Runway Exit Speed Estimation Models

Mani Bhargava Reddy Bollempalli

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

Master of Science

In

Civil Engineering

Antonio A. Trani, Chair

Susan Hotle

Kevin P. Heaslip

July 31, 2018

Blacksburg, Virginia

Keywords: Runway Exit, Runway Geometry, Runway Exit Database

Runway Exit Speed Estimation Models

ABSTRACT

Mani Bhargava Reddy Bollempalli

Increasing air traffic in the U.S.A has led to runway capacity limitations at the airports. Increasing the capacity of the existing runways involves reducing the runway occupancy time of an aircraft landing on a runway. The location of runway exits plays an important role in minimizing the runway occupancy time. Locating an optimal location for an exit is getting complex with a rapid increase in the number of aircraft types. So, the Air Transportation and Systems Laboratory at Virginia Tech developed the Runway Exit Interactive Design Model (abbreviated as REDIM). This model finds the optimal exit location considering multiple aircraft and a variety of environmental conditions.

To find the optimal exit location, REDIM simulates the landing aircraft behavior. The kinematic model simulating the aircraft landing behavior in REDIM using pseudo-nonlinear deceleration heuristic algorithm. REDIM models the aircraft landing behavior into five phases. The five phases are: 1) a flare phase, 2) a free roll period occurring between the aircraft touchdown and the brakes initiation 3) the braking phase, 4) a second free roll phase starting after the braking phase and ending before the turnoff maneuver and 5) a turnoff maneuver phase. The major contributors to the runway occupancy time (ROT) are the braking phase (60% of ROT) and the turnoff phase (25% of ROT).

Calculating the turnoff time requires few input variables such as deceleration rate along the turnoff and the speed at which an aircraft takes an exit (exit speed at the point of curvature). The deceleration rate along the turnoff is specific to every aircraft.

This study involves predicting the exit speed at the point of curvature based on the type of exit taken. It begins with collecting the exit geometry parameters of 37 airports in the U.S.A. The exit geometry parameters define the type of exit. The ASDE-X data provides the observed exit speeds at the point of

curvature for these exits. This study examines a few models with observed exit speeds as the response variable and exit geometry as the predictor variables.

Runway Exit Speed Estimation Models

GENERAL AUDIENCE ABSTRACT

Mani Bhargava Reddy Bollempalli

Increasing air traffic in the U.S.A has led to runway capacity limitations at the airports. Increasing the capacity of the existing runways involves reducing the runway occupancy time of an aircraft landing on a runway. The location of runway exits plays an important role in minimizing the runway occupancy time. Locating an optimal location for an exit is getting complex with a rapid increase in the number of aircraft types. So, the Air Transportation and Systems Laboratory at Virginia Tech developed the Runway Exit Interactive Design Model (abbreviated as REDIM). This model finds the optimal exit location considering multiple aircraft and a variety of environmental conditions.

To find the optimal exit location, REDIM simulates the landing aircraft behavior. The kinematic model simulating the aircraft landing behavior in REDIM using pseudo-nonlinear deceleration heuristic algorithm. REDIM models the aircraft landing behavior into five phases. The five phases are: 1) a flare phase, 2) a free roll period occurring between the aircraft touchdown and the brakes initiation 3) the braking phase, 4) a second free roll phase starting after the braking phase and ending before the turnoff maneuver and 5) a turnoff maneuver phase. The major contributors to the runway occupancy time (ROT) are the braking phase (60% of ROT) and the turnoff phase (25% of ROT).

Calculating the turnoff time requires few input variables such as deceleration rate along the turnoff and the speed at which an aircraft takes an exit (exit speed at the point of curvature). The deceleration rate along the turnoff is specific to every aircraft.

This study involves predicting the exit speed at the point of curvature based on the type of exit taken. It begins with collecting the exit geometry parameters of 37 airports in the U.S.A. The exit geometry parameters define the type of exit. The ASDE-X data provides the observed exit speeds at the point of

curvature for these exits. This study examines a few models with observed exit speeds as the response variable and exit geometry as the predictor variables.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Antonio Trani for his guidance throughout the project and giving me an opportunity to work at Air Transportation Systems Laboratory.

To the individuals in the FAA, I am grateful for your financial support in my graduate career.

To Navid Mirmohammadsadeghi, thank you for providing the data for analysis. To, Nick Hinze and Howard Swingle, thank you very much for your input during the project. To Mihir Rimjha, thank you for helping with the runway exit database.

Additionally, I would like to thank Dr. Hotle for providing valuable suggestions during the model development. Also, I am grateful to Dr. Heaslip for participating in my committee and providing guidance during the second semester of my graduate career

To my friends, I thank you for making me feel home. This thesis paper is dedicated to my parents and extended family who provided constant support during my education.

Table of Contents

List of Tables

List of Figures

Nomenclature

AC – Advisory Circular

ADG – Aircraft Design Group

ANN – Artificial Neural Network

ASDE – Airport Surface Detection Equipment

ATL - Hartsfield–Jackson Atlanta International Airport

CLE - Cleveland Hopkins International Airport

CLT - Charlotte Douglas International Airport

DCA - Ronald Reagan Washington National Airport

DEN - Denver International Airport

DFW - Dallas/Fort Worth International Airport

DTW - Detroit Metropolitan Wayne County Airport

FAA – Federal Aviation Administration

IAD - Washington Dulles International Airport

IAH - George Bush Intercontinental Airport

ICAO – International Civil Aviation Organization

REDIM – Runway Exit Design Interactive Model

ROT – Runway Occupancy Time

TDG – Taxiway Design Group

LAS - McCarran International Airport

LAX - Los Angeles International Airport

LGA - La Guardia Airport

MCO - Orlando International Airport

MEM - Memphis International Airport

MSP - Minneapolis–Saint Paul International Airport

ORD - O'Hare International Airport

PHX - Phoenix Sky Harbor International Airport

SAN - San Diego International Airport

SFO - San Francisco International Airport

CHAPTER 1: INTRODUCTION

1.1 Background and Problem Statement

Airports play a big role in urban transportation. They help connecting people, corporations and governments leading to different business activities. In the recent years, airports have accommodated increasing traffic leading to higher operations count. Airport delays and congestion have been major problems to the aviation industry. 2008 had the worst congestion problem with around 29% of the flights in the USA delayed or canceled [1]. There are two ways to solve this problem. The first approach is to improve the infrastructure at the airports. This approach is direct and solves the problem by increasing the capacity. However, matching the supply with the demand at the airports takes a great deal of time. The planning and construction of the infrastructure at the airport is a long-term process and expensive. The second approach tries to improve the efficiency of the existing infrastructure.

There are many pieces of infrastructure at the airports which work in harmony. A bottleneck in the capacity at any individual level of the infrastructure can cause delays. Every component of the infrastructure influences the capacity. The capacity is defined as the processing capability of a service facility over some period [2]. The capacity of a critical component of infrastructure at an airport dictates the capacity of the airport. An airport with sufficient runways and few gates tend to experience congestion at the gates. Similarly, airports with sufficient number of gates and few runways might experience delay at the runways. Instead of building new runways at the airport, efficiency of runways can be increased by reducing the runway occupancy time. In the USA, simultaneous runway occupancy by two aircraft is not allowed [3]. Which implies one aircraft is allowed on the runway at a given time. This helps reduce runway collisions. The following arrival can only take place once the previous aircraft clears the runway. The following aircraft initiates a go-around if the preceding aircraft has not cleared the runway. Therefore, to increase the runway throughput it is beneficial to reduce the runway occupancy time. Where Runway Occupancy time for an arriving aircraft is measured from the runway threshold to clearing the runway at the runway exit for the next arrival.

1.2 Previous Research in Runway Exit Design

Robert Horonjeff started the initial work on this subject. Horonjeff proposed a mathematical model to find the locations of exit taxiways and proposed standards for 45^0 and 30^0 angle geometries. FAA and ICAO later adopted these exit angle geometries [7]. Horonjeff's work was the first to acknowledge the relationship between exit location and exit geometry resulting in a mathematical model to find the exit taxiway location. However, the limitations of this study include only using two exit speeds. Also, his study used a limited aircraft fleet mix. The aircraft fleet mix needed revision for further use. Horenjoff studied the effect of turning radius on exit speed of Boeing KC-135 aircraft. Based on this study, FAA in 1970, proposed the standards for designing the high-speed exits. This design used 1800 ft radius of curvature as the centerline of the 30^0 exit turnoff. The design speed of the high-speed exit was 26.7 m/sec (60 mph). Similarly, the 45^0 exits had a design speed of 17 m/sec.

Schoen et.al [8] developed a computer algorithm which computes the coordinates of the exit geometry by considering the exit speed and the turning ability of the aircraft. Schoen's study showed that aircraft moment of inertia plays an important role in determining the turnoff maneuver trajectory. This work discussed the possibility of very low ROT values such as 30 seconds by using large turning radius and very high exit speeds.

Carr et.al [9] tried new turnoff for optimal runway occupancy times. Carr tested different types of exit geometries, lighting scenarios in a Lockheed L1011 flight simulator and came up with a 'wide throat' geometry. This geometry has an entrance spiral length of 244 m (800 ft) which tapers off with 122 m (400 ft) turn radius. This exit geometry is implemented at many commercial airports such as Orlando International, Miami International, Indianapolis and Baltimore-Washington International airports. In cases where the distance between the runway and the closest taxiway distance is less than 183 m (600 ft), the wide throat exit is a suitable one. However, the turn radius of this exit is the limiting factor in using this exit at high speeds.

The latest advisory circular AC/150-5300-13A suggests many changes compared to the old 30^0 exit geometry adopted in the early seventies. It currently suggests a turn radius of 457 m (1500 ft) for the exit. This helps the pilot to find the difference between other exits and a standard high-speed exit. The aerodrome design manual (part 2) advises use of 275m central radius for Rapid Exit Taxiways. They also suggest an inside fillet curve radius of 253 m for these exits.

Horonjeff et.al [6] proposed an optimization tool to maximize the aircraft arrival rate at runways under saturated operational conditions. However, there is uncertainty over the bivariate distribution of mean distances and time taken to decelerate to a predetermined exit speed for an aircraft [2]. Moreover, this model did not address the environmental factors affecting the airfield such as wind speed. Other factors such as aircraft operational factors such as aircraft landing weight variations are not addressed also.

Daellenbach et.al [11] in 1974, based his model on Horonjeff's approach and added a few details. Daellenbach made the model more realistic by removing the assumption of a specific arrival pattern. However, Daellenbach's model requires more extensive data on landing distributions under multiple scenarios.

Joline et.al [14] in 1974 developed a dynamic programming model that finds the optimal exit locations and their number along the runway. The objectives for this model are reducing the ROT and exit construction cost. Moreover, Joline's model classified the aircraft into three categories. Aircraft size was the factor used in classification. Consequently, distributions of ideal exit locations for different aircraft classes are found based on the data obtained at Chicago O'Hare Airport (ORD). The ideal locations are then found by weighing these individual exit locations with the proportions of the three aircraft classes. Like the other models, Joline's model didn't address critical parameters such as design exit speed and aircraft landing weight.

Trani, Hobeika and Sherali et al in 1991 used dynamic programming algorithm with the continuous simulation programming to find out optimal exit locations. Moreover, they have considered realistic airport environment and included other aircraft specific factors in the model. This model adds more flexibility and realism to finding the optimal exit locations.

1.3 Aircraft Landing Process

The kinematic model REDIM uses deploys a pseudo-nonlinear deceleration heuristic algorithm to simulate the aircraft landing behavior on a runway. The aircraft landing profile in REDIM can be modeled into following phases: 1) a flare phase, 2) a free roll period occurring between the aircraft touchdown and the brakes initiation, 3) the braking phase, 4) a second free roll phase starting after the braking phase and ending before the turnoff maneuver and 5) a turnoff maneuver phase. The major contributors to the runway occupancy time (ROT) are the braking phase (60% of ROT) and the turnoff phase (25% of ROT) [12].

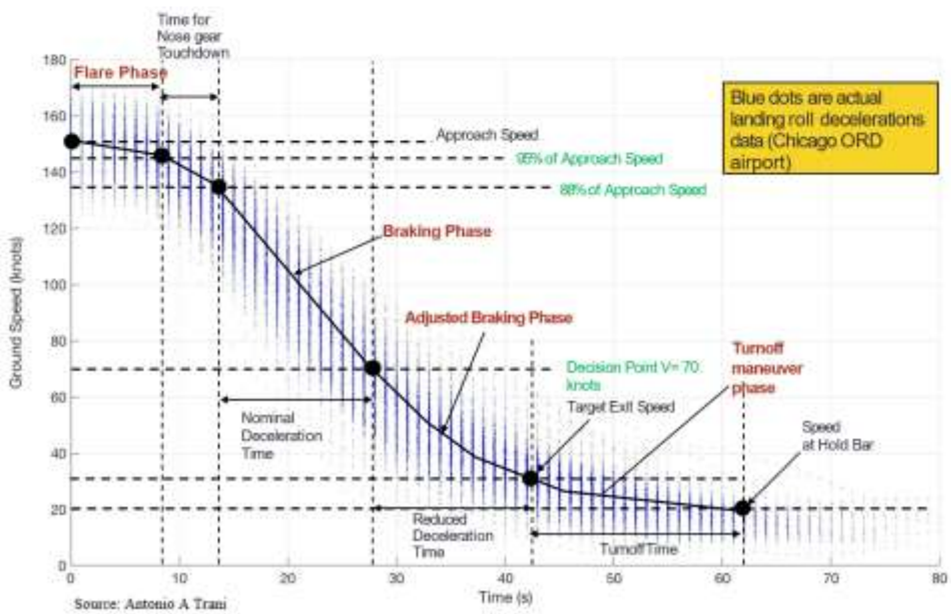


Figure 1 : Aircraft Landing Profile. Source-Antonio A Trani.

1.4 Turnoff Maneuver Phase

The turnoff trajectory in REDIM 2.0 approximates the instantaneous turn radius of the aircraft along the turnoff by using a reduced order model. The REDIM uses a fourth-order aircraft dynamic model by considering the three degrees of displacement freedoms and yawing motion associated with the turning aircraft [12]. The ROT is the sum of the times taken to complete all the four phases in the landing process.

Minimizing the ROT involves minimizing times taken to complete these individual phases. The turnoff time in REDIM 2.0 is estimated using a continuous simulation algorithm which predicts the turnoff trajectory for an aircraft from the point of curvature of the exit to the point where aircraft effectively clears the runway edge (for high speed exits, the wing tip needs to be cleared. For right angle exits, the tail needs to be cleared).

Calculating the turnoff times involves using input parameters such as deceleration rate along the turnoff and the exit speed at the point of curvature (entry point of the exit). The deceleration rates are aircraft specific. Using turnoff deceleration rate distributions from ASDE-X data (from fourteen airports analyzed for the model) a deceleration rate is picked randomly during each simulation to calculate the turnoff time. The other input parameter, the exit speed at the point of curvature is highly dependent on the type of exit being taken. Exits with high turn radius like the high-speed exits have higher exit speeds. Similarly, the exits with low turn radius such as the right angle exits have lower exit speeds. Therefore, the exit speed at the point of curvature needs to be predicted before calculating the turnoff time.

1.5 Research Objectives

The objective of this study is to predict the exit speed at the point of curvature using the exit geometry parameters. The exit speed at the point of curvature is highly dependent on the type of exit taken. The input parameters for the model differentiate between the types of exit taken. The input parameters for these models include the exit geometry parameters such as path length, radius of the turn and exit angle.

Calculating the turn-off time involves using the predicted exit speeds at the point of curvature. REDIM 2.0 uses mean exit speed at the point of curvature for different exit types to calculate the turn-off time. However, REDIM 3.0 uses the models developed in this study to predict the mean exit speed at the point of curvature. This model also helps predict the mean exit speed at the point of curvature for non standard exits in REDIM 3.0. This was not the case with REDIM 2.0

1.6 Thesis Outline

The thesis begins with the background and the problem associated with increasing the runway efficiency and reducing the ROT. Followed by this, it discusses about previous research work done in this field. It discuses about the advances in the research starting with Horenjoff's work in 1950's on exit taxiway locations to Trani's work on developing a computer interactive model to find optimal exit locations. Later, it discusses about the aircraft landing process uncovering the basic way the landing process is analyzed from the runway threshold to the exit hold bar. The landing process is classified into four phases. The last phase is the turn off phase starting from the point of curvature of the exit to the hold bar. ROT is the sum of individual times to complete all the phases. Calculating the turnoff time needs exit speed at the point of curvature as one of the inputs. The following chapters better explain how the exit speed is calculated at the point of curvature.

Chapter 2 discusses about the runway exit database at 37 ASDE-X airports in U.S.A. This database has information about exit geometries of all the airports analyzed. The information included in the database helps predict the exit speed at the point of curvature. Later in chapter 3, the variables in the runway exit database are used as explanatory variables in different models tried to calculate the exit speed at the point of curvature.

Chapter 3 discusses about the models used to predict the exit speed at the point of curvature. The observed exit speeds at the point of curvature are obtained from the Airport Surface Detection Equipment -X (ASDE-X) data from different airports. The ASDE-X data comprises of the information about track coordinates of all aircraft operations that took place at different facilities. Using the track information, the exit taken by an aircraft is known. Consequently, the observed exit speed at the exit is known from the exit taken and the track information. The observed exit speed is the response variable and the explanatory variables are the exit geometry parameters from the runway exit database.

Chapter 4 summarizes the findings of the study and makes recommendations for future research.

CHAPTER 2: RUNWAY EXIT DATABASE

2.1 Runway Exit Database Fields

The Runway exit database contains information about all the exits analyzed. The data is populated using Google Earth. The data collected includes information about a) runway exit geometry b) runway exit location (The distance of the point of curvature from the runway threshold) and additional information on the hold bar location and parallel taxiway distance.

This database has an airport as the primary structure with various fields including:

- Runway ID
- Exit ID
- Exit location (The distance from threshold to the starting point of curvature in ft)
- Exit path length (The distance of the point of curvature to hold bar in ft)
- Exit angle (degrees)
- Hold bar distance (The distance between runway centreline and the hold bar in ft)
- Equivalent radius (ft)
- Closest taxiway distance (The distance to the closest taxiway in ft)
- Taxiway width (ft)

Table 1 : A Typical Template of the Runway Exit Database.

Field	Format	Example	Remarks
Airport	String	IAD	FAA code
Runway ID	String	19L	Name of the runway
Exit ID	String	K3	Name of the exit
Exit Location	Numeric	3291 ft	The distance of exit from the threshold in ft
Exit path length	Numeric	1118 ft	Distance from the point of curvature to the hold bar in ft

Hold bar distance	Numeric	341 ft	Distance from the runway centerline to the hold bar in ft
Runway Angle	Numeric	180	Runway heading in degrees in ft
Equivalent radius	Numeric	1800 ft	Turn radius associated with the exit in ft
Closest taxiway distance	Numeric	700 ft	Distance from the runway centerline to the closest taxiway distance in ft
Taxiway Width	Numeric	80 ft	The width of the taxiway in ft

2.2 Definitions

The runway exit geometry is defined using the following parameters

1) Path Length (ft)
2) Radius (ft)
3) Exit Angle (degrees)
4) Taxiway Width (ft)
5) Hold bar Distance (ft)
6) Closest Taxiway Distance (ft)

1. Path Length – It is the distance measured in feet from the entry-point/Point of Curvature of an exit to the Hold Bar. See in **Figure 2**.

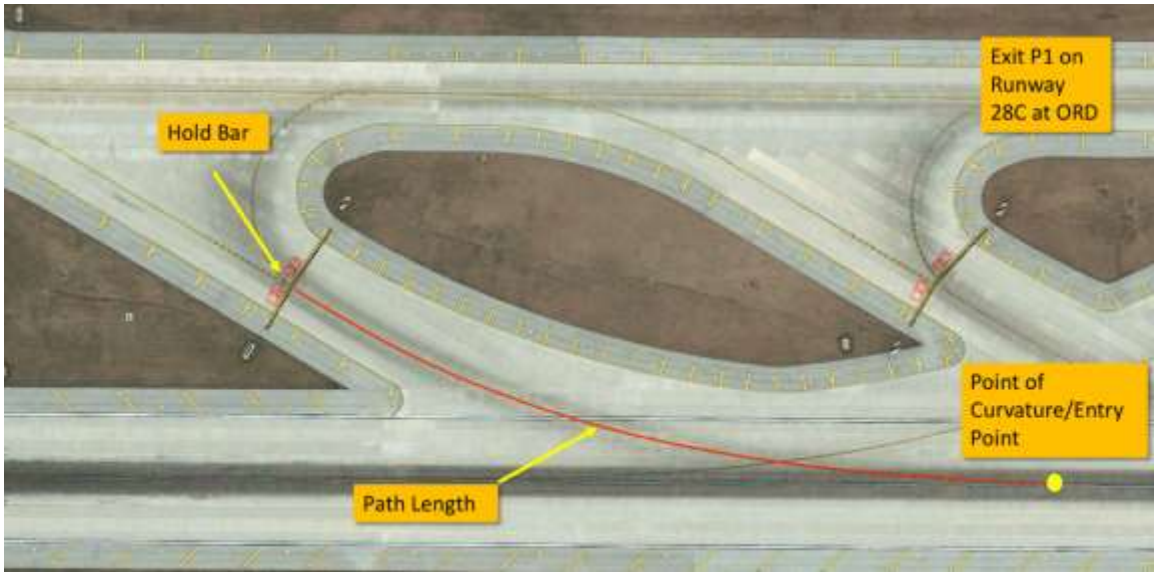


Figure 2: Path length is the Red Curve from Point of Curvature of an Exit to the Hold Bar.

2. Radius – It is the radius of the circle with runway centerline as the tangent at the point of curvature of the exit. Typically for a high speed exit, the radius is equal to 1800 ft. See in **Figure 3**.

Figure 3 : Radius of the Turn for a High-Speed Exit W3 on Runway 10R at ORD.

3. Exit Angle – It is the angle at which an exit steers away from the runway centerline. For high speed exit, the exit angle is 30^0. See in the **Figure 4**.

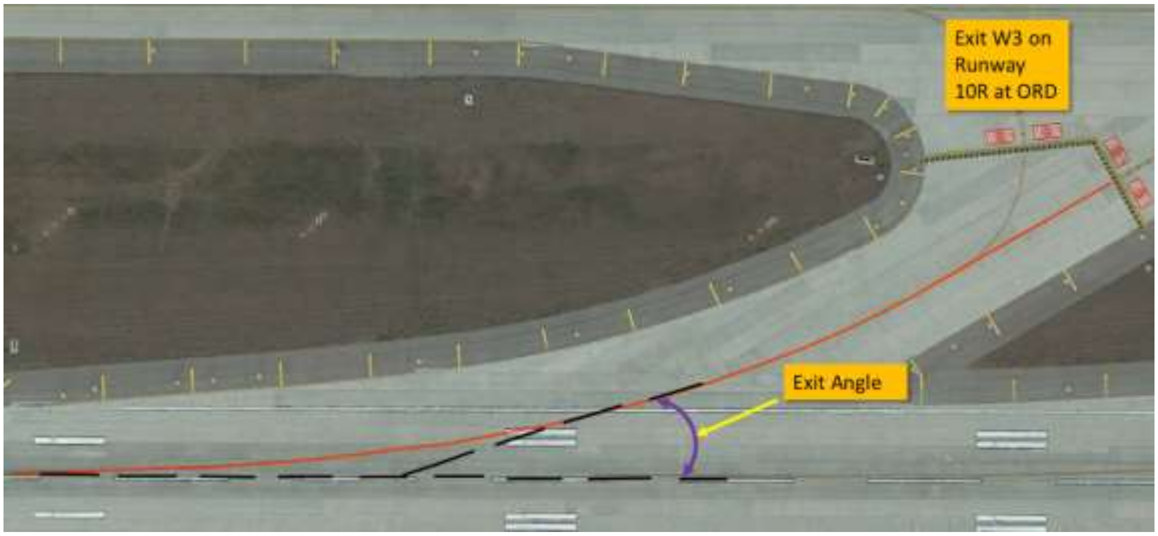


Figure 4 : Exit Angle of a High-Speed Exit W3 on Runway 10R at ORD is 30^0.

