i+1)=cdf_distance_1(i)+N1(i);
end
for i=1:16
   cdf_distance_2(i+1)=cdf_distance_2(i)+N2(i);
end
```

```
for i=1:267
   if (size((data(:,1)==msaAirInfo(i).msaID),2)~=0)
   pos=find((data(:,1)==msaAirInfo(i).msaID));
   MSAinfo(i).numair=size(msaAirInfo(i).airCnt,2);
   MSAinfo(i).msaID=msaAirInfo(i).msaID;
   if data(pos,2)<=250&(MSAinfo(i).numair==0)
      MSAinfo(i).popdensity=1;
      x3=[0:10:200];
      time=zeros(1,10000);
      pos=zeros(1,10000);
      rand('state',sum(100*clock))
      for j=1:10000
      monte_carlo=0.975*rand;
      speed=interp1(cdf_speed_1,x1,monte_carlo);
      distance=interp1(cdf_distance_1,x3,monte_carlo);
      time(j)=distance/speed*60;
      if time(j)==NaN
       time(j)=0
       end
      pos(j)=monte_carlo;
      end
      MSAinfo(i).accesstime=mean(time)+65.71;
```

```
    MSAinfo(i).egresstime=0.75*mean(time)+20;
elseif 250<data(pos,2)&data(pos,2)<=500&(MSAinfo(i).numair==0)
    MSAinfo(i).popdensity=3;
    x3=[0:10:200];
    time=zeros(1,10000);
    pos=zeros(1,10000);
    rand('state',sum(100*clock))
    for j=1:10000
    monte_carlo=0.975*rand;
    speed=interp1(cdf_speed_3,x1,monte_carlo);
    distance=interp1(cdf_distance_1,x3,monte_carlo);
    time(j)=distance/speed*60;
    if time(j)==NaN
     time(j)=0
     end
    pos(j)=monte_carlo;
    end
    MSAinfo(i).accesstime=mean(time)+65.71;
    MSAinfo(i).egresstime=0.75*mean(time)+20;
elseif 500<data(pos,2)&data(pos,2)<=750&(MSAinfo(i).numair==0)
    MSAinfo(i).popdensity=4;
    x3=[0:10:200];
```

```
    time=zeros(1,10000);
    pos=zeros(1,10000);
    rand('state',sum(100*clock))
    for j=1:10000
    monte_carlo=0.975*rand;
    speed=interp1(cdf_speed_4,x1,monte_carlo);
    distance=interp1(cdf_distance_1,x3,monte_carlo);
    time(j)=distance/speed*60;
    if time(j)==NaN
     time(j)=0
     end
    pos(j)=monte_carlo;
    end
    MSAinfo(i).accesstime=mean(time)+65.71;
    MSAinfo(i).egresstime=0.75*mean(time)+20;
elseif 750<data(pos,2)&data(pos,2)<=1000&(MSAinfo(i).numair==0)
    MSAinfo(i).popdensity=5;
    x3=[0:10:200];
    time=zeros(1,10000);
    pos=zeros(1,10000);
    rand('state',sum(100*clock))
    for j=1:10000
```

```
    monte_carlo=0.975*rand;
    speed=interp1(cdf_speed_5,x1,monte_carlo);
    distance=interp1(cdf_distance_1,x3,monte_carlo);
    time(j)=distance/speed*60;
    if time(j)==NaN
     time(j)=0
     end
    pos(j)=monte_carlo;
    end
    MSAinfo(i).accesstime=mean(time)+65.71;
    MSAinfo(i).egresstime=0.75*mean(time)+20;
elseif data(pos,2)>1000&(MSAinfo(i).numair==0)
    MSAinfo(i).popdensity=6;
    x3=[0:10:200];
    time=zeros(1,10000);
    pos=zeros(1,10000);
    rand('state',sum(100*clock))
    for j=1:10000
    monte_carlo=0.975*rand;
    speed=interp1(cdf_speed_6,x1,monte_carlo);
    distance=interp1(cdf_distance_1,x3,monte_carlo);
    time(j)=distance/speed*60;
```