4. Taxiway Width – This corresponds to the width of the taxiway taken. See in the **Figure 5**.

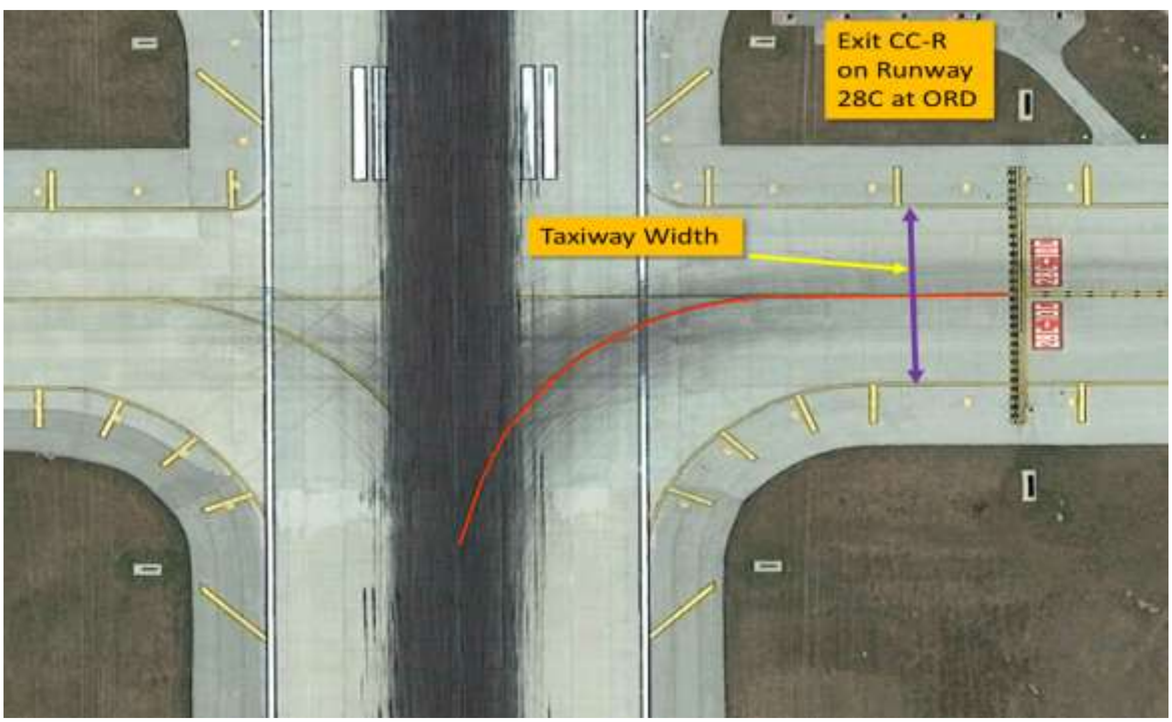


Figure 5 : Taxiway Width of a Exit CC-R on Runway 28C, ORD is 115 ft.

5. Hold Bar Distance – The distance measured from the runway centerline to the hold bar centerline is hold bar distance. See in the **Figure 6.**

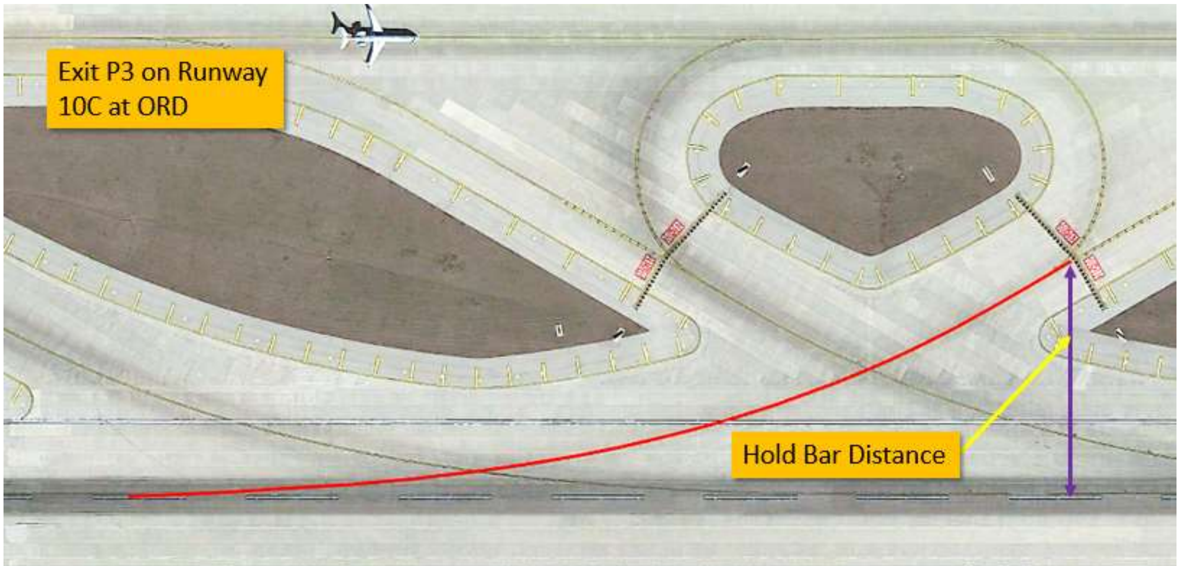


Figure 6 : Hold Bar Distance at Exit P3 on Runway 10C, ORD is 305 ft.

6. Closest Taxiway Distance – The distance measured from the runway centerline to the centerline of the closest parallel taxiway. See in the **Figure 7**.

Figure 7 : Closest Taxiway Distance at Exit P3 on Runway 10C, ORD is 600 ft.

2.3 State of Runway Exits in the USA

Many airports analyzed in the runway exit database have different exit types. The exits in the runway exit database are broadly classified into following categories:

- Standard High-Speed Exits- These exits have exit angle lower than 35^0 and path length greater than 1000 ft.
- Non Standard High-Speed Exits- These exits have exit angle lower than 35^0 and path length lower than 1000 ft.
- Right Angle Exits- These exits have exit angles between 85^0 and 95^0.
- Wide Throat Exits- These exits have exit angles lower than 45^0 and path length greater than 800 ft.
- Obtuse Angle Exits- These exits have exit angles greater than 95^0.
- Non Standard Exits- These exits do not belong to any of the above mentioned exit types.

The **Figure 8** shows that very few airports such as DFW, DEN, IAD, and DTW have all standard high-speed exits. Airports such as LGA have many non-standard high-speed exits with very few standard high-speed exits. On the other hand, MCO has many wide throat exits due to which the airport doesn't have any high-speed exits. Few busy airports such as SFO, PHX, and CLE have no standard high-speed exits.

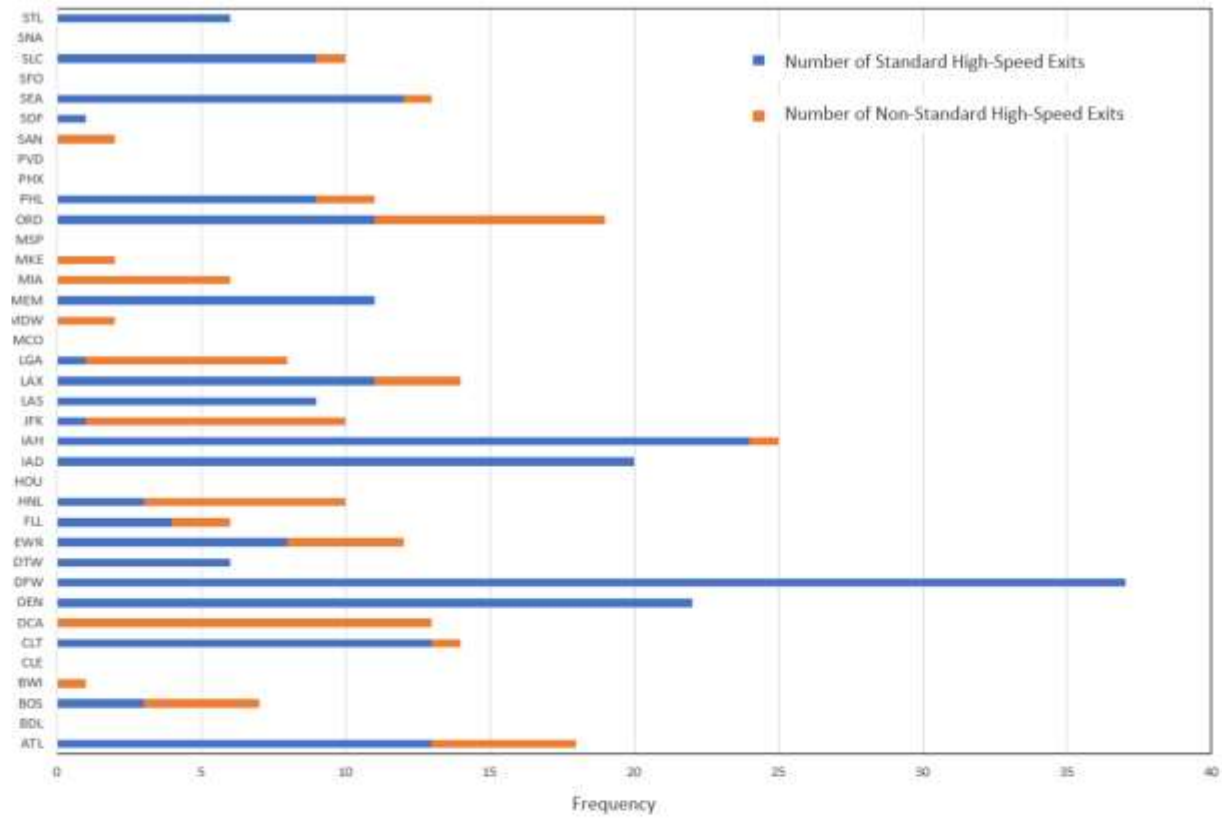


Figure 8 : Number of Standard and Non-Standard High-Speed Exits at Different Airports in the USA.

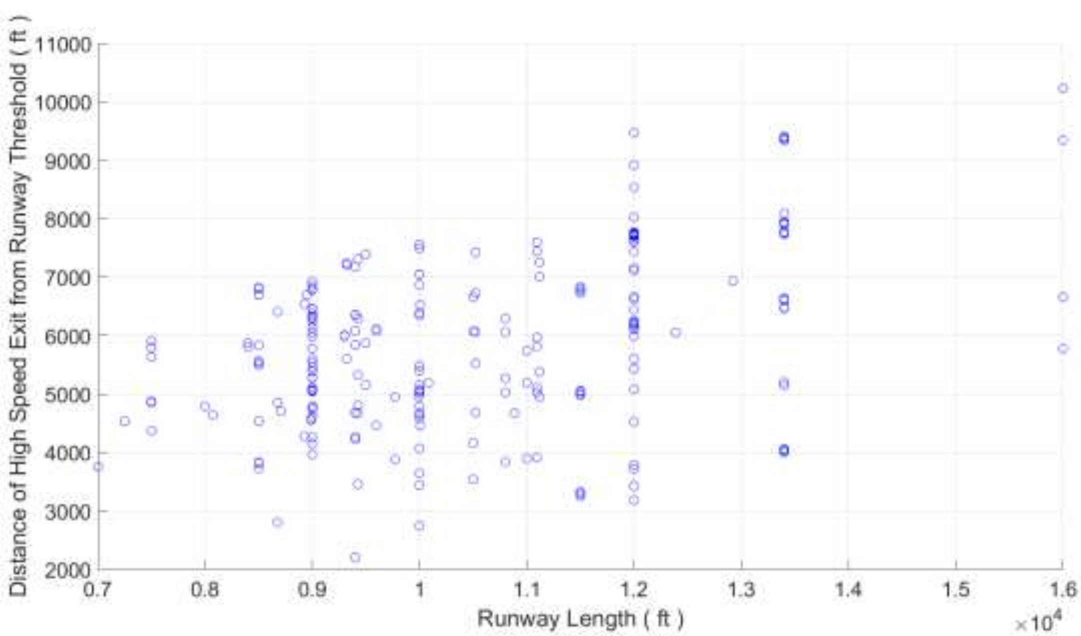
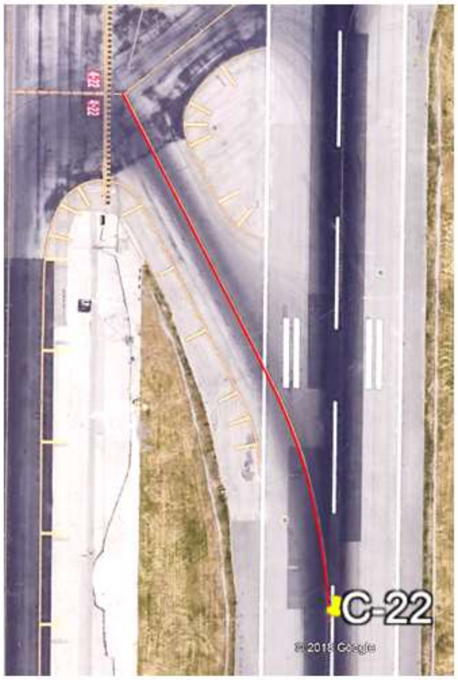


Figure 9 : Distance of High-Speed Exit from Runway Threshold vs Runway Length for High-Speed Exits.

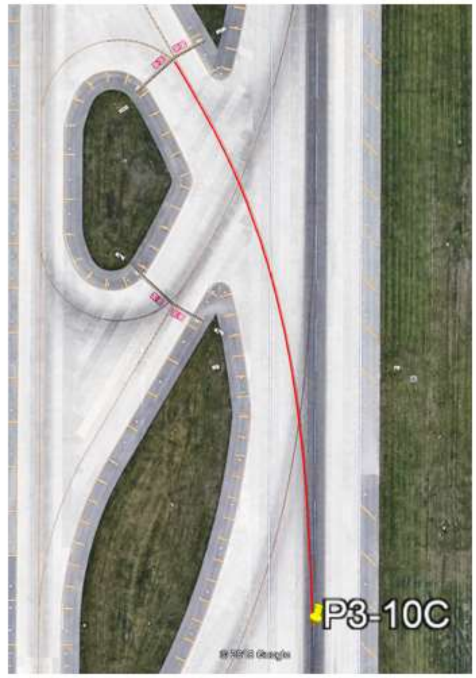
2.3.1 High-Speed Exits

According to the data, **Figure 9** shows that high-speed exits are placed at a minimum distance of 2201 ft from runway threshold. However, at many airports, high-speed exits start 3200 ft from the runway threshold. The closest high-speed exit to the runway threshold (2201 ft) is found on runway 26L at IAH. Aircraft fleet mix is a driving factor in placing the high-speed exits along the runway. A reason for closer high-speed exit at IAH might be the aircraft fleet mix using that runway. If many general aviation aircraft use runway 26L, it is appropriate to place the high-speed exit closer to the runway threshold.

The standard high-speed exits have a minimum path length of 1000 ft and exit angle of less than 35 degrees. However, the exit angles reported in the exit database are from Google Earth. Exit angle measurements performed using Google Earth have technical errors and are not accurate. Sometimes, the exit angle measured for a typical high-speed exit is more than 30 degrees. Therefore, all the exits in the database having exit angles less than 35 degrees are considered to be high-speed exits. However, an exit angle alone cannot differentiate a standard high-speed exit from a non-standard high-speed exit. Non-standard high-speed exits have lower path lengths and lower turn radius. As mentioned earlier, small airports such as LGA have many non-standard high-speed exits. For example, the highly used exit Charlie on runway 22 (C-22) at LGA is a non-standard high-speed exit. Whereas, exit P3-10C at ORD is a standard high-speed exit.



Exit C-22 at LGA



Exit P3-10C at ORD

Figure 10 : Exit C on Runway 22 at LGA is a Non-Standard High-Speed Exit. Exit P3 on Runway 10C at ORD is Standard High-Speed Exit.

Table 2 : Exit Geometry Parameters of Non-Standard High-Speed exit (C-22) and Standard High-Speed Exit (P3-10C).

Field	C-22	P3-10C
Airport	LGA	ORD
Runway ID	22	10C
Exit ID	C	P3
Exit Angle (degrees)	30	30
Exit path length (ft)	620	1295
Hold bar distance (ft)	245	310
Radius (ft)	750	1800

Figure 10 shows that pilots using exit C-22 have less freedom to maneuver the exit at higher exit speeds. However, in case of exit P3-10C, the pilots tend to take the exit at higher exit speeds.

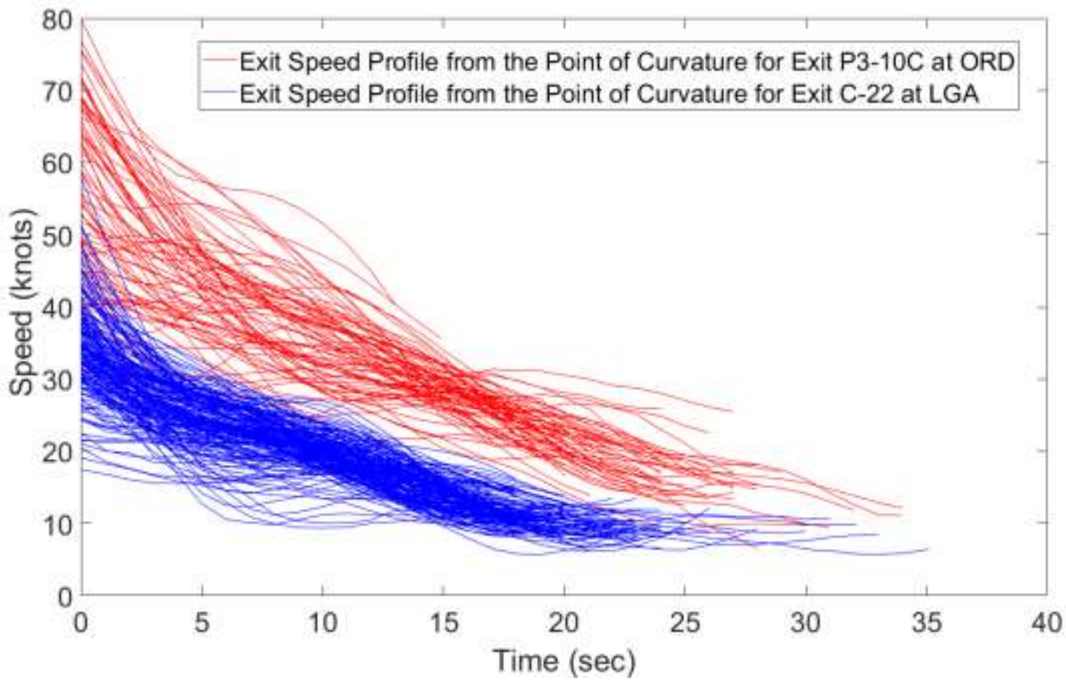


Figure 11 : The Exit Speed Profile of a Standard (P3-10C) and a Non-Standard High-Speed Exit (C-22).

The **Figure *11*** shows an aircraft exit a standard high-speed exit P3-10C at ORD with a mean speed of 55 knots. Whereas, for a non-standard high-speed exit such as C-22 at LGA, the pilot takes it at a mean speed of 35 knots.

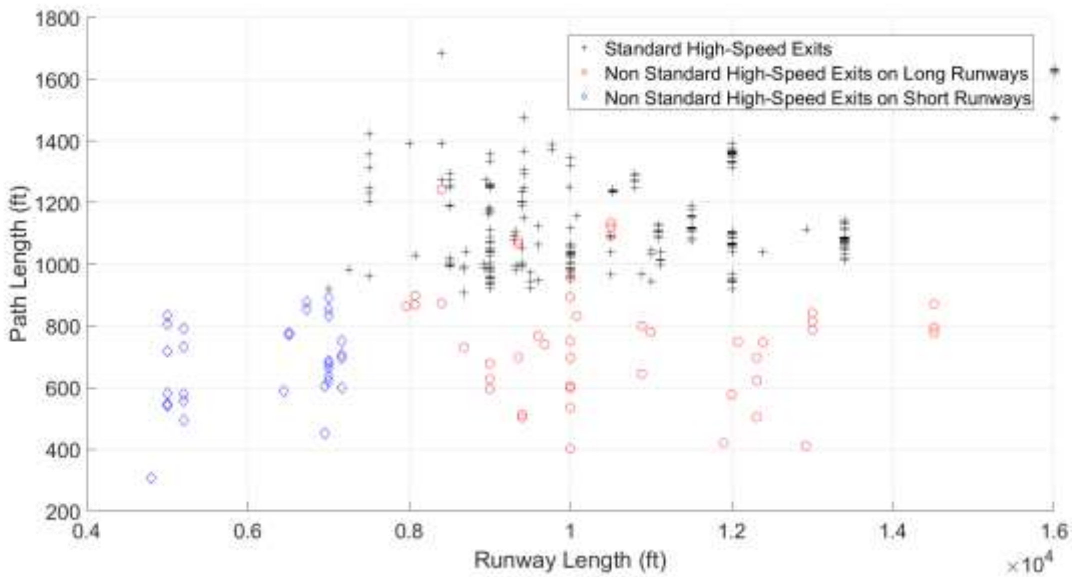


Figure 12 : Path Length (ft) vs Runway Length (ft) for Standard High-Speed Exits and Non-Standard High-Speed Exits on Short and Long Runways.

In the runway exit database, there are 238 records of standard high speed exits at airports analyzed. Additionally, there are 77 non-standard high speed records. An exit is said to be a standard high speed exit when the path length of the exit is greater than 900 ft, exit angle is lower than 35^0 and turn radius is more than 1400 ft. All other exits with exit angle below 35^0 not satisfying the criteria above mentioned are non-standard high speed exits. **Figure *12*** shows that out of 77 non-standard high speed exits, 31 are found on runways shorter than 7500 ft.. The other 46 are found on runways greater than 7500 ft. It is comprehensible behind the reason why many non-standard high speed exits are found along short runways. It is because these short runways are typically used by aircraft types ADG I and ADG II (sometimes ADG III) and It's uneconomical to build standard high speed exits over such small runways.

However, there are considerable number of non-standard high speed exits on long runways. The reason behind such non-standard high speed exits on long runways is probably due to the space constraint around the runway. Sometimes, the hold bar distance for the non-standard high-speed exits is shorter than the hold bar distance of standard high-speed exits leading to shorter path length for non-standard high-speed exits.

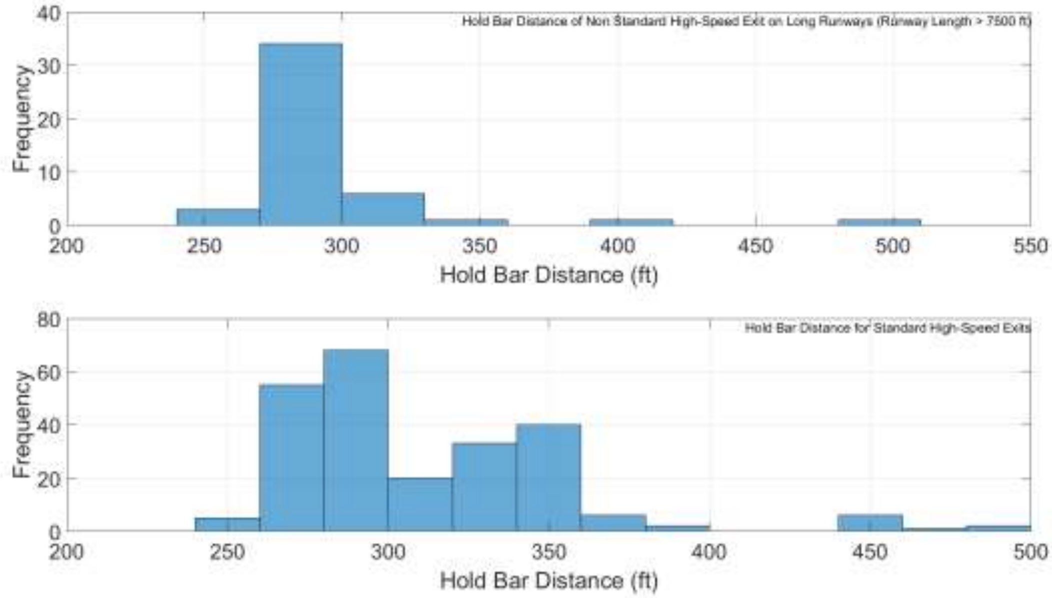


Figure 13 : Histogram of Hold Bar Distance for Non-Standard High-Speed Exits 'Top'. Histogram of Hold Bar Distance for Standard High-Speed Exits 'Bottom'.

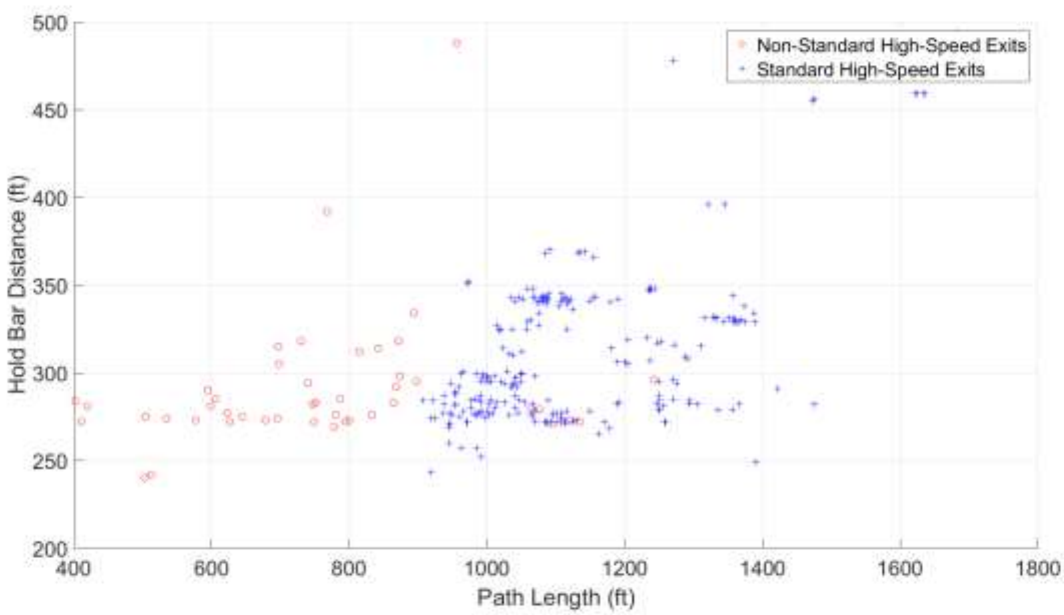


Figure 14 : Hold Bar Distance vs Path Length for Standard and Non-Standard High-Speed Exits.

As observed in the **Figure 13**, the average hold bar distance for standard high-speed exit is 311 ft. Similarly, the average hold bar distance for a non-standard high-speed exit is 290 ft. However, one can observe that there are many standard high-speed exits with hold bar distance around 270 ft similar to non-standard high-speed exits.

Path Length is an important factor in determining whether a high-speed exit is standard or non-standard. As mentioned earlier, exits having shorter path lengths tend to be non-standard high-speed exits and vice versa. One important reason for lower path lengths is considered to be the hold bar distance. However, as **Figure 14** shows the standard high-speed exits tend to have higher hold bar distance. Nevertheless, there are many standard high-speed exits with hold bar distances approximately equal to that of non-standard high speed exits.

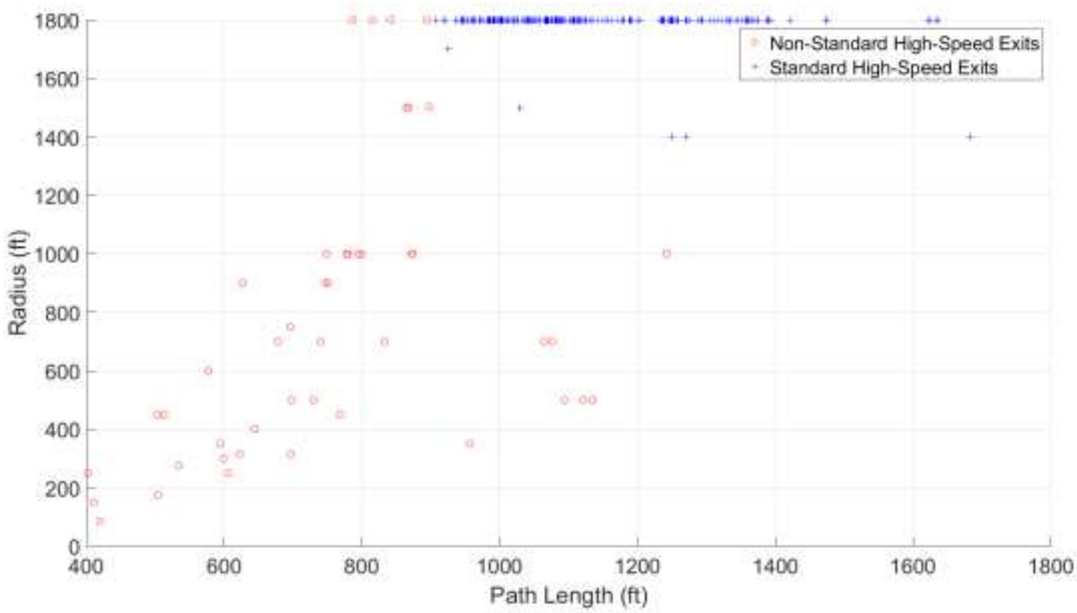


Figure 15 : Radius vs Path Length for Standard and Non-Standard High-Speed Exits.

Therefore, one can state that lower hold bar distances may cause non-standard high speed exits. However, this might not be the only explanation. The radius of the turn might offer a better explanation for lower path lengths in non-standard high-speed exits. From the **Figure 15**, the path length of the non-standard high-speed exits increases with the radius of the turn. The radius of the standard high-speed is typically 1800 ft.

Looking into the data, the non-standard high-speed exits have lower radius and hold bar distance. More importantly, if a high-speed exit has a lower radius it tends to be a non-standard high-speed despite having good hold bar distance. Therefore, when designing for a new high-speed exit, the designer should not undermine the radius of the turn for a high-speed exit.

2.3.2 Right Angle Exits

The right angle exits according to FAA advisory circular AC 150/5300 – 13A are designed based on the distance between the runway centerline and taxiway centerline and taxiway design group (TDG) of the aircraft using the exit.

Dimension (see Figure 4-17)	**TDG**													
	2				**4**			**5**	**6**				**7**	
Runway Centerline to Taxiway Centerline Distance	240	250	300	350	300	350	400	400	400	450	500	550	500	550
W-0 (ft)	17.5	17.5	17.5	17.5	25	25	25	37.5	37.5	37.5	37.5	37.5	41	41
W-1 (ft)	27	27	26	26	34	38	37	49	53	53	54	54	55	55
W-2 (ft)	50	50	49	49	62	77	75	84	102	105	101	99	100	99
W-3 (ft)	28	27	25	24	49	43	38	52	63	58	53	50	54	52
L-1 (ft)	190	186	185	185	288	322	316	312	414	429	432	433	394	398
L-2 (ft)	75	75	75	75	125	128	130	125	175	164	155	150	154	150
L-3 (ft)	50	50	49	49	119	77	75	84	102	92	93	93	94	93
R-Fillet	0	0	0	0	90	0	0	0	0	0	0	0	0	0
R-CL (ft)	65	65	65	65	110	105	100	100	135	130	125	120	125	120
R-Outer	82	82	82	82	138	130	129	165	200	192	190	186	200	198

Figure 16 : Dimensions for Runway Entrance/Exit Taxiways (Table 4-12 from AC 150/5300–13A).

Figure 16 shows that TDG 2 requires a minmum centerline radius of 65 ft irrespective of the runway centerline to taxiway centerline distance. The maximum centerline radius of 135 ft is observed in the case of TDG 6 when the runway centerline to taxiway centerline distance is 400 ft.

At airports analyzed, the exit angle is used to select the right angle exits. Right Angle exits are the exits with exit angles between 85^0 and 95^0. A total of 1693 exits satisfy this criteria. Calculating the exit angles using Google Earth is inaccurate. Moreover, it also involves human error. A wide range of exit angles (85^0 to 95^0) accounts for such inaccuracy.

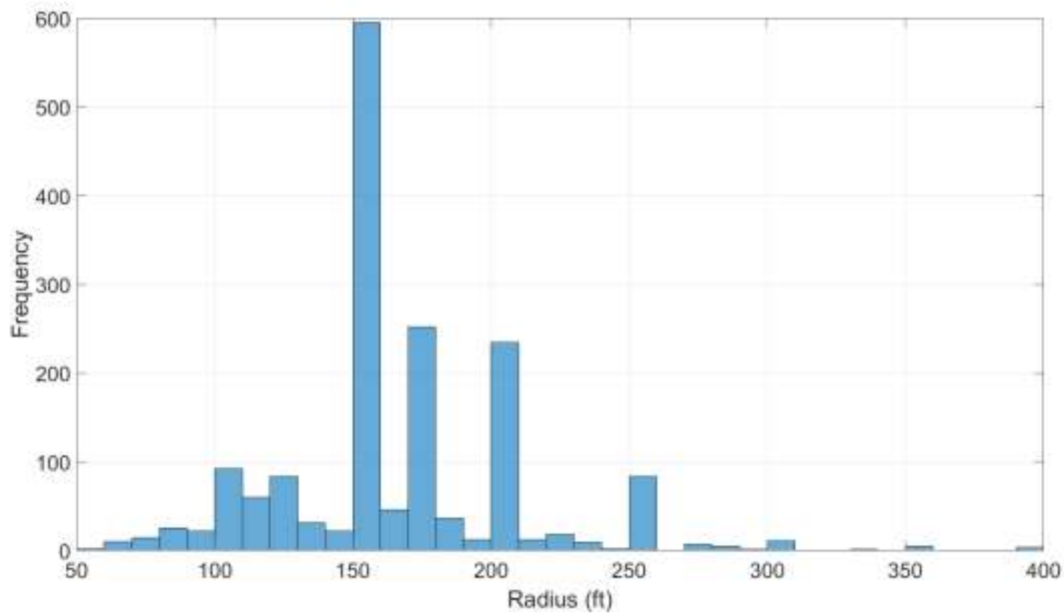


Figure 17 : Histogram of Right-Angle Exit Radius.

The **Figure 17** shows the most common design radius for a right angle exit at airports analyzed is 150 ft. Many right angle exits at DFW have a turn radius of 175 ft due to which there is a hike in the number of right angle exits with 175 ft radius. Similarly, other airports such as ATL, LAS, IAH, MEM, DTW and MSP have right angle exits with a turn radius of 200 ft. MSP is only airport with a significant number (15 records) of right angle exits having a radius of 250 ft.

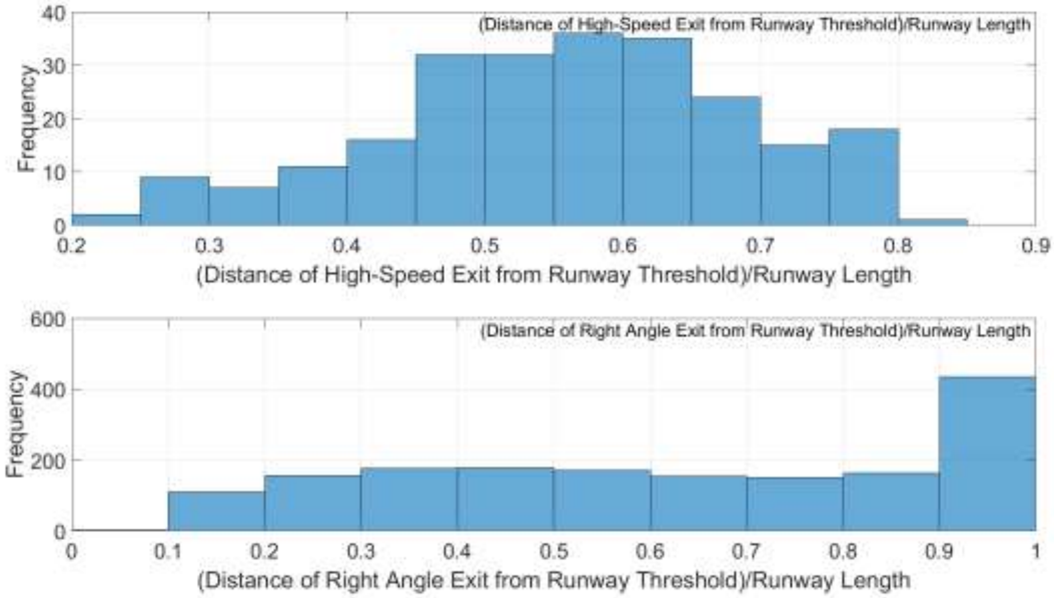


Figure 18 : Histogram of (Distance of Exit from Runway Threshold)/Runway length for High-Speed Exit 'Top'. Histogram of (Distance of Exit from Runway Threshold)/Runway length for Right Angle Exit 'Bottom'.

The database consists information about the distance of exit from the runway threshold. Which is the distance of point of curvature (entry point) of the exit from the runway threshold. Short runways have exits closer to the threshold. Similarly, for long runways, the distance of exit from runway threshold is more as the aircraft tend to land long and have more freedom. However, to observe the trend in the locations of right angle exits along the runways (at all airports) normalization technique is used. Normalizing the distance of right angle exit from the runway threshold with runway length is suitable to observe the trend. As observed in the **Figure 18**, the right angle exits are designed uniformly all over the runway at many airports with many present at the runway end. However, the high-speed exits are located mostly at the center of the runway.

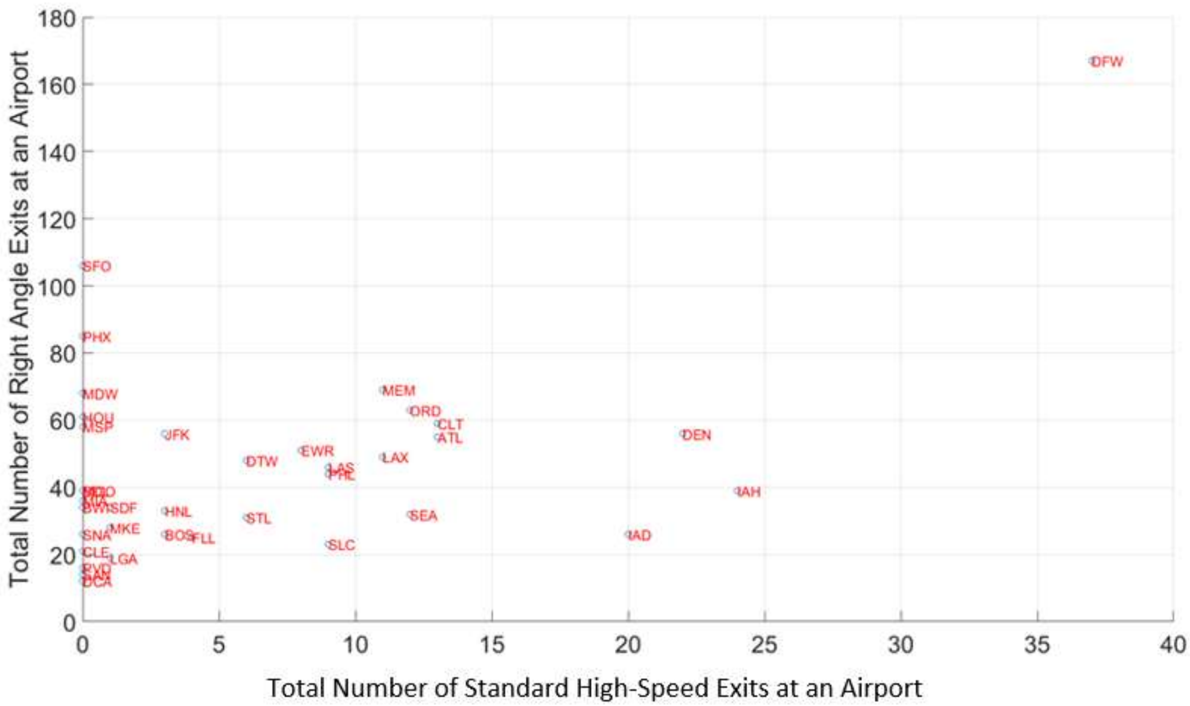


Figure 19: Total Number of Right Angle exits vs Total Number of Standard High-Speed Exits at Airports in the USA.

The state of airports can be studied using the **Figure 19**. The busy commercial airports in the USA are generally the ones with high number of standard high-speed exits and right angle exits. ATL, ORD, DEN, DFW, LAX and others are few such airports.

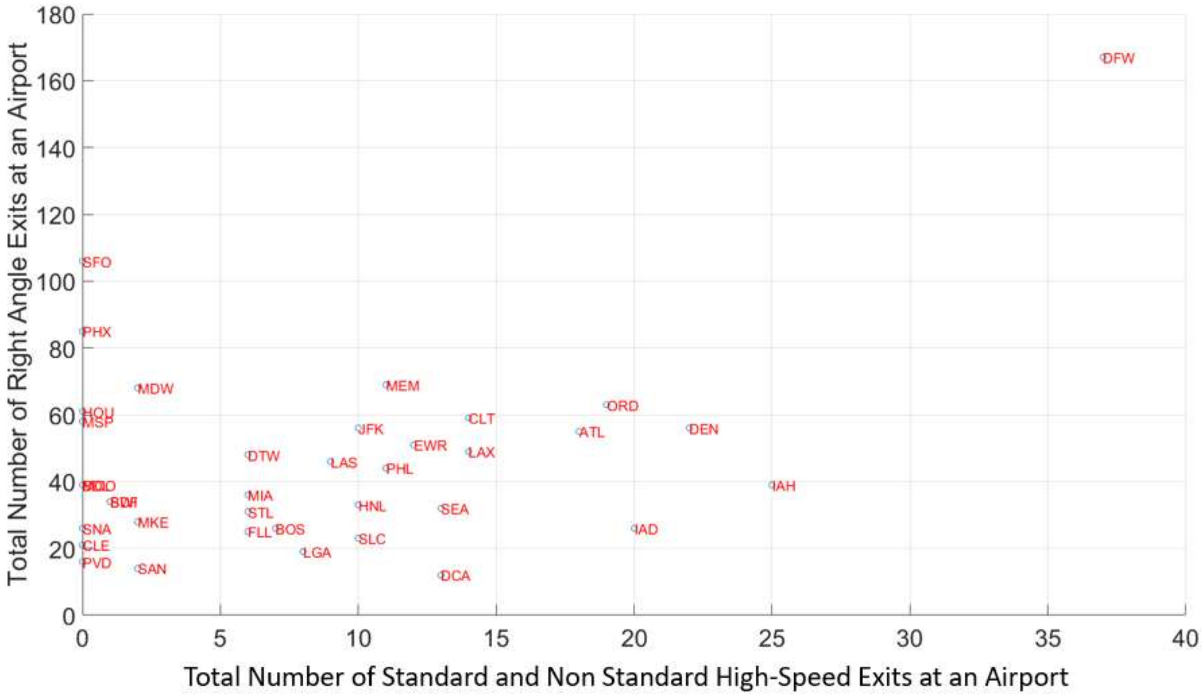


Figure 20 : Total Number of Right Angle Exits vs Total Number of Standard and Non Standard High-Speed Exits at Airports in the USA.

The **Figure *20*** provides a better explanation about the state of airports by including standard and non-standard high-speed exits. Small and busy Airports such as LGA and DCA occupying less areas have many non-standard high-speed exits. Such small airports stand out from the cluster when non-standard high-speed exits are included.

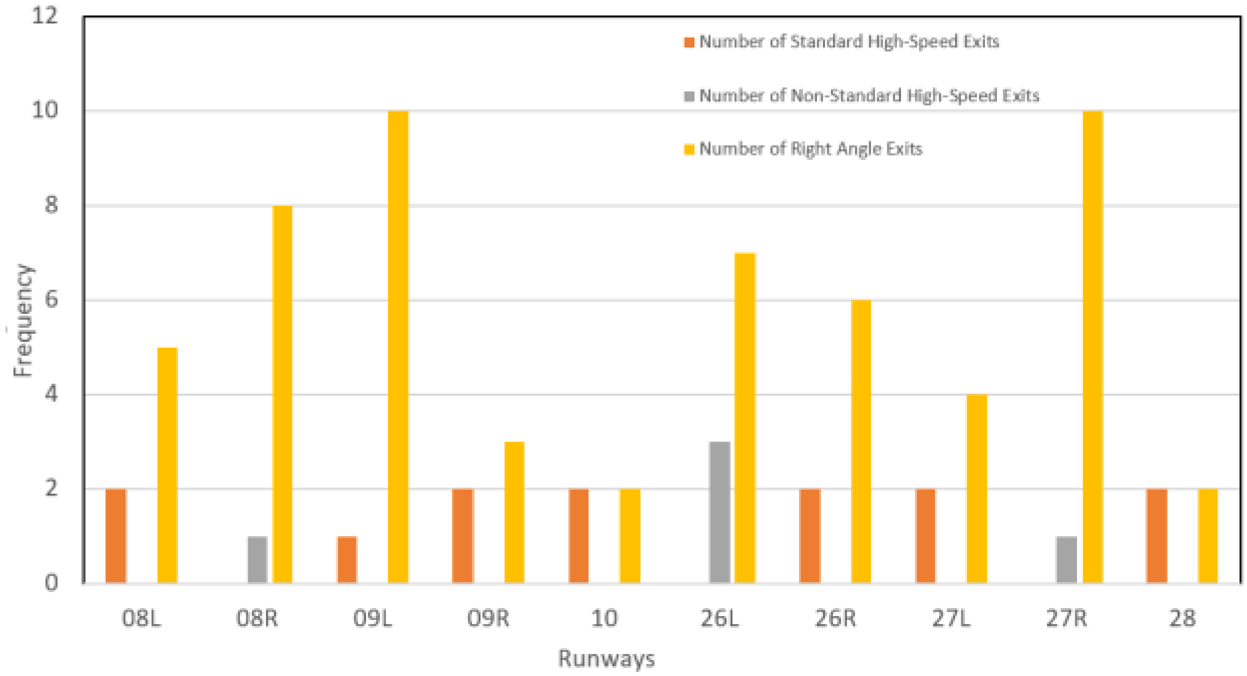


Figure 21 : Number of Right Angle, Non-Standard and Standard High-Speed Exits at Different Runways at ATL.