```
    if time(j)==NaN
     time(j)=0
     end
    pos(j)=monte_carlo;
    end
    MSAinfo(i).accesstime=mean(time)+65.71;
    MSAinfo(i).egresstime=0.75*mean(time)+20;
elseif data(pos,2)<=250&(msaAirInfo(i).airCnt(size(msaAirInfo(i).airCnt,2))>=1)
    MSAinfo(i).popdensity=1;
    x3=[0:10:160];
    time=zeros(1,10000);
    pos=zeros(1,10000);
    rand('state',sum(100*clock))
    for j=1:10000
    monte_carlo=0.975*rand;
    speed=interp1(cdf_speed_1,x1,monte_carlo);
    distance=interp1(cdf_distance_2,x3,monte_carlo);
    time(j)=distance/speed*60;
    if time(j)==NaN
     time(j)=0
     end
    pos(j)=monte_carlo;
```

```
end
MSAinfo(i).accesstime=mean(time)+65.71;
MSAinfo(i).egresstime=0.75*mean(time)+20;
            elseif  250<data(pos,2)&data(pos,2)<=500&(msaAirInfo(i).airCnt(size(msaAirInfo(i).airCnt,2))>=1)
MSAinfo(i).popdensity=3;
x3=[0:10:160];
time=zeros(1,10000);
pos=zeros(1,10000);
rand('state',sum(100*clock))
for j=1:10000
monte_carlo=0.975*rand;
speed=interp1(cdf_speed_3,x1,monte_carlo);
distance=interp1(cdf_distance_2,x3,monte_carlo);
time(j)=distance/speed*60;
if time(j)==NaN
 time(j)=0
 end
pos(j)=monte_carlo;
end
MSAinfo(i).accesstime=mean(time)+65.71;
MSAinfo(i).egresstime=0.75*mean(time)+20;
```

```
                elseif  500<data(pos,2)&data(pos,2)<=750&(msaAirInfo(i).airCnt(size(msaAirInfo(i).airCnt,2))>=1)
    MSAinfo(i).popdensity=4;
    x3=[0:10:160];
    time=zeros(1,10000);
    pos=zeros(1,10000);
    rand('state',sum(100*clock))
    for j=1:10000
    monte_carlo=0.975*rand;
    speed=interp1(cdf_speed_4,x1,monte_carlo);
    distance=interp1(cdf_distance_2,x3,monte_carlo);
    time(j)=distance/speed*60;
    if time(j)==NaN
     time(j)=0
     end
    pos(j)=monte_carlo;
    end
    MSAinfo(i).accesstime=mean(time)+65.71;
    MSAinfo(i).egresstime=0.75*mean(time)+20;
                elseif  750<data(pos,2)&data(pos,2)<=1000&(msaAirInfo(i).airCnt(size(msaAirInfo(i).airCnt,2))>=1)
    MSAinfo(i).popdensity=5;
    x3=[0:10:160];
```

```
time=zeros(1,10000);
pos=zeros(1,10000);
rand('state',sum(100*clock))
for j=1:10000
monte_carlo=0.975*rand;
speed=interp1(cdf_speed_5,x1,monte_carlo);
distance=interp1(cdf_distance_2,x3,monte_carlo);
time(j)=distance/speed*60;
if time(j)==NaN
 time(j)=0
 end
pos(j)=monte_carlo;
end
MSAinfo(i).accesstime=mean(time)+65.71;
MSAinfo(i).egresstime=0.75*mean(time)+20;
elseif data(pos,2)>1000&(msaAirInfo(i).airCnt(size(msaAirInfo(i).airCnt,2))>=1)
MSAinfo(i).popdensity=6;
x3=[0:10:160];
time=zeros(1,10000);
pos=zeros(1,10000);
rand('state',sum(100*clock))
for j=1:10000
```

```
        monte_carlo=0.975*rand;
        speed=interp1(cdf_speed_6,x1,monte_carlo);
        distance=interp1(cdf_distance_2,x3,monte_carlo);
        time(j)=distance/speed*60;
        if time(j)==NaN
         time(j)=0
         end
        pos(j)=monte_carlo;
        end
        MSAinfo(i).accesstime=mean(time)+65.71;
        MSAinfo(i).egresstime=0.75*mean(time)+20;
    end
    MSAinfo(i).airID=msaAirInfo(i).airID;
end
end
```