The analysis extends to runway level at all the airports analyzed. Not all runways at an airport have standard high-speed exits. For example **Figure 21** shows at ATL (has five standard high-speed exits and 13 non-standard high-speed exits) only runways 08L, 09L and 09R have standard high-speed exits. Other runways 26L and 27R have non-standard high-speed exits. All Runways have right-angle exits as observed in the **Figure 21**.

CHAPTER 3: RUNWAY EXIT SPEED MODEL

The runway exit speed model predicts the exit speed at the point of curvature using the exit geometry. Path length, Radius, exit angle and hold bar distance define the exit geometry. ASDE-X data is the source of observed exit speeds at the point of curvature. The following section briefly discusses about the data sources used and the airports analyzed.

3.1 Data Sources Used

Building the runway exit speed models involved using ASDE-X data from fourteen different airports. **Table *3*** shows the airports analyzed. All airports support operations of smaller aircraft belonging to ADG's I, II and III. For large aircraft types, not all facilities are designed to support them.

Table 3 : Airports ASDE-X Data Analyzed for the Exit Speed Model

Airports	Dates Analyzed	ADG I	ADG II	ADG III	ADG IV	ADG V	ADG VI
SFO	2016/01 – 2016/12	✓	✓	✓	✓	✓	✓
SAN	2016/01- 2016/02 2016/07, 2016/09	✓	✓	✓	✓	✓	**NA**
ORD	2016/01- 2016/02 2016/07 – 2016/08	✓	✓	✓	✓	✓	✓
MCO	2016/01- 2016/02 2016/07, 2016/09	✓	✓	✓	✓	✓	**NA**
LGA	2016/01- 2016/02 2016/07, 2016/09	✓	✓	✓	✓	**NA**	**NA**
LAX	2016/01 – 2016/12	✓	✓	✓	✓	✓	✓
IAD	2016/01- 2016/02 2016/07, 2016/09	✓	✓	✓	✓	✓	✓
MIA	2016/01 – 2016/12	✓	✓	✓	✓	✓	✓
DCA	2016/01- 2016/02 2016/07 – 2016/08	✓	✓	✓	✓	**NA**	**NA**
CLT	2016/01 – 2016/12	✓	✓	✓	✓	✓	**NA**
IAH	2016/01 – 2016/08	✓	✓	✓	✓	✓	✓

EWR	2016/01- 2016/02 2016/07	✓	✓	✓	✓	✓	**NA**
SEA	2016/01-2016/08	**NA**	**NA**	**NA**	**NA**	✓	**NA**
JFK	2016/01-2016/12	**NA**	**NA**	**NA**	**NA**	✓	✓

The **Table *4*** shows that small airports such as DCA, SAN and LGA have few to none operations of ADG V and ADG VI aircraft types. At SAN, using four months of ASDE-X data, only 86 operations belonged to ADG V. Similarly, with EWR, only 179 operations belonged to ADG V during the three months of ASDE-X data analyzed. The airports having infrastructure to support big aircraft types are SFO, ORD, LAX, IAD, MIA and IAH. Majority of the operations at all airports belong to ADG III as many domestic aircraft belong to this category.

Table 4: Number of Observations Analyzed from all Airports for Different ADG's

ADG	ADG I	ADG II	ADG III	ADG IV	ADG V	ADG VI
SFO	955	24,689	114,489	12086	7976	765
SAN	419	2,024	16,054	726	86	**NA**
ORD	299	56,560	56,106	4176	6361	840
MCO	868	3,336	40,808	4642	949	**NA**
LGA	117	13,924	43,632	67	**NA**	**NA**
LAX	1226	26,971	229,852	29268	31046	4180
IAD	1607	17,344	17,868	2258	3039	323
MIA	2989	18,246	115,078	31076	14555	369
DCA	20	13,268	20,775	179	**NA**	**NA**
CLT	3697	62,067	181,792	3001	2894	**NA**

IAH	688	49,676	83,531	4882	5929	657
EWR	105	100,026	26,747	7192	179	**NA**
SEA	**NA**	**NA**	**NA**	**NA**	3,624	**NA**
JFK	**NA**	**NA**	**NA**	**NA**	28,983	4,170

The ASDE-X data from these airports provides the observed exit speeds (response variable of the model) at the point of curvature for different exits. Further, the flight track coordinates from ASDE-X provides information about which exit taken. However, obtaining the exit geometry parameters (predictors used in the model) of the exit taken involves using the runway exit database mentioned in the chapter 2. Therefore, the data required for this analysis uses two data sources:

- ASDE-X Data

Runway Exit Database**Table *5*** shows the final data for analysis for ADG VI after parsing the ASDE-X data using MATLAB and associating each operation with the exit from the exit database.

Table 5 : Final Dataset used for Regression Analysis to Determine Mean Exit Speeds for ADG VI

Airport	Runway End	Exit Name	Number of Observations	Observed Mean Exit Speeds (knots)	Exit Angle (degrees)	Path Length (ft)	Radius (ft)	Hold Bar Distance (ft)	Remaining Length Along the Runway (ft)	Runway Length (ft)
EWR	04R	P4	16	32.2	30	894	1800	334	4167	10000
EWR	22L	N	34	46.4	30	1321	1800	396	4607	10000
EWR	22L	V	19	29.2	48	672	800	274	3059	10000
IAD	01R	K2	210	41.3	30	1190	1800	342	4779	11500
IAD	19C	Y7	43	38.2	30	1110	1800	342	4697	11500
IAD	19L	K7	55	36.8	30	1076	1800	340	4666	11500
IAH	08L	FH	72	35.7	30	1038	1800	294	2220	9000
IAH	08L	FJ	30	18.8	90	425	200	291	604	9000
IAH	08L	FK	14	14.0	90	405	200	291	239	9000
IAH	08R	NN	48	35.8	30	1004	1800	295	3056	9402
IAH	08R	NP-R	64	12.8	90	387	200	292	215	9402
IAH	26L	NF	76	36.2	30	998	1800	295	3050	9402
IAH	26L	NR-L	176	11.8	90	365	150	292	831	9402

IAH	26L	NR-R	28	14.3	90	365	150	292	831	9402
IAH	26R	FC	26	20.0	90	427	200	291	604	9000
IAH	26R	FD	75	36.4	30	1025	1800	294	2208	9000
IAH	26R	NE	23	13.5	90	397	200	291	232	9000
JFK	04L	G-L	21	24.3	42	579	500	276	5619	12079
JFK	04R	E	455	22.0	52	694	400	393	392	8400
JFK	04R	FA	80	35.9	20	1271	1400	478	2528	8400
JFK	04R	FB	464	39.2	20	1682	1400	494	2528	8400
JFK	13L	4L	22	15.2	90	352	150	275	3280	10000
JFK	13L	Y-R	66	13.7	90	360	150	274	2479	10000
JFK	13L	ZA-R	99	32.7	45	749	1000	275	4583	10000
JFK	22L	J	1215	41.6	28	1393	1800	481	2596	8400
JFK	22L	Z	777	15.4	63	531	175	394	241	8400
JFK	22R	K2	10	11.0	90	320	100	270	3894	12079
JFK	31L	MC	29	15.0	90	378	175	273	9100	14511
JFK	31L	MD	159	31.1	30	872	1000	318	8595	14511
JFK	31L	PA	683	20.5	90	477	400	271	7396	14511
JFK	31R	CB-L	16	12.7	90	341	150	272	2223	10000
JFK	31R	V-L	23	18.3	90	440	350	272	3604	10000
LAX	06L	V	182	9.7	90	340	150	245	172	8926
LAX	07R	A4	41	15.5	60	422	250	272	2821	11095
LAX	07R	F-R	64	12.9	90	371	175	272	226	11095
LAX	07R	G-R	45	13.4	98	371	150	281	3535	11095
LAX	24R	AA	927	36.9	30	1000	1800	278	2385	8926
LAX	24R	BB	1588	11.3	90	358	150	243	193	8926
LAX	25L	A7	41	46.5	30	1093	1800	272	6067	11095
LAX	25L	N-L	623	13.9	90	354	150	273	3744	11095
LAX	25L	T-L	582	13.9	90	357	150	273	2190	11095
LAX	25L	T-R	29	15.3	90	357	150	273	2190	11095
LAX	25L	U-L	42	15.3	135	464	175	336	352	11095
MIA	08R	L11	38	14.2	90	331	100	271	207	10506
MIA	08R	M10	23	41.6	45	1063	600	290	2832	10506
MIA	9	Q8	49	16.7	127	368	160	273	1962	13016
MIA	9	T5	148	42.3	37	1084	800	276	5356	13016
MIA	9	U	41	44.2	47	1041	600	271	6140	13016
MIA	26L	M2	19	18.4	90	452	220	292	1516	10506
MIA	26L	Z-L	21	46.5	45	1092	600	288	3556	10506
ORD	10C	F-L	40	16.4	90	413	175	291	2788	10801
ORD	10C	F-R	56	15.3	90	413	175	290	2788	10801
ORD	10C	GG-L	16	15.6	90	409	200	290	259	10801
ORD	10C	HH-L	29	16.2	90	403	170	293	1566	10801

ORD	10C	HH-R	44	15.3	90	404	170	291	1566	10801
ORD	10C	P6	42	53.0	30	1270	1800	296	4509	10801
ORD	10L	EE-R	12	13.5	90	400	175	292	3077	13000
ORD	10L	F-R	19	12.5	90	390	160	290	4945	13000
ORD	28C	CC-L	27	15.5	90	379	175	289	1841	10801
ORD	28C	CC-R	100	14.1	90	380	175	293	1841	10801
ORD	28C	DD-L	122	15.3	90	393	173	292	3090	10801
ORD	28C	DD-R	104	15.4	90	392	175	291	3090	10801
ORD	28C	P1	105	50.2	30	1249	1800	295	4739	10801
ORD	28R	DD-L	47	17.4	90	450	190	291	3158	13000
ORD	28R	P4	18	16.1	135	454	165	313	5682	13000
SFO	19L	H	48	39.4	25	1073	1400	301	2829	8650
SFO	28L	D-L	44	17.7	95	349	200	247	3887	11381
SFO	28R	K-R	13	27.4	50	607	550	292	3842	11870
SFO	28R	Q	21	32.9	30	917	1800	281	3536	11870
SFO	28R	T	393	43.0	20	1193	1400	323	5970	11870
SFO	28R	U-R	210	21.8	40	648	550	332	1290	11870

3.2 Challenges with the Data

The ASDE-X data used for this analysis had challenges with few exits mentioned ahead. These challenges include:

- Bimodal exit angle distribution
- Right angle exit with long path length

Bimodality in the exit angle distribution occurs when pilots use a runway exit at different exit angles. Exits such as N1 on runway 01 at DCA and A1-L on runway 27L at ORD are examples of bimodal exit angle distribution.

Exit N1 on runway 01 at DCA is a right angle exit according to the runway exit database. However, when looked into the ASDE-X data, many observations show that pilots take this exit at 45^0 due to its wide throat area. This causes inconsistency in the data. The exit speeds at which a 45^0 exit is taken is much higher to 90^0 turn. When training the model with this data, it realizes that the exit speeds at N1-01 are higher than usual right angle exits. However, it tries to justify this behavior at N1-01 by lowering the differences between predicted and observed speeds. This isn't good for the model.

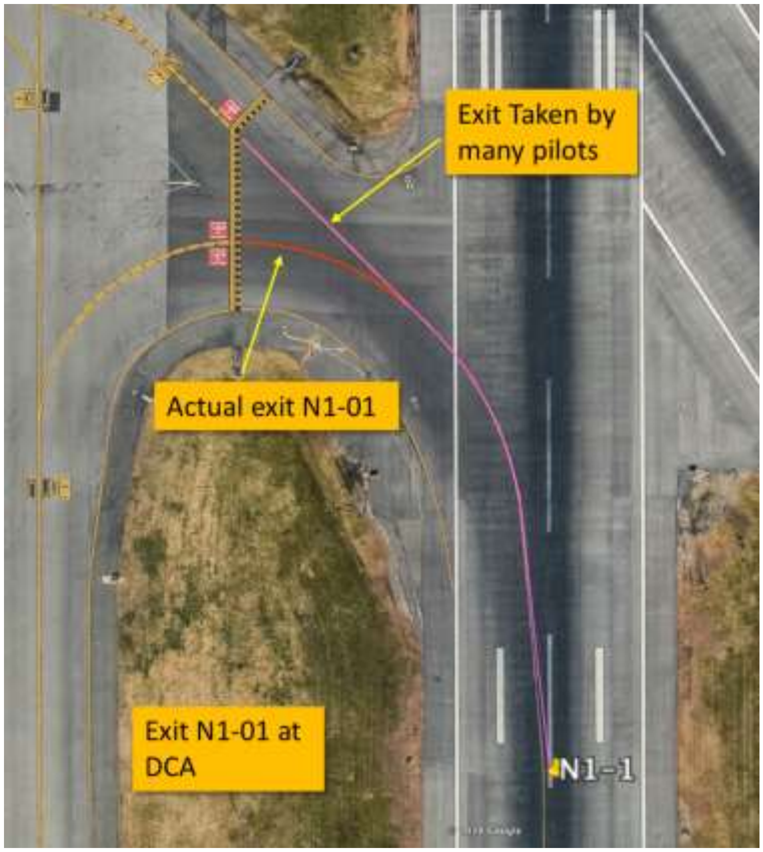


Figure 22 : Exit N1-01 at DCA is a Right-Angle Exit. However, Pilots also use it as a 45^0 Exit.

This eventually makes the model predict higher exit speeds for the right angle exits. Therefore, solving this problem involved removing all the observations with 45^0 exit.

Other challenge with the data includes right angle exits with long path length. Exit C6-B on runway 18L at CLT is one such exit having longer path length compared to a typical right angle exit. **Figure *23*** shows another exit, C6-B on runway 18L at CLT. It is a right angle exit in the database. However, this exit is a non-standard right angle exit. With a path length of 826 ft, this exit has high exit speeds compared to the typical right angle exits. Therefore, considering this exit would make the model predict high exit speeds for right angle exits. By removing the records of this exit, the fidelity of the model improves.

Figure 23 : Exit C6-B on Runway 18L at CLT is a Non-Standard Right-Angle Exit with Long Path Length.

Similarly, **Figure *24*** shows another exit A1-L. It is a high-speed exit on runway 27L at ORD. However, sometimes pilots use this exit to take a right angle exit or a back turn based on the location of their terminal. Due to this, the behavior of the pilots taking this exit is different sometimes. Those pilots who take a right angle exit tend to take the exit at a lower speed. Accounting for such behavior is difficult. The model upon looking into the data at A1-L realizes the exit speed at a high-speed exit is not high. Moreover, it tries to justify this behavior by minimizing the difference between the observed and predicted speeds at A1-L. This reduces the predicted speed for a typical high-speed exit. Therefore, removing the observations at A1-L with exit angles equal to and greater than 90^0 can increase the fidelity of the model.

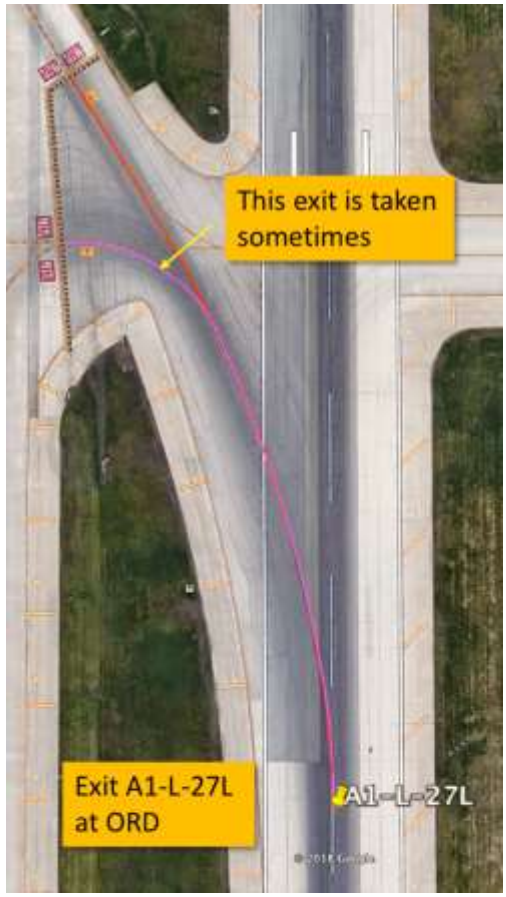


Figure 24 : Exit A1-L on Runway 27L at ORD is a Standard High-Speed Exit. However, Pilots also use it as a Right-Angle Exit.

3.3 Runway Exit Speed Models

The speed at which a pilot takes an exit is dependent on the type of exit taken. A pilot can take a high-speed exit at an average speed of 42 knots. Meanwhile, a right angle exit at an average speed of 20 knots. The time taken for a pilot to manuever the exit from point of curvature to hold bar is dependent on the speed he/she manuevers. Therefore, to predict the time taken to manuever the exit, exit speed at the point of curvature is necessary. The runway exit model predicts the exit speed at the point of curvature based on the geometry associated with the exit. For example, the exit with low turning radius contraints the pilot from using the exit at higher speeds. Similarly, if the exit angle is high, the pilot has less freedom to take the exit at a higher speed. Therefore, factors determining the speed at which pilot takes an exit are :

- Path length (ft)

- Radius (ft)
- Exit Angle (ft)
- Hold bar distance (ft)
- (Remaining length along the runway)/Runway Length

Path Length, Radius and Exit Angle are strong predictors while hold bar distance is a weak predictor.

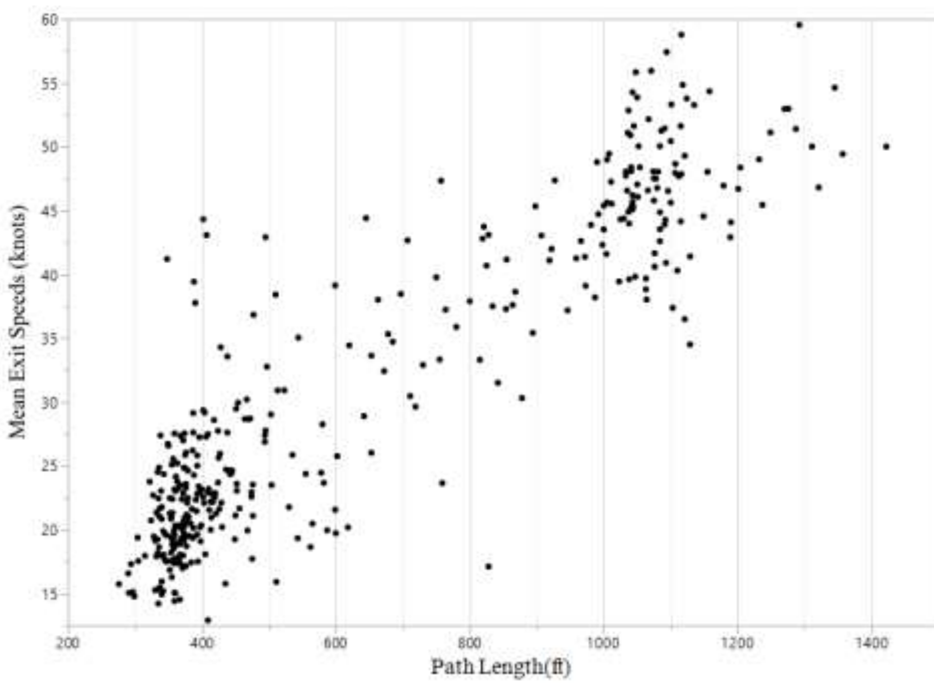


Figure 25 : Mean Exit Speed (knots) vs Path Length (ft) for ADG III

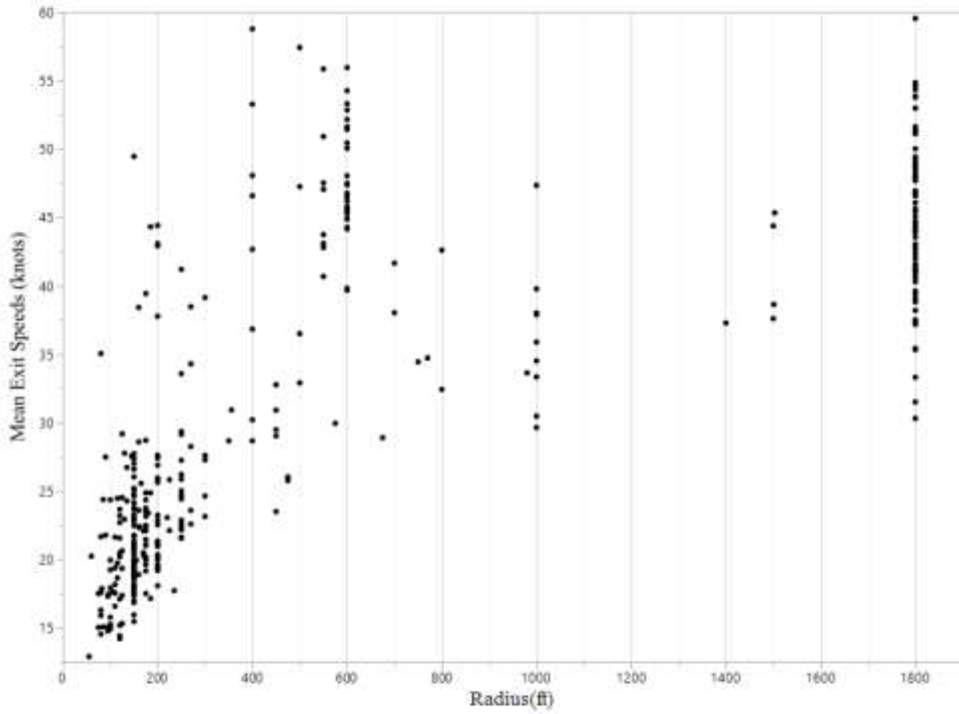


Figure 26 : Mean Exit Speed (knots) vs Radius (ft) for ADG III

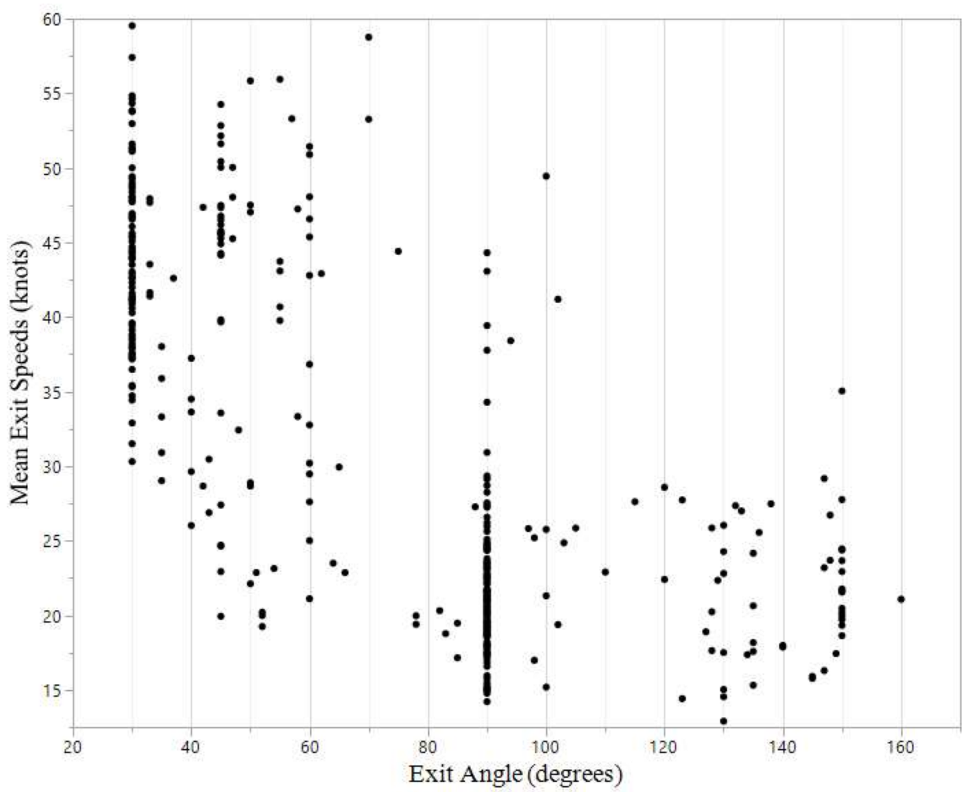


Figure 27 : Mean Exit Speed (knots) vs Exit Angle (degrees) for ADG III

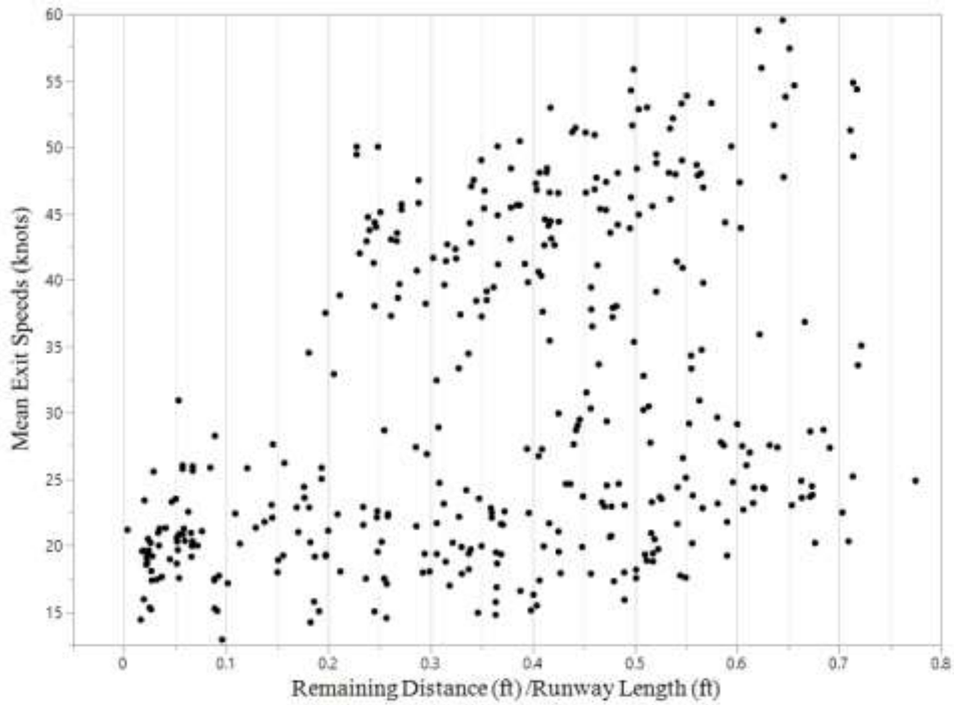


Figure 28 : Mean Exit Speed (knots) vs (Remaining Distance along runway (ft)/Runway Length (ft)) for ADG III

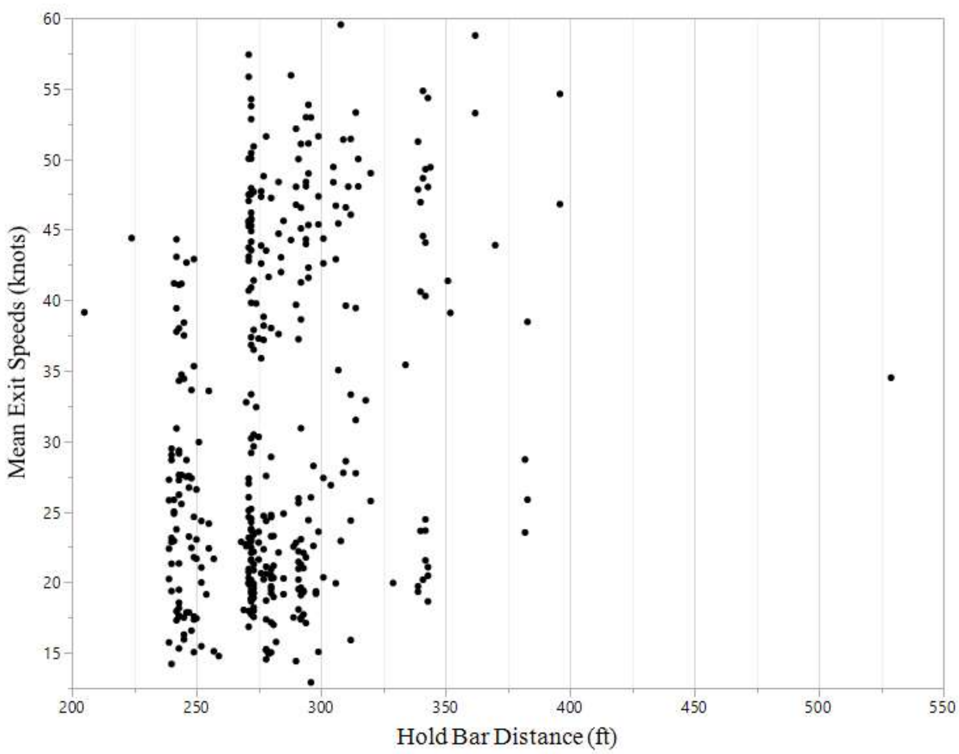


Figure 29 : Mean Exit Speed (knots) vs Hold Bar Distance (feet) for ADG III

Each marker in the plots above represent an exit. Many exits have a standard for hold bar distance due to which high correlation is not observed. Nevertheless, exit speeds and hold bar distance have a weak correlation. The exits with hold bar closer to runway centerline tend to have lower exit speeds and vice versa. The outlier ***Figure 29*** shows having a hold bar distance of 527 ft is exit Y5 on runway 4R at ORD. ***Figure 30*** shows this exit is a non-standard exit (an exit whose design is not mentioned in FAA advisory circular). The reason for such large hold bar distance is due to the presence of other runway 28R nearby. Many airports in USA have non-standard exits due to such reasons.

Figure 30 : Exit Y5 on Runway 4R at ORD Having Large Hold Bar Distance.

Tables below show the correlation between exit speed and predictors for different ADG’s.

Table 6 : Correlation Matrix for ADG I

ADG's	Mean Exit Speed (knots)	Path Length (ft)	Exit Angle (degrees)	Radius (ft)	Hold Bar Distance (ft)	Remaining Length/Runway Length
Mean Exit Speed (knots)	1	0.92	-0.8	0.85	0.41	0.07
Path Length (ft)	0.92	1	-0.74	0.87	0.56	-0.06
Exit Angle (degrees)	-0.8	-0.74	1	-0.78	-0.19	0.17
Radius (ft)	0.85	0.87	-0.78	1	0.52	-0.05
Hold Bar Distance (ft)	0.41	0.56	-0.19	0.52	1	0.09
Remaining Length/Runway Length	0.07	-0.06	0.17	-0.05	0.09	1

Table 7 : Correlation Matrix for ADG II

ADG's	Mean Exit Speed (knots)	Path Length (ft)	Exit Angle (degrees)	Radius (ft)	Hold Bar Distance (ft)	Remaining Length/Runway Length
Mean Exit Speed (knots)	1	0.92	-0.75	0.81	0.3	0.13
Path Length (ft)	0.92	1	-0.71	0.83	0.47	0.08
Exit Angle (degrees)	-0.75	-0.71	1	-0.77	-0.11	0.08
Radius (ft)	0.81	0.83	-0.77	1	0.34	0.05
Hold Bar Distance (ft)	0.30	0.47	-0.11	0.34	1	0.14
Remaining Length/Runway Length	0.13	0.08	0.08	0.05	0.14	1

Table 8 : Correlation Matrix for ADG III

ADG's	Mean Exit Speed (knots)	Path Length (ft)	Exit Angle (degrees)	Radius (ft)	Hold Bar Distance (ft)	Remaining Length/Runway Length
Mean Exit Speed (knots)	1	0.89	-0.71	0.72	0.25	0.40
Path Length (ft)	0.89	1	-0.72	0.82	0.45	0.27
Exit Angle (degrees)	-0.71	-0.72	1	-0.76	-0.13	-0.13
Radius (ft)	0.72	0.82	-0.76	1	0.34	0.22
Hold Bar Distance (ft)	0.25	0.45	-0.13	0.34	1	0.09
Remaining Length/Runway Length	0.40	0.27	-0.13	0.22	0.09	1

Table 9 : Correlation Matrix for ADG IV

ADG's	Mean Exit Speed (knots)	Path Length (ft)	Exit Angle (degrees)	Radius (ft)	Hold Bar Distance (ft)	Remaining Length/Runway Length
Mean Exit Speed (knots)	1	0.93	-0.73	0.67	0.21	0.47
Path Length (ft)	0.93	1	-0.76	0.79	0.38	0.38
Exit Angle (degrees)	-0.73	-0.76	1	-0.77	-0.10	-0.25
Radius (ft)	0.67	0.79	-0.77	1	0.31	0.30
Hold Bar Distance (ft)	0.21	0.38	-0.10	0.31	1	0.14
Remaining Length/Runway Length	0.47	0.38	-0.25	0.30	0.14	1

Table 10 : Correlation Matrix for ADG V

ADG's	Mean Exit Speed (knots)	Path Length (ft)	Exit Angle (degrees)	Radius (ft)	Hold Bar Distance (ft)	Remaining Length/Runway Length
Mean Exit Speed (knots)	1	0.94	-0.78	0.78	0.25	0.53
Path Length (ft)	0.94	1	-0.83	0.83	0.41	0.44
Exit Angle (degrees)	-0.78	-0.83	1	-0.82	-0.27	-0.41
Radius (ft)	0.78	0.83	-0.82	1	0.28	0.42
Hold Bar Distance (ft)	0.25	0.41	-0.27	0.28	1	0.04
Remaining Length/Runway Length	0.53	0.44	-0.41	0.42	0.04	1

Table 11 : Correlation Matrix for ADG VI

ADG's	Mean Exit Speed (knots)	Path Length (ft)	Exit Angle (degrees)	Radius (ft)	Hold Bar Distance (ft)	Remaining Length/Runway Length
Mean Exit Speed (knots)	1	0.94	-0.86	0.87	0.36	0.57
Path Length (ft)	0.94	1	-0.88	0.89	0.58	0.49
Exit Angle (degrees)	-0.86	-0.88	1	-0.87	-0.43	-0.47
Radius (ft)	0.87	0.89	-0.87	1	0.41	0.49
Hold Bar Distance (ft)	0.36	0.58	-0.43	0.41	1	0.05
Remaining Length/Runway Length	0.57	0.49	-0.47	0.49	0.05	1

Upon close observation, one can observe higher correlation among the predictor variables and response variable in ADG VI. This can be attributed to lower count in the data for ADG VI. Moreover, the explanatory variable $\frac{\text{Remaining length along runway}}{\text{Runway length}}$ is not a strong predictor for ADG I and II. This could be due to exiting the runways early by using the first exits. This is explainable behavior because most of these small aircraft reach exit speeds earlier than the bigger aircraft.

Table 12 : Number of Exits Taken by Aircraft Belonging to Different ADG's from Airports Analyzed

ADG's	All Exits	High-Speed Exits	Right Angle Exits	Wide Throat Exits
ADG I	109	33	29	7
ADG II	362	109	57	34
ADG III	397	144	58	43
ADG IV	276	85	51	43
ADG V	232	80	48	26
ADG VI	71	34	15	3

The **Table *12*** shows that high-speed exits are highly taken exits across all ADG's followed by the right angle exits. The remaining exits belong to the non-standard exits category. However, looking into the number of observations at each exit type, ADG VI use right angle exits more frequently than the high-speed exits.

Three types of runway exit models are tried in predicting the exit speed at the point of curvature. They are as follows:

- Generic Multi-Linear Regression Model
- Quadratic Model
- Artificial Neural Networks Model

3.3.1 Generic Multi-Linear Regression Model

A Multi-linear regression tries to model the relationship between two or more predictors and a response variable. The model attempts this by fitting a linear equation to the observed data. At the point of curvature, the exit speeds of aicrafts belonging to different Aircraft Design Groups (ADG) is different. Aircrafts belonging to Aircraft Design Group V (ADG V) and Aircraft Design Group VI (ADG VI) tend to take more right angle exits than aicrafts belonging to Aircraft Design Group II (ADG II) and Aircraft Design Group (ADG III). This can be observed in **Figure 31**. Due to this the exit speed distributions of aircraft belonging to ADG V and ADG VI is more bimodal compared to ADG II and ADG III. Accounting for such different exit speed behavior requires developing distinct exit speed models for each ADG. Therefore, data belonging to each ADG is separately analyzed.

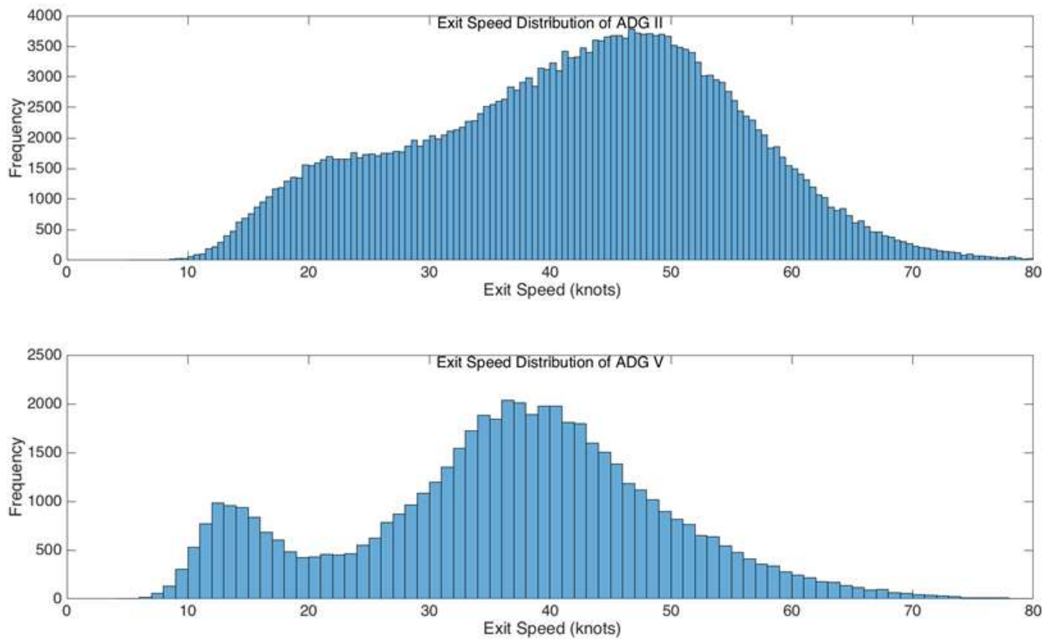


Figure 31 : Exit speed distribution of ADG V has More Bimodality than ADG II.

The correlation matrices for different ADG's highlight correlation between response variable and explanatory variables. Also, strong and moderate correlation is found among the explanatory variables. Accounting for such correlation among explanatory variables and avoiding multicollinearity involves use of interaction terms in the multi-linear regression model. The explanatory variables are also standardized because all don't have common units. Path length, radius and hold bar distance have common units. However, when using exit angle, the variables need to be standardized.

The model is valid within the bounds of the data used for the analysis. It is not assured that model performs well when input parameters are beyond the bounds of the data used. The multi-linear regression model is the weighed model where weighing factor is the number of observations.

Table 13 : Bounds of the Data used for Different ADG's

ADG's	Path Length (ft)		Radius (ft)		Hold Bar Distance (ft)		Exit Angle (degrees)		Remaining Length along Runway/Runway Length	
	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound
ADG I	290	1422	55	1800	240	396	30	150	0.022	0.78
ADG II	276	1422	55	1800	205	529	20	150	0.003	0.89
ADG III	276	1422	55	1800	205	529	30	160	0.003	0.77
ADG IV	299	1422	75	1800	205	529	20	150	0.006	0.69
ADG V	299	1682	75	1800	240	529	20	150	0.016	0.72
ADG VI	320	1682	100	1800	243	494	20	135	0.019	0.63

Observing **Table *13***, one can notice that the lower bound for radius increasing with increasing ADG's. The data shows that larger aircraft cannot maneuver exits with lower turn radius. This is noticeable with ADG's IV, V and VI. Similarly, the minimum hold bar distance increases for the larger aircraft.

The functional form of the generic multi-linear regression model is as follows :

$$\text{Mean Exit Speed (knots)}$$

$$= \alpha + \alpha_{90}*\beta_{90} + \alpha_{wide\ throat}*\beta_{wide\ throat} + \alpha_1*(\text{Standardized}\left(\frac{Path\ Length}{Hold\ Bar\ Distance}\right)) + \alpha_2*(\text{Standardized}\left(\frac{Radius}{Exit\ Angle}\right)) + \alpha_3*\left(\frac{Remaining\ Length\ along\ Runway}{Runway\ Length}\right) + \alpha_4*(\text{Standardized(Exit Angle)}) + \alpha_5*(\text{Standardized(Path Length)}) + \alpha_6*(\text{Standardized}\left(\frac{Path\ Length}{Radius}\right))$$

Where, all dimensions are in feet and exit angle in degrees. β_{90} and $\beta_{\text{wide throat}}$ are exit speed intercepts for right angle exits, obtuse angle exits and wide throat exits.

$$\beta_{90} = \begin{cases} 1, if\ 85 < Exit\ Angle < 95 \\ 0,\ \ Other\ Cases \end{cases}$$

$$\beta_{\text{wide throat}} = \begin{cases} 1, if\ 45 \leq Exit\ Angle\ and\ Path\ Length > 800\ ft \\ 0,\ \ Other\ Cases \end{cases}$$

Table 14 : Coefficients of Predictors Used in Multi-Linear Regression Model

ADG's	α	α_{90}	$\alpha_{\text{Wide Throat}}$	α_1	α_2	α_3	α_4	α_5	α_6
ADG I	24.5	0	0	7.62	0	11.63	-1.86	0	0
ADG II	29.3	0	0	8.15	2.18	6.41	0	0	0
ADG III	27.1	0	0	9.38	0	10.52	-1.01	0	0
ADG IV	33.1	-5.0	0	0	0	0	0	10.83	1.38
ADG V	22.8	0	0	8.16	2.27	16.71	0	0	0
ADG VI	22.0	-5.7	17.9	0	7.65	16.08	0	0	0

And, Standardized$\left(\frac{Path\ Length}{Hold\ Bar\ Distance}\right) = \frac{\frac{Path\ Length}{Hold\ Bar\ Distance} - \text{mean}\left(\frac{Path\ Length}{Hold\ Bar\ Distance}\right)}{standard\ Deviation\left(\frac{Path\ Length}{Hold\ Bar\ Distance}\right)}$. This definition holds good for other explanatory variables used in the model.

Table 15 : Mean and Standard Deviation of Explanatory Variables in the Models Used.

ADG's	$\frac{Path\ Length}{Hold\ Bar\ Distance}$		$\frac{Radius}{Exit\ Angle}$		Exit Angle		Path Length		$\frac{Path\ Length}{Radius}$	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
ADG I	2.45	1.08	NA	NA	64.8	38.2	NA	NA	NA	NA
ADG II	2.24	1.03	17.68	23.52	NA	NA	NA	NA	NA	NA
ADG III	2.21	1.025	NA	NA	74.7	36.4	NA	NA	NA	NA
ADG IV	NA	NA	NA	NA	NA	NA	695.7	332.7	1.817	1.06
ADG V	2.38	1.09	21.2	24.97	NA	NA	NA	NA	NA	NA
ADG VI	NA	NA	20.14	25.94	NA	NA	NA	NA	NA	NA

Besides these explanatory variables, adding others didn't improve the model. Moreover, due to the addition of other explanatory variables, the model parameter estimates inflated due to correlation among the explanatory variables. This phenomenon is often termed as multi-collinearity. Not including all variables and only considering important variables can help avoid such issues. Path length and radius are two important explanatory variables used among all others. However, including radius as an explanatory variable also inflated the parameter estimates in the model. To avoid this from happening, the model used an interaction term between path length and radius $\frac{Path\ Length}{Radius}$. Similarly, Exit angle has correlation with radius. To avoid correlation issues, an interactive term $\frac{Radius}{Exit\ Angle}$ is used.

On exploring the data, the average exit speed of wide throat exits is found to be higher to average exit speed of entire data. Similarly, the average exit speed of right angle exits is found to be lower to average exit speed of entire data. These differences are accounted for by introducing intercepts for these type of exits in the model. The intercept $\beta_{wide\ throat}$ is positive due to higher average exit speed on wide throat exits. Similarly, β_{90} negative due to lower average exit speeds at right angle exits.

However, the model predicts only mean exit speed for an exit. This implies, the model gives only one exit speed for an exit. In reality, this isn't the case. During every observation, pilots takes the same exit at different speeds. To emulate such behavior, standard deviation is added in model.

The **Figure 32** shows the mean predicted exit speeds observed at every exit against mean observed exit speeds observed at the corresponding exit. In this plot, each dot corresponds to one exit. Careful analysis leads to a simple conclusion that the data is clustered into two groups. The cluster to the right belongs to the high-speed exits and the one to the left belongs to the right-angle exits. Also, the mean exit speeds observed at the high-speed exits is lower than the mean exit speeds observed at the right angle exits. Section **3.7 Validation Plots** shows plots for other ADG's

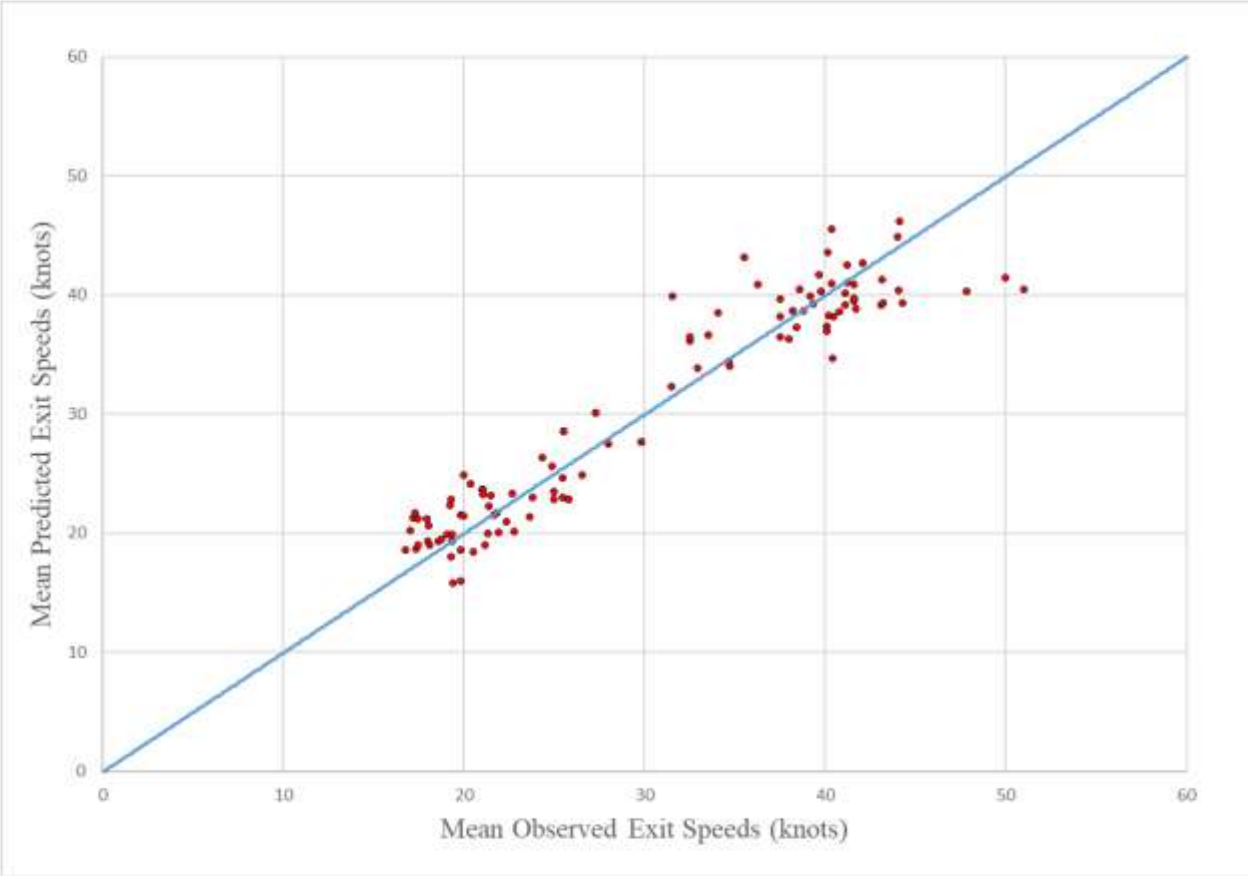


Figure 32 : Mean Predicted Exit Speeds (knots) vs Mean Observed Exit Speeds (knots) for ADG I using Multi-Linear Regression Model

3.3.2 Quadratic Model

The quadratic model uses path length as the only explanatory variable. Trying models with a higher order (Ex: cubic or quartic) of path length didn't yield better results. The only downside of this model is it predicts lower exit speeds for path length greater than 1500 ft. At airports such as DEN where the path lengths of high-speed exits are greater than 1500 ft, the model predicts lower exit speeds. This is contradictory to the general trend that exit speed increases with the path length. In these cases, the linear model does a better work in predicting the exit speed compared to the quadratic model.

The functional form of the quadratic model is as follows:

$$\textit{Mean Exit Speed (knots)} = \alpha + \alpha_1 * \textit{Path Length} + \alpha_2 * (\textit{Path Length})^2$$

Table 16 shows the coefficients of the predictors used in the quadratic model.

Table 16 : Coefficients of Predictors Used in Quadratic Model

ADG's	α	α_1	α_2
ADG I	8.93	0.034	-5.524e-6
ADG II	5.30	0.048	-9.669e-6
ADG III	5.04	0.049	-1.01e-05
ADG IV	0.37	0.055	-1.246e-05
ADG V	-6.27	0.068	-2.111e-05
ADG VI	-8.78	0.0691	-2.228e-05

3.3.3 Artificial Neural Networks Model

A neural network is designed to simulate the way a human brain processes information. ANNs are computer programs, inspired biologically to gather their knowledge by detecting the relationships and patterns in the data and learn (or are trained) through experience rather than programming [13]. The neural network modeling involves two steps. Firstly, the model trains with the data. In our analysis, training phase uses 67% of the data. During this phase, the model adjusts according to its error. The second phase is the validation phase. Our analysis uses 33% of the data during this phase. In this model, the inputs are real numbers and the outputs of each neuron are computed using a non-linear function.

Additionally, the network uses 10 hidden neurons. The number of neurons can be changed according to the convenience. However, increasing the number of hidden neurons beyond 10 didn't yield better results. Further, adding additional hidden layer creates more non linearity to the model. This didn't yield better results. Hence, the model uses 10 hidden neurons in our analysis. The inputs for the neural network are path length, radius and exit angle. The neurons use sigmoid transfer function. However, the user has other preferences for transfer functions involving either linear or Gaussian functions. A squared penalty method is used in this analysis. Also, there are many other penalty methods using absolute difference between observed and predicted exit speeds as the penalty.

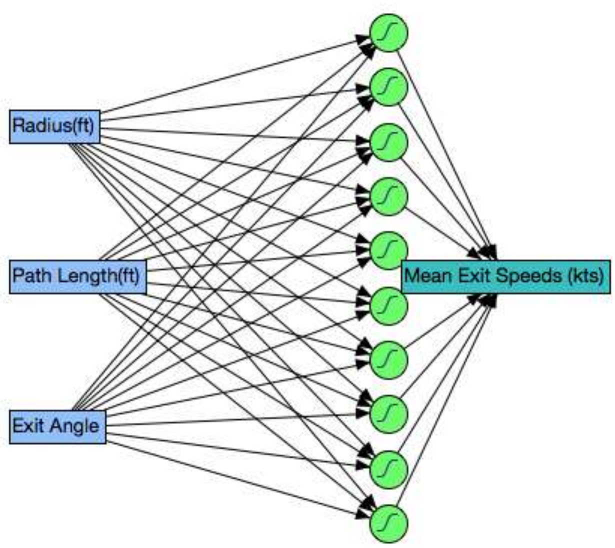


Figure 33: The Artificial Network has 10 Hidden Neurons and 3 Inputs.

Figure 33 shows the neural network has 10 hidden neurons with three inputs. The three inputs for this model include path length, radius and exit angle. The output of this model is the mean exit speed at the point of curvature for an exit.

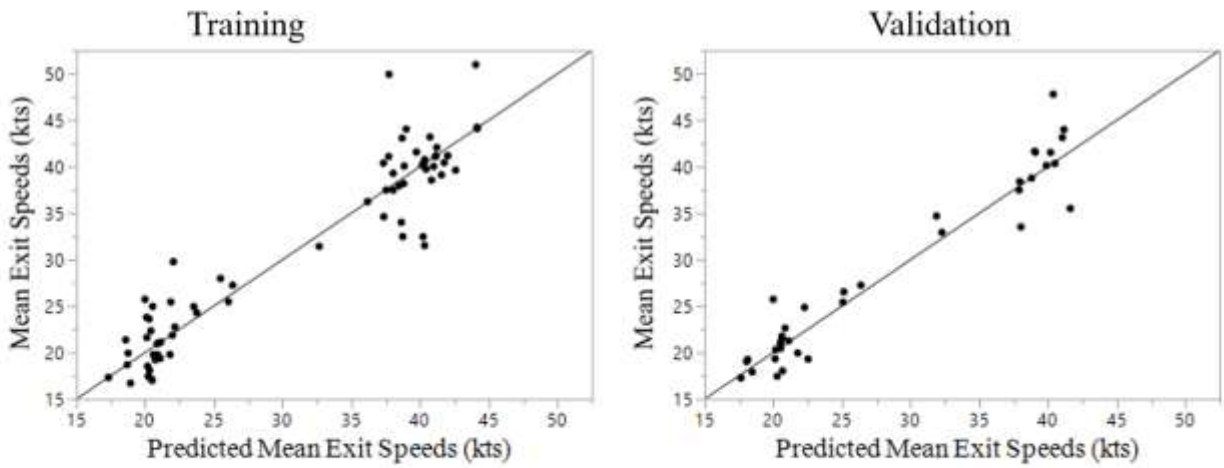


Figure 34 : Training and Validation Plots for Observed and Predicted Mean Exit Speeds at the Point of Curvature of an Exit for ADG I.

The **Figure** ***34*** shows that the training and validation plots of mean exit speeds are along the 45^0 line (Y=X). Section **3.7 Validation Plots** show plots for other ADG's.

3.4 Standard Deviation of Exit Speeds

The mean exit speeds predicted by the model are deterministic. They consider only exit geometry parameters in predicting the speed at the point of curvature. The exit geometry parameters of a particular exit don't change with time. This implies the exit speed predicted is deterministic because it gives only one exit speed value. However, in reality the pilots take the same exit at different speeds as mentioned previously. To imitate such a behavior, we include standard deviation in the model.

The data is the source to extract standard deviation in exit speeds. On Observing the data, the standard deviation of different exit types is found to be different. For example, the standard deviation of a right angle exit is usually lower to a high-speed exit. Similarly, the standard deviation of a wide throat exit is higher to the right angle exit. Accounting for such differences involves extracting standard deviation for each exit type.

3.5 Methodology to Extract Standard Deviation of Exit Speeds

The first step is parsing the data into different ADG's. As mentioned previously, this analysis uses different models for different ADG's. Similarly, on observing the data, the standard deviation of exit speeds for aircraft types belonging to different ADG's is not the same.

Extracting standard deviation involves following these steps:

a) Make a list of all the exits taken by the aircraft belonging to the same ADG. This list contains information about all the exits including the ones having non-normal exit speed distribution. It contains a field for normality check of exit speed which is equal to zero when the exit speed is a normal distribution and equal to one when the exit speed is a non-normal distribution. The threshold for a minimum number of observations at any exit is equal to 15. For exits with a minimum number of observations lower than 15 are not considered for further analysis. Flagging these exits as '2' under normality check for exit speeds removes these exits from further analysis. The threshold for a minimum number of observations changes with the ADG analyzed. For example, ADG III having the highest number of observations has a higher threshold for minimum number of observations (100). Similarly, ADG IV having the lowest number of observations has a minimum threshold for the number of observations (15). 30 is the minimum number of observations required to qualify a sample for normality test. However, upon trying this, not enough samples remained for extracting standard deviation in case of ADG VI. Therefore, the minimum threshold is reduced to 15 for ADG VI. Similarly, the normality check for exit speed is equal to '0' when the number of observations are greater than the minimum threshold and the exit speed distribution passes the normality test. The normality test used in this case is the kstest (one-sample Kolmogorov-Smirnov test).

b) The second step involves putting together all the exits having a minimum number of observations and passed the normality test. This step further segregates these exits into different types such as high-speed exit, right angle exit, wide throat exit, obtuse exit and non-standard exit. Exits with path length greater than 1000 ft and exit angle between 25^0 to 35^0 are high-speed exits. Similarly, exits with exit angles between 85^0 and 95^0 are right angle exits. For wide throat exits, path lengths are greater than 800 ft and the exit angles are greater than 50^0. Non-standard exits are the ones which don't satisfy the criteria for any other type of exit.

c) Finding the standard deviation for the different types of exits is the next step. In case of a high-speed exit, the standard deviation of exit speeds tend to change with the relative location of the high-speed exit along the runway. In cases when the high-speed exit is closer to the runway threshold, the standard deviation is high. Similarly, when the high-speed exit is farther away from the runway threshold, the standard deviation is low. Therefore, for high-speed exits, the standard deviation in exit speed also considers the relative location of the exit from the runway threshold. For other exit types, the standard deviation is independent of the location of the exit.

d) For right angle exits, the model takes highest standard deviation of the exit speeds among the right angle exits. Moreover, it should even satisfy the criteria for minimum number of observations which is equal to 100.

e) For obtuse angle exits, the model takes the standard deviation of the exit with exit angle greater than or equal to 130^0. In this case, there is no criteria for a minimum number of observations as there are few to none exits with number of observations greater than 100.

Table 17 : Variation in Standard Deviation of Exit Speeds for a High-Speed Exit Located Differently Along the Runway (ADG I).

$\frac{\text{Remaining length along the runway}}{\text{Runway length}}$	ADG I	ADG II	ADG III	ADG IV	ADG V	ADG VI
Greater than 0.5	9.6	11.5	10.5	11.3	11.0	9.8
Between 0.5 and 0.35	8.6	9.9	8.9	9.6	9.4	9.8
Less than 0.35	8.3	9.1	8.0	8.9	9.1	8.9

Table *17* shows the standard deviation of high-speed exits located closer to the runway threshold is higher compared to that of high-speed exits located away from the threshold. Many pilots taking the first high-speed exits tend to take it at relatively higher exit speeds. However, few other pilots tend to brake hard and take the first high-speed exits at lower speeds. This could be the reason for high standard deviation of exit speeds at high-speed exits closer to the runway threshold. By the time many pilots reach runway end, they tend to have consistently lower exit speeds and behave similarly. This is the reason for lower standard deviation in exit speeds at high-speed exits closer to the runway end.

In case of ADG VI analysis, the data is not sufficient enough to calculate standard deviation of exit speeds at high-speed exits placed at different locations along the runway. Therefore, as **Table *17*** shows the standard deviation of exit speeds at two different location categories (greater than 0.5, between 0.5 and 0.35) for a high-speed exit is same for ADG VI group.

Table 18 : Standard Deviation of Exit Speeds for Different Exit Types.

Exit Type	ADG I	ADG II	ADG III	ADG IV	ADG V	ADG VI
Right Angle Exit	6.0	5.8	5.1	4.7	3.9	3.9
Wide Throat Exit	10.5	10.5	10.5	11.4	9.6	4.1
Non-Standard Exit	6.9	5.4	9.4	3.7	5.2	4.1
Obtuse Angle Exit	4.5	4.3	4.2	4.4	4.7	3.5

Moreover, the number of observations at wide throat exits is also low for ADG VI. Therefore, the standard deviation of exit speeds obtained for wide throat exits is not reliable for ADG VI. Therefore, the standard deviation for a wide throat exit is equal to standard deviation of a non-standard exit.

The **Table *18*** shows the exit speeds observed at wide throat exits is dispersed highly. Due to this, the standard deviation of exit speeds observed at wide throat exits is higher compared to other exit types. For, right angle exits, and obtuse angle exits, the standard deviation is lower as the pilots show consistently similar behavior is using these exits. The turn radius of right angle exits confines pilots from taking the right-angle exits at higher exit speeds. This also reduces the variability in the exit speeds observed at the right angle exits and is the reason for lower standard deviation in exit speeds at the right angle exits.

3.6 Results

The R^2 fit values of all model types are in the same range for different ADG's. However, the neural networks model performs consistently better than other models. For ADG VI, the R^2 fit values are higher than other ADG's. This can be due to lower number of observations in this category. Moreover, the aircraft belonging to this category mainly take right angle exits. The exit speeds observed at right angle exits have low standard deviation. This could be the other reason for higher R^2 fit values in this category.

Table 19 : R^2 Fit Values of Different Models for Different ADG's

Model	ADG I	ADG II	ADG III	ADG IV	ADG V	ADG VI
Multi-linear Regression Model	0.94	0.88	0.90	0.89	0.88	0.95
Quadratic Model	0.85	0.86	0.80	0.87	0.90	0.93
Neural Networks Model	0.95	0.93	0.89	0.93	0.93	0.96

The model predicts mean exit speeds at a particular exit using the exit geometry parameters from the exit database.

However, the randomness in the data is simulated by using the standard deviation obtained from the exit speeds at the point of curvature from ASDE–X data. **Figure *35*** shows the cumulative density plots of predicted and observed exit speeds at the point of curvature for ADG I**.** The model predicts the exit speeds in the range of the observed exit speeds.

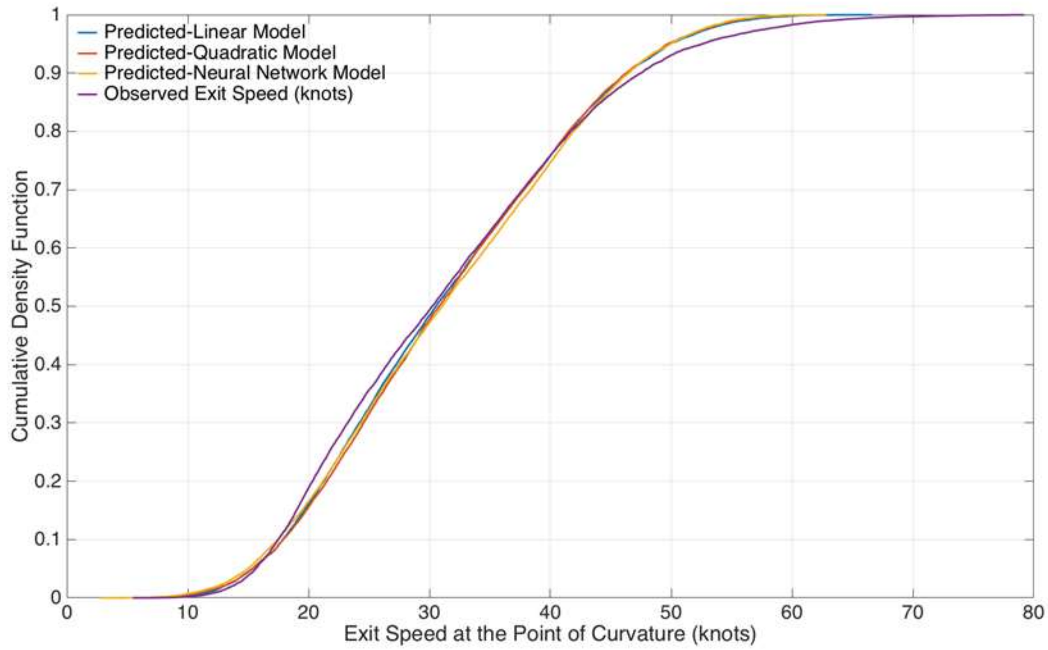


Figure 35 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for all Exits (ADG I).

However, the robustness of a model is determined by how good it predicts exit speeds for all exit types. Therefore, cumulative density function plots for different exit types are made. **Figure 36** shows the observed and predicted exit speeds at the point of curvature for high-speed exits have median exit speed of 48 knots. Also, observed and predicted exit speeds have 95% observations below 65 knots.

For right angle exits, **Figure 37** shows the median observed and predicted exit speed (linear model) at the point of curvature is 22 knots. At 95% and above the observed and predicted exit speed plots diverge. The predicted exit speeds are conservative and don't exceed speeds of 35 knots at the point of curvature.

For wide throat exits, **Figure 38** shows the median observed and predicted exit speed by linear model are 43 and 45 knots. However, at lower percentile, the model predicts lower speeds compared to the observed speeds at the point of curvature. It is safe to predict lower exit speeds because higher exit speeds lead to higher lateral loads on the aircraft landing gear while maneuvering the exit.

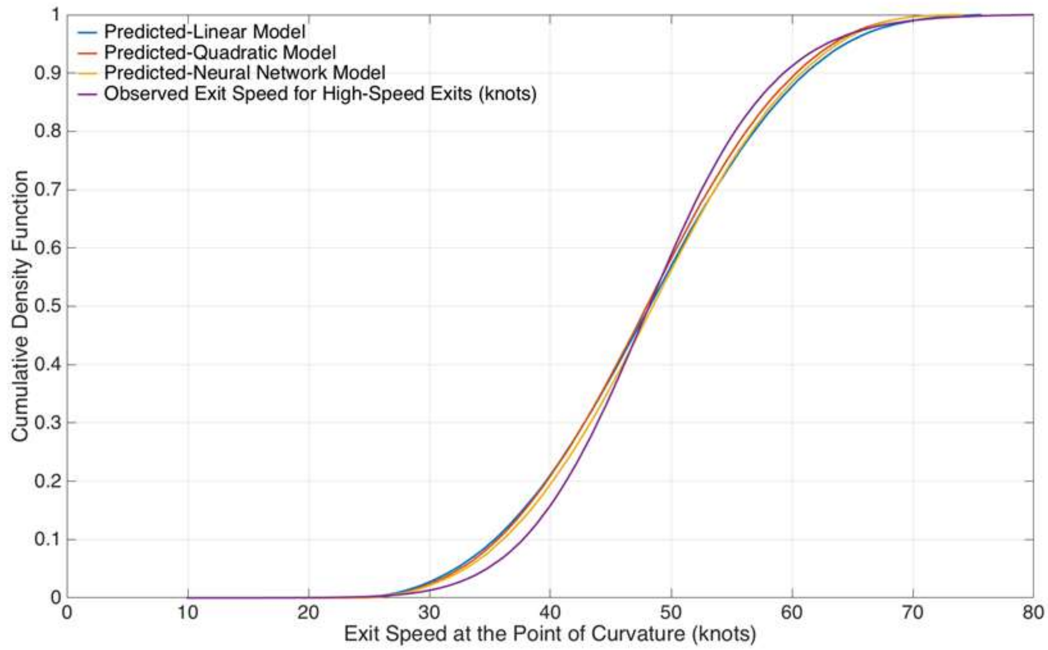


Figure 36 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for High-Speed Exits (ADG II).

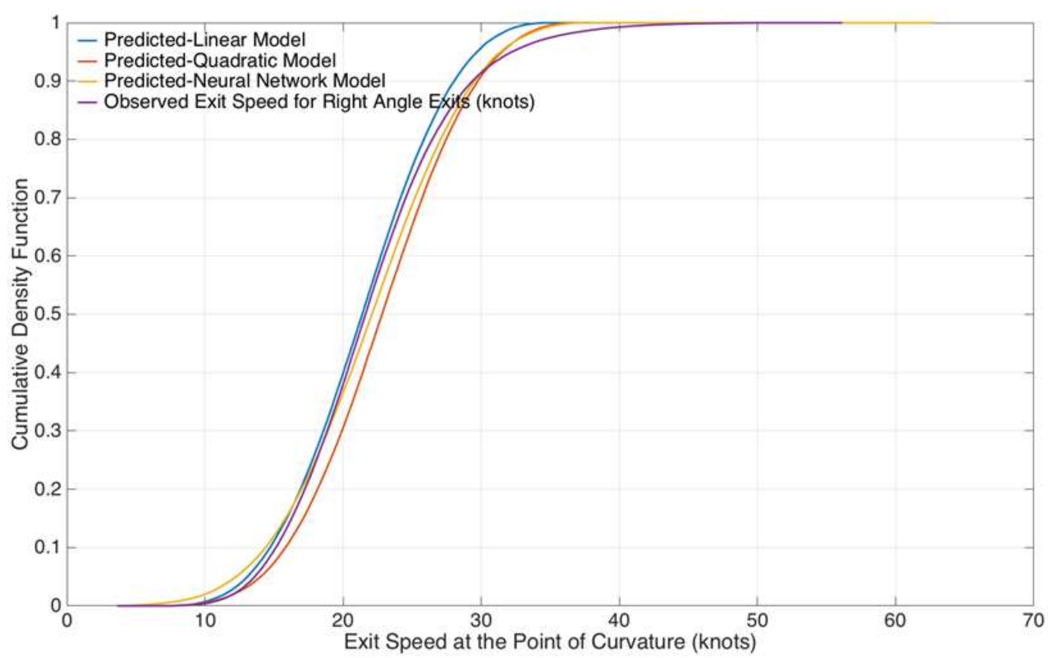


Figure 37 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Right Angle Exits (ADG III).

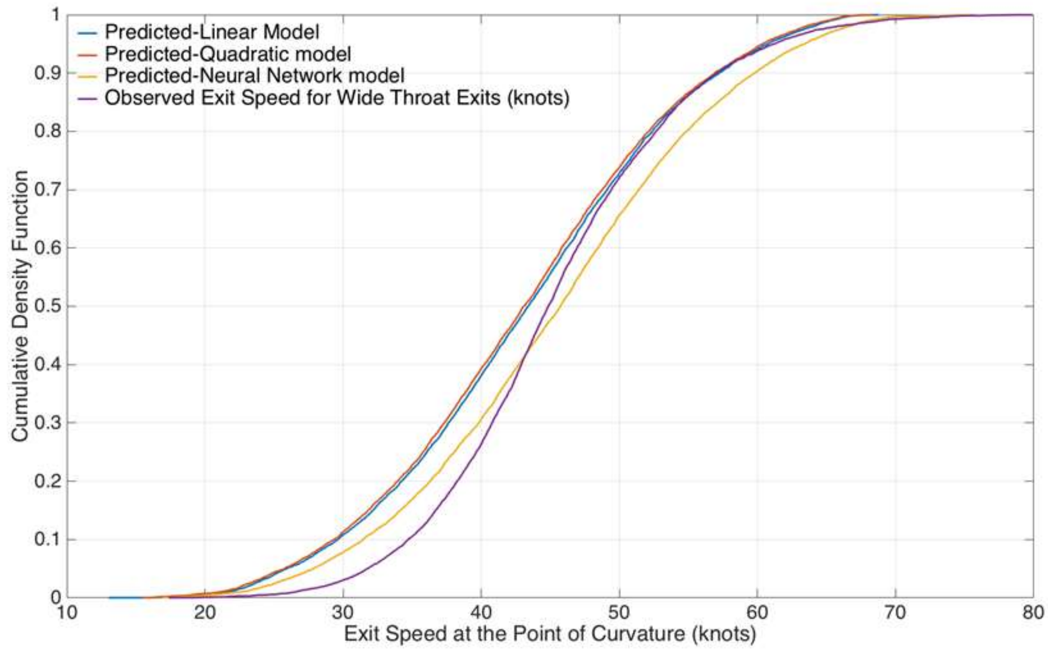


Figure 38 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Wide Throat Exits (ADG IV).

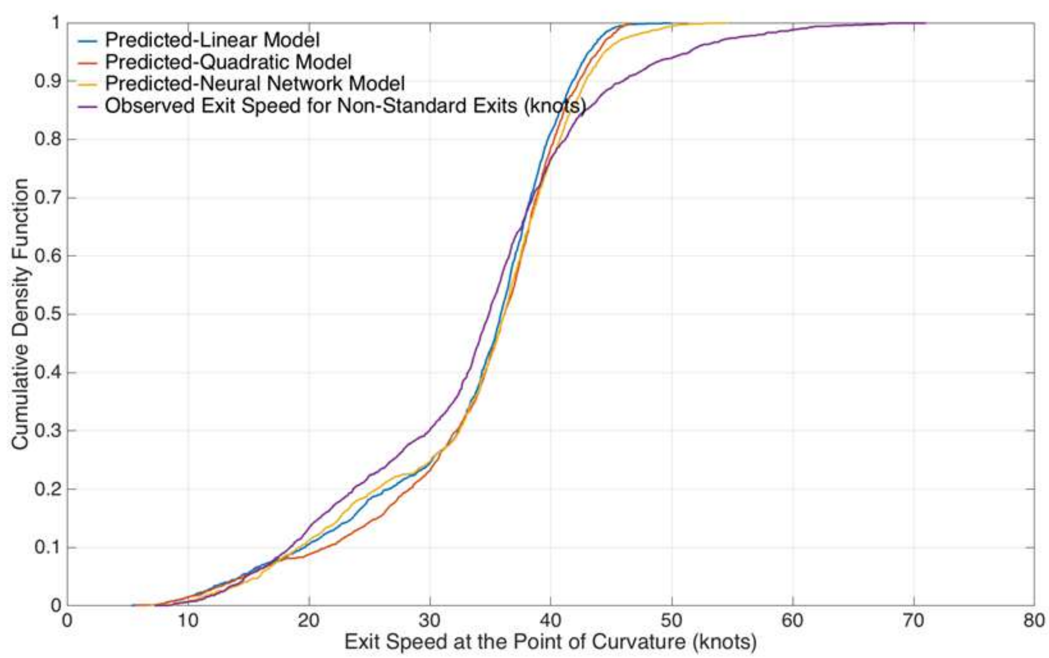


Figure 39 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Non Standard Exits (ADG VI).

For non-standard exits which are not either high-speed exits, right angle exits, wide throat exits or obtuse angle exits, **Figure 39** shows the cumulative density function plots of observed and predicted exit speeds. The median observed and predicted exit speeds are almost equal. However, the observed exit speeds diverge from predicted exit speeds at higher percentile. Such similar plots for different ADG's can be found in the next section under validation plots.

3.7 Validation Plots

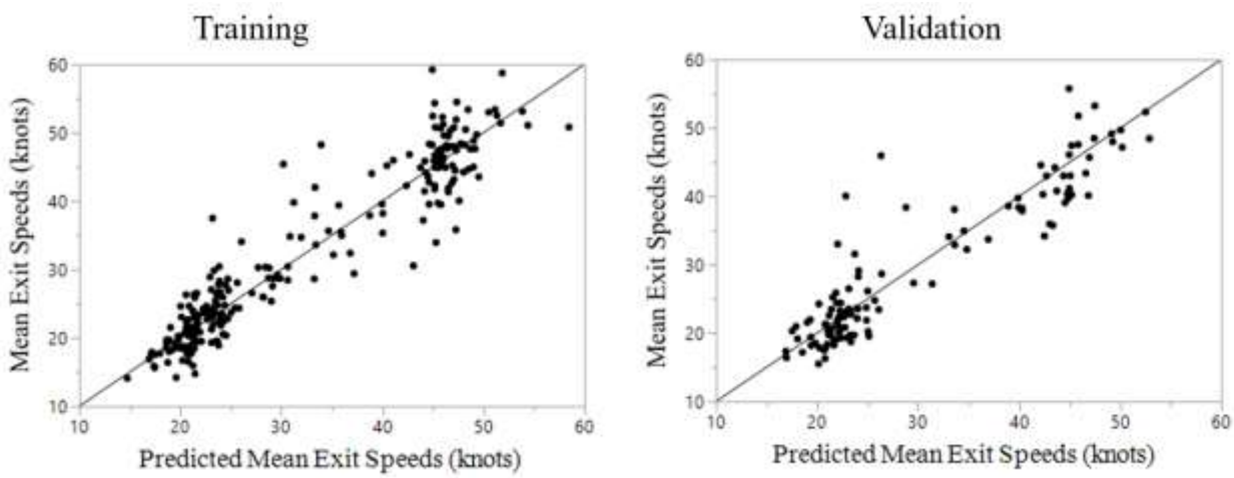


Figure 40 : Training and Validation Plots for Observed and Predicted Mean Exit Speeds at the Point of Curvature of an Exit for ADG II.

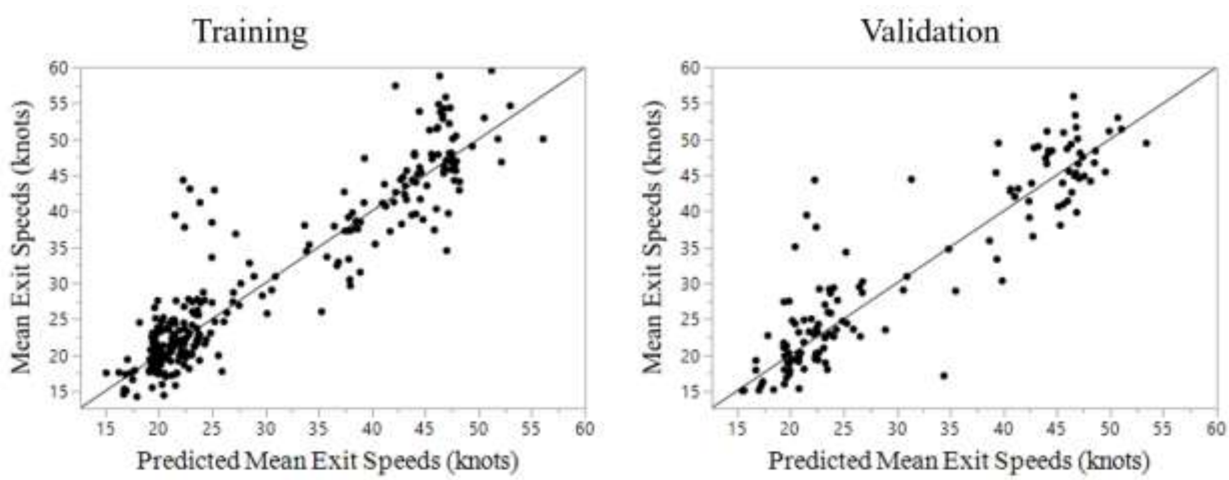


Figure 41 : Training and Validation Plots for Observed and Predicted Mean Exit Speeds at the Point of Curvature of an Exit for ADG III.

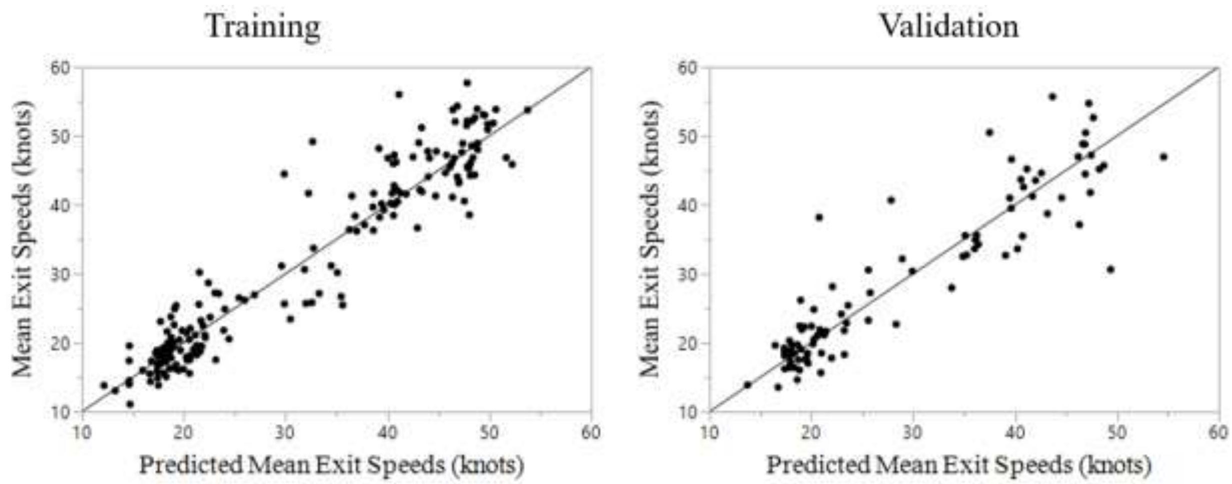


Figure 42 : Training and Validation Plots for Observed and Predicted Mean Exit Speeds at the Point of Curvature of an Exit for ADG IV.

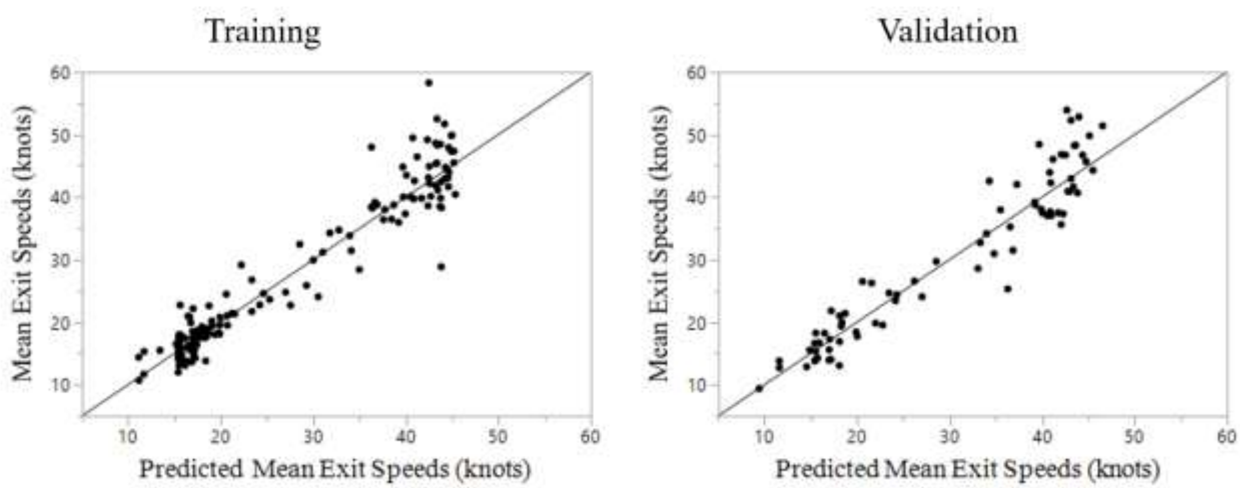


Figure 43 : Training and Validation Plots for Observed and Predicted Mean Exit Speeds at the Point of Curvature of an Exit for ADG V.

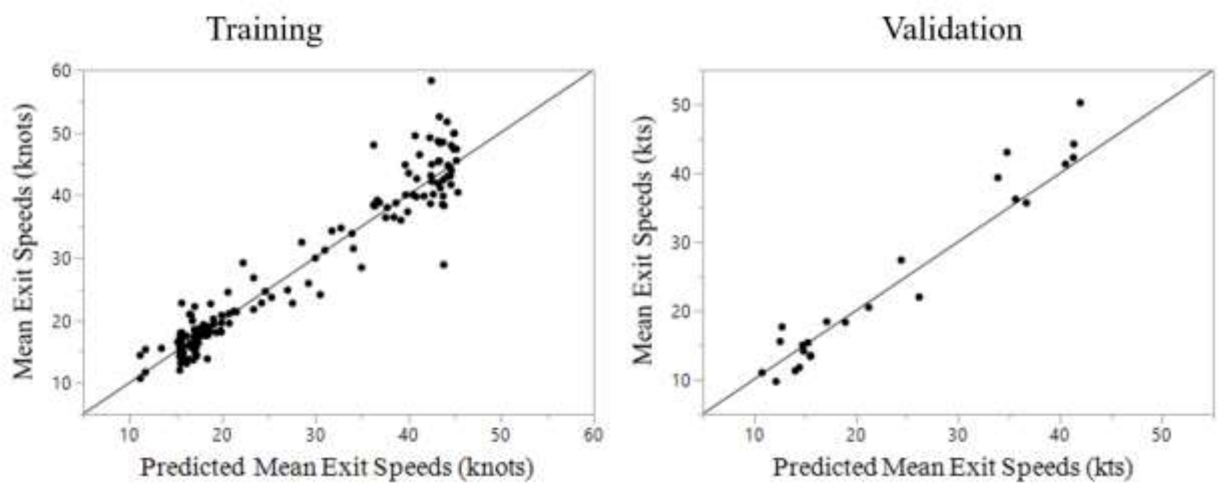


Figure 44 : Training and Validation Plots for Observed and Predicted Mean Exit Speeds at the Point of Curvature of an Exit for ADG VI.

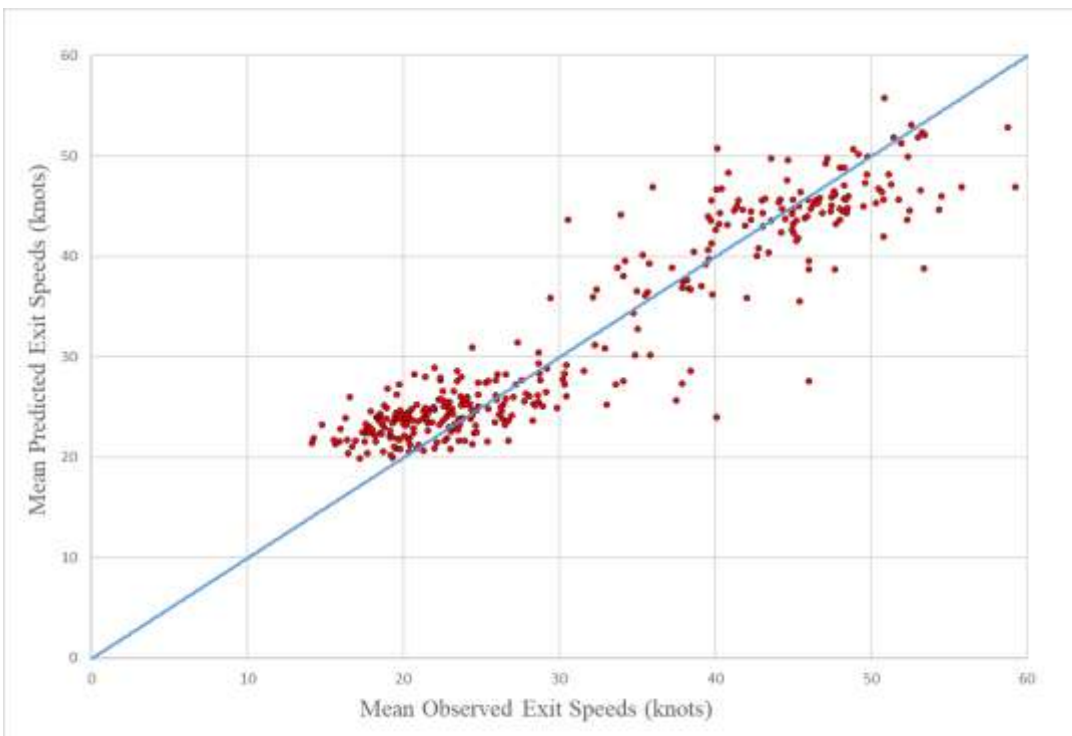


Figure 45 : Mean Predicted Exit Speeds (knots) vs Mean Observed Exit Speeds (knots) for ADG II Using Multi-Linear Regression Model

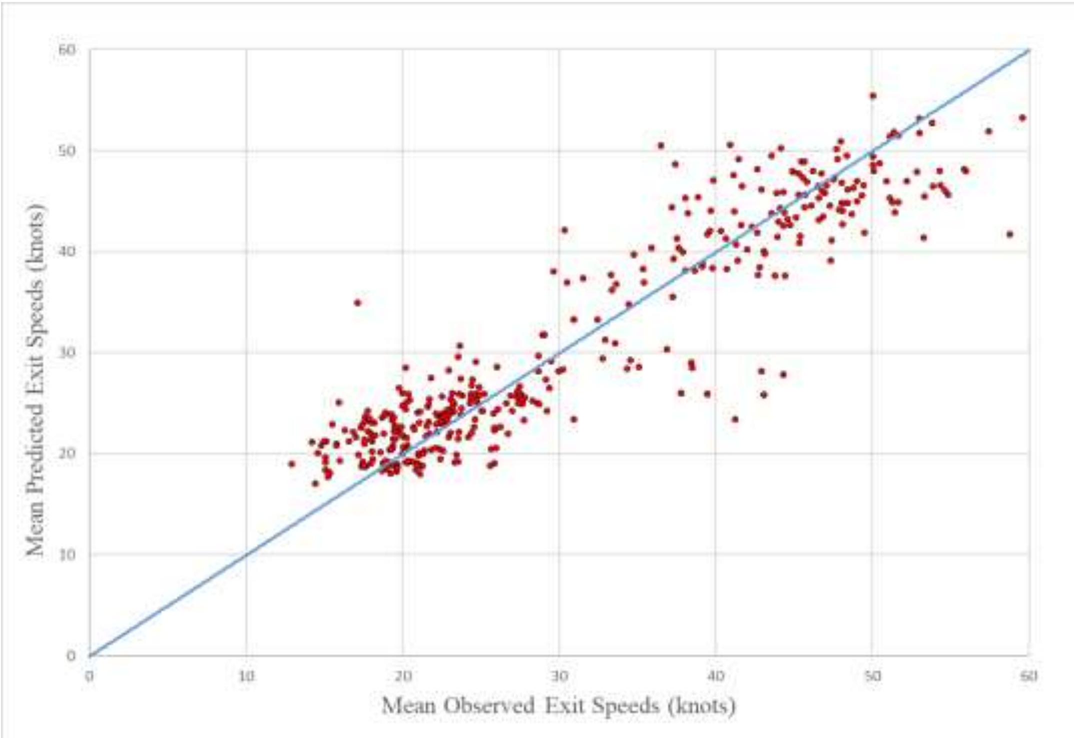


Figure 46 : Mean Predicted Exit Speeds (knots) vs Mean Observed Exit Speeds (knots) for ADG III Using Multi-Linear Regression Model

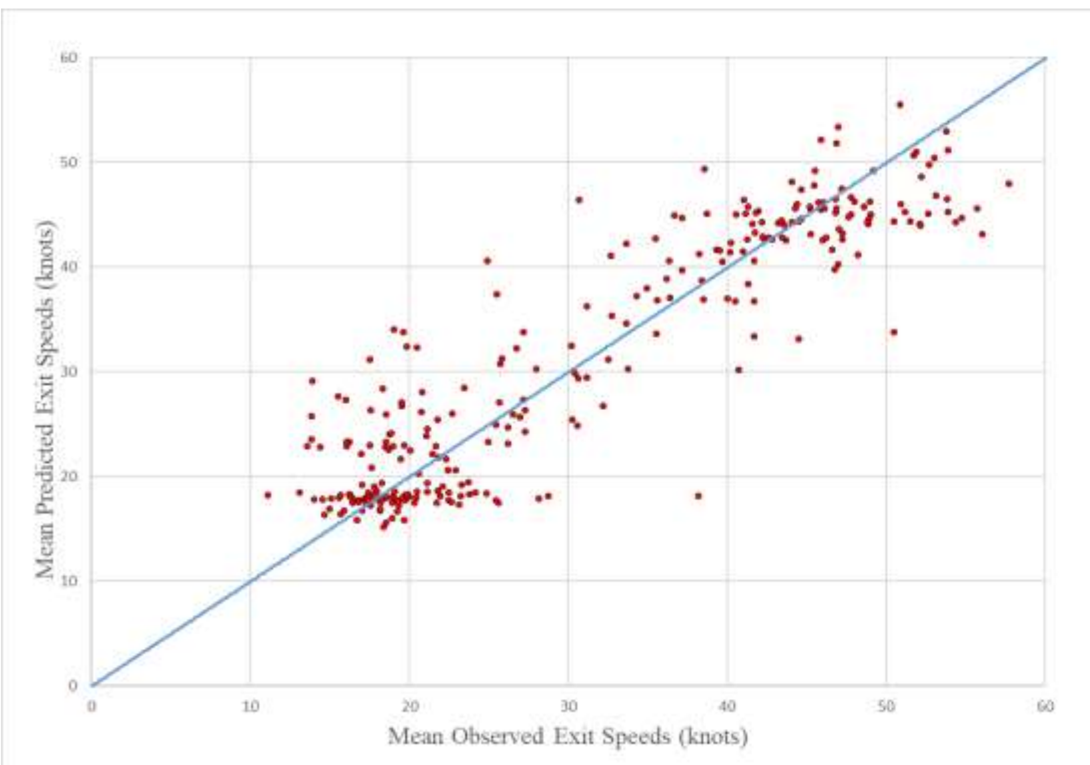


Figure 47 : Mean Predicted Exit Speeds (knots) vs Mean Observed Exit Speeds (knots) for ADG IV Using Multi-Linear Regression Model

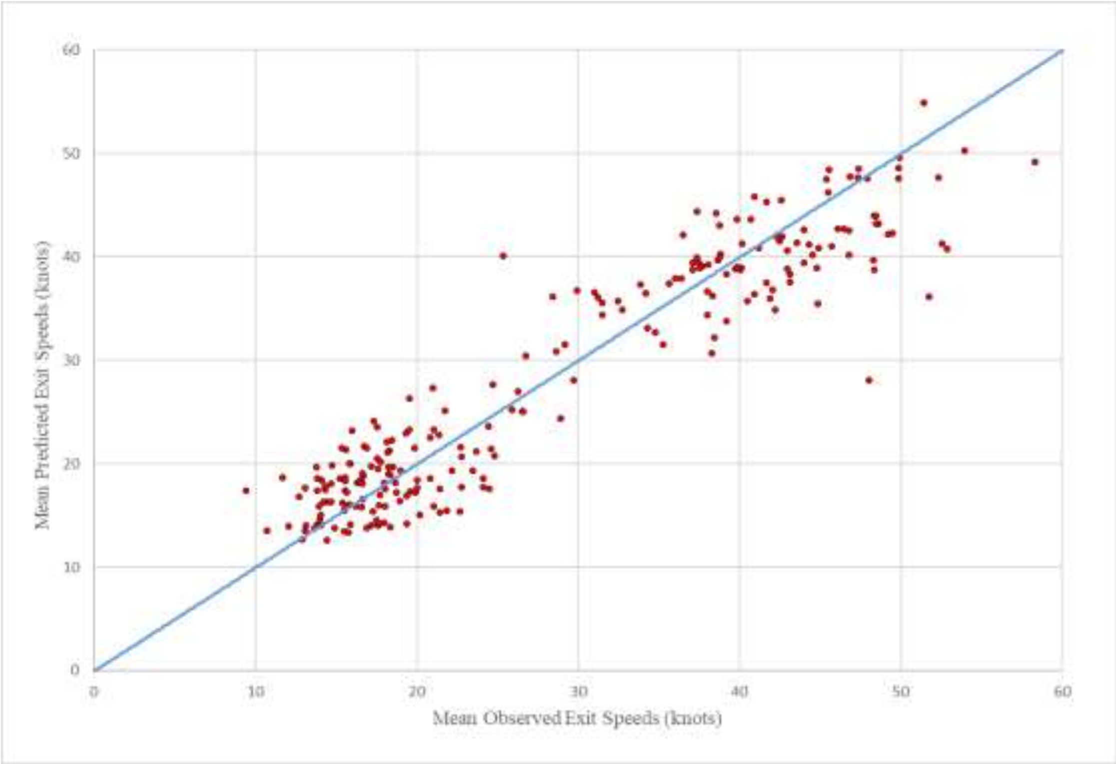


Figure 48 : Mean Predicted Exit Speeds (knots) vs Mean Observed Exit Speeds (knots) for ADG V

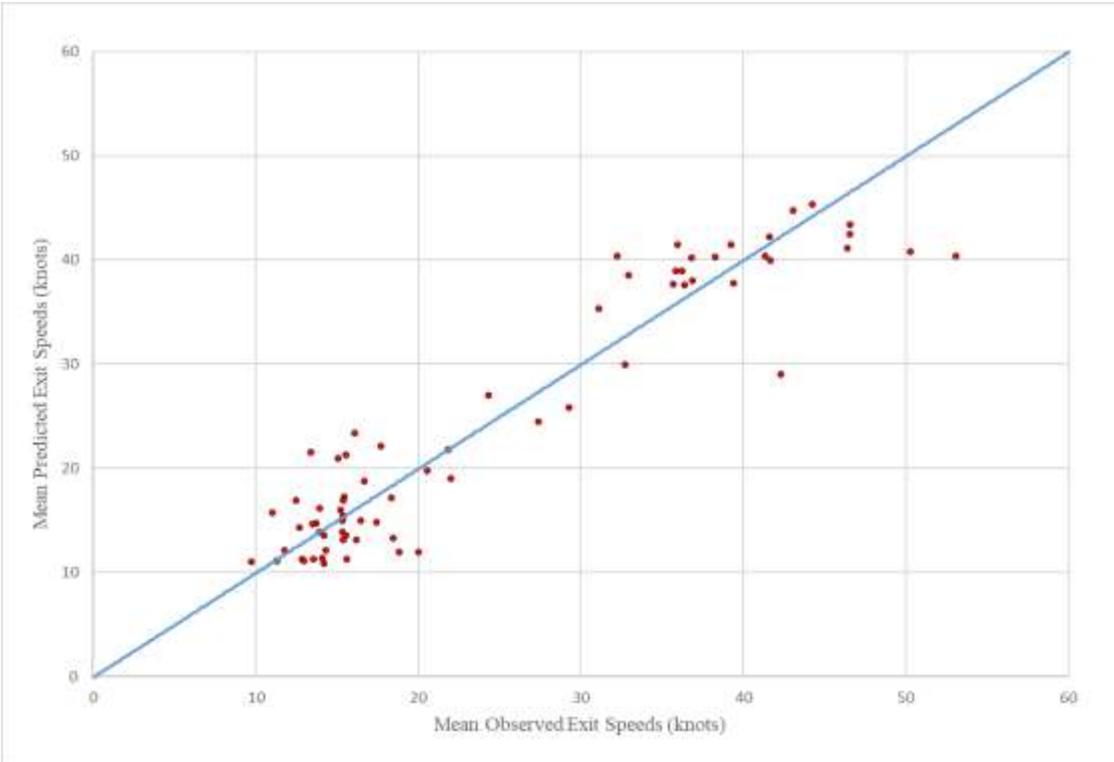


Figure 49 : Mean Predicted Exit Speeds (knots) vs Mean Observed Exit Speeds (knots) for ADG VI Using Multi-Linear Regression Model

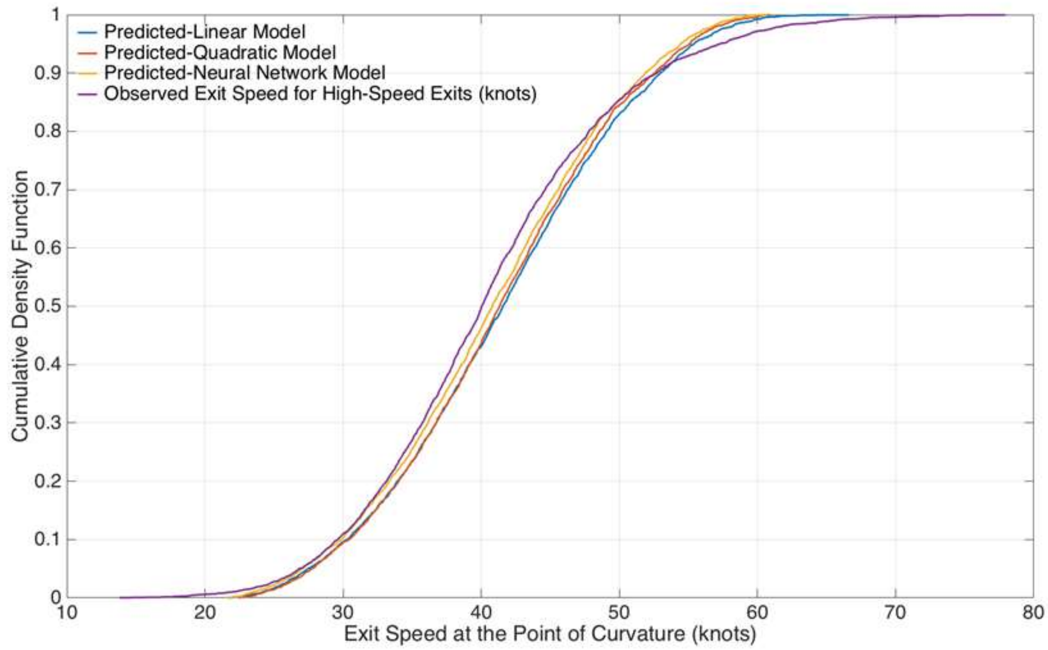


Figure 50 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for High-Speed Exits (ADG I).

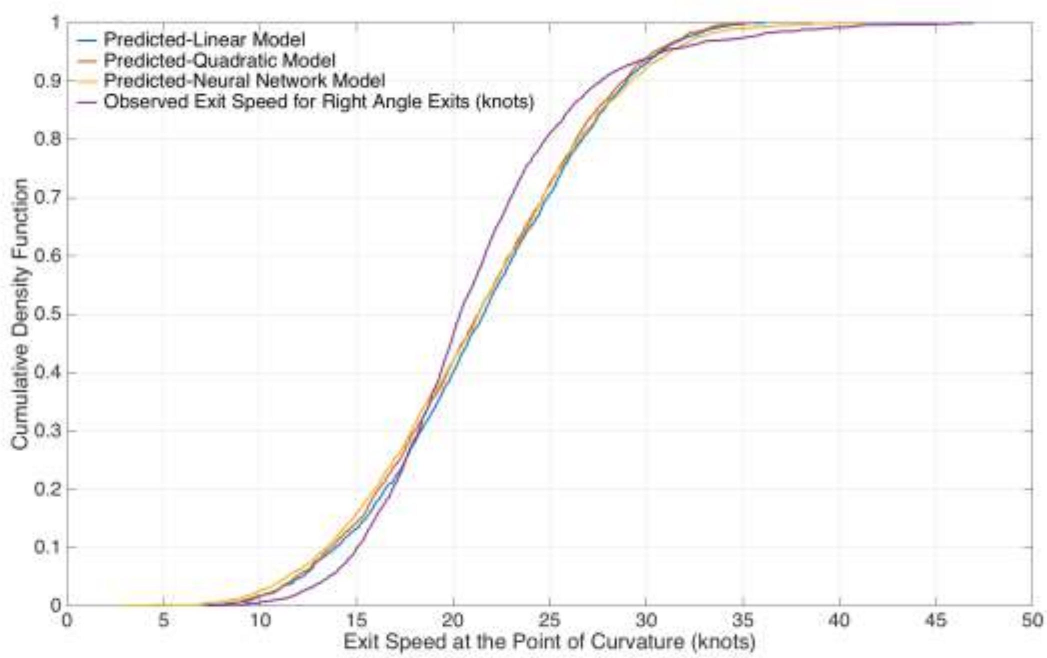


Figure 51 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Right Angle Exits (ADG I).

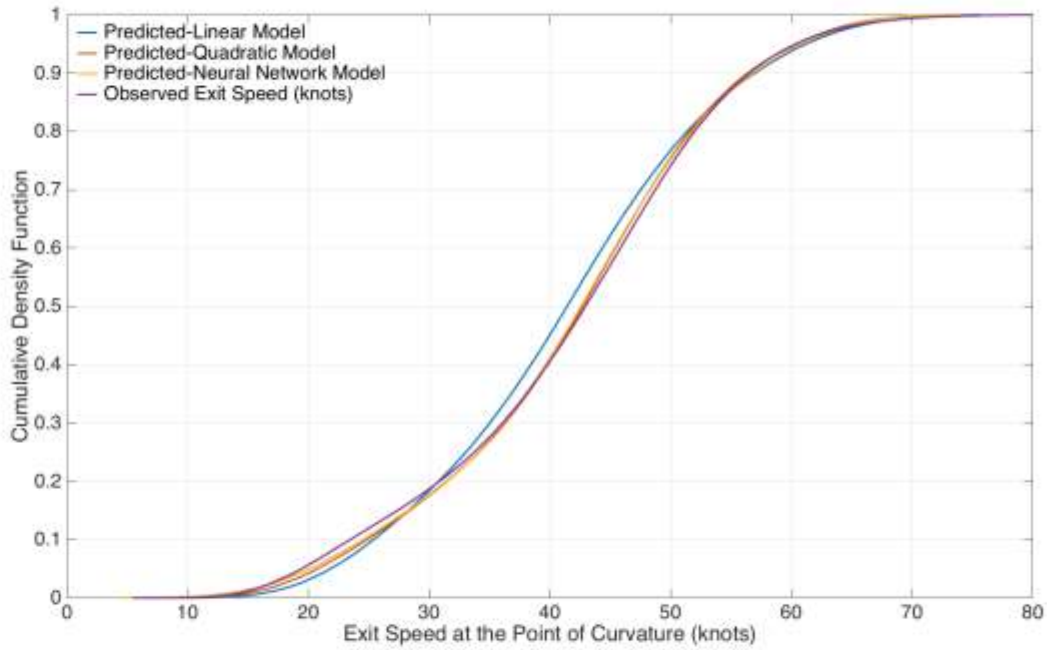


Figure 52 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for all Exits (ADG II).

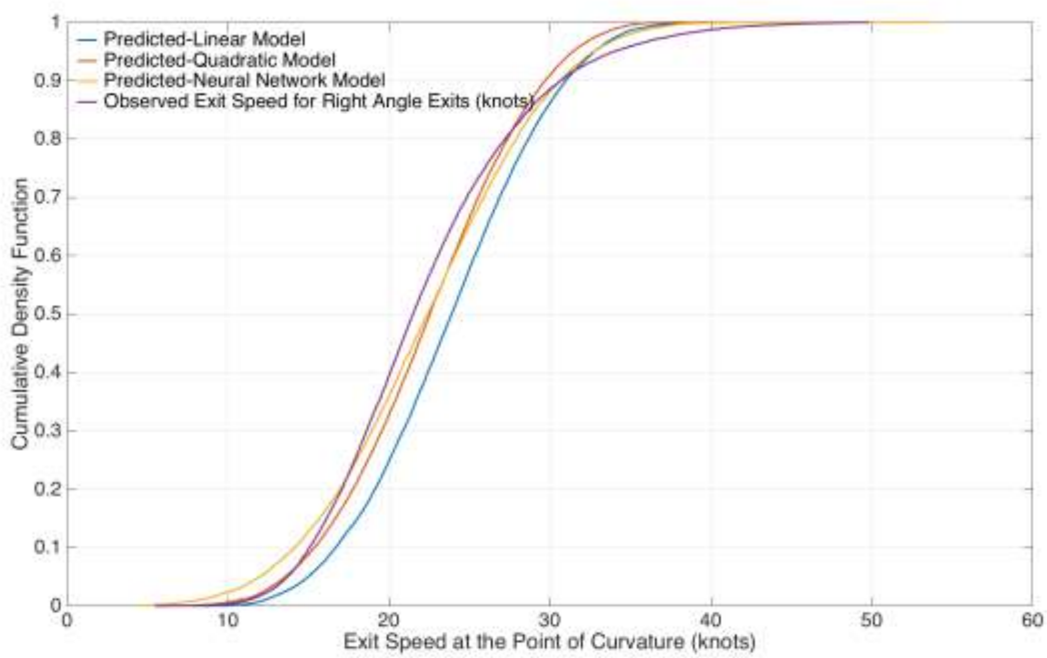


Figure 53 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Right Angle Exits (ADG II).

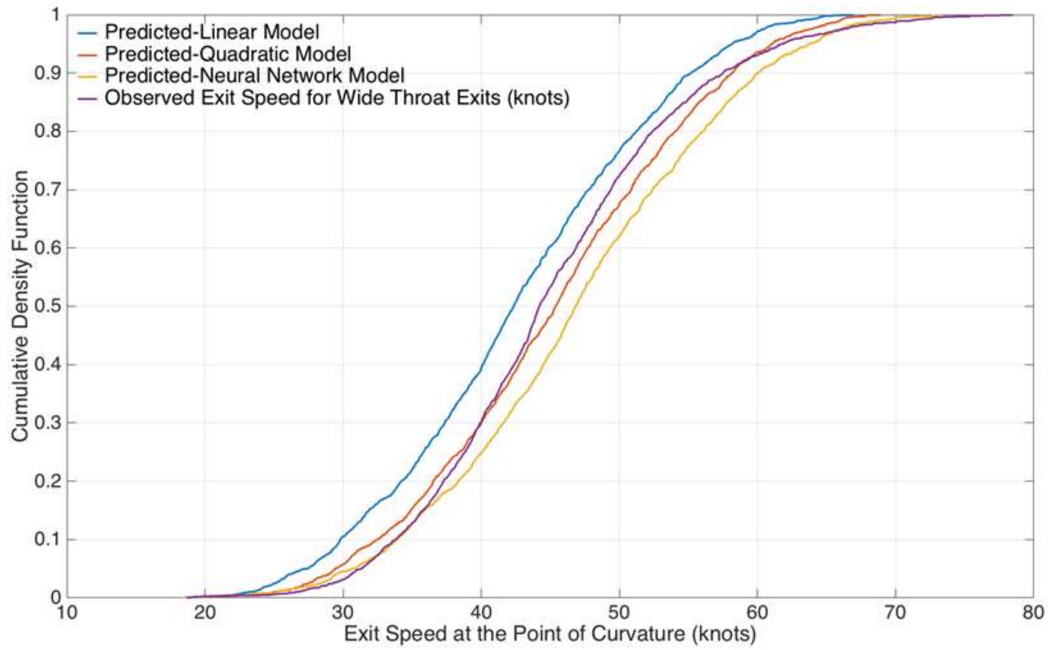


Figure 54 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Wide Throat Exits (ADG II).

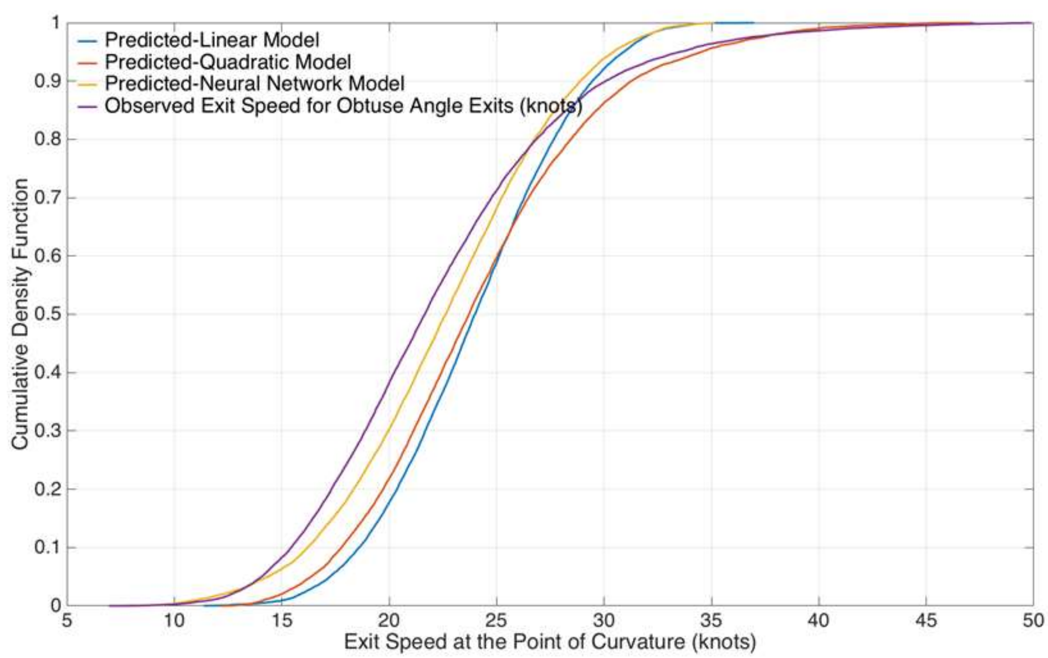


Figure 55 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Obtuse Angle Exits (ADG II).

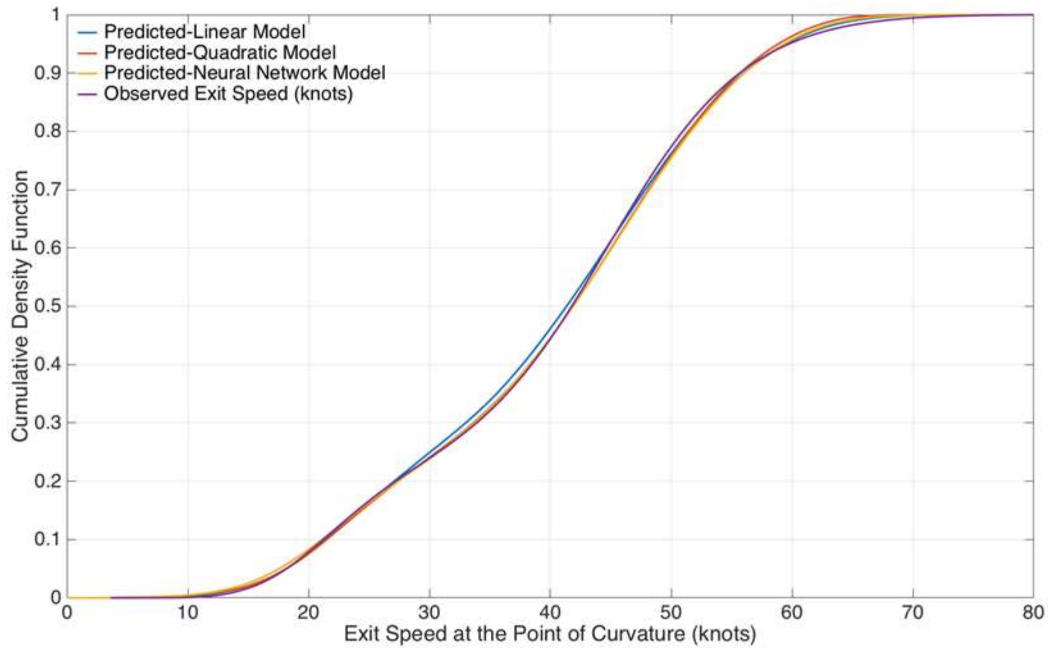


Figure 56 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for all Exits (ADG III).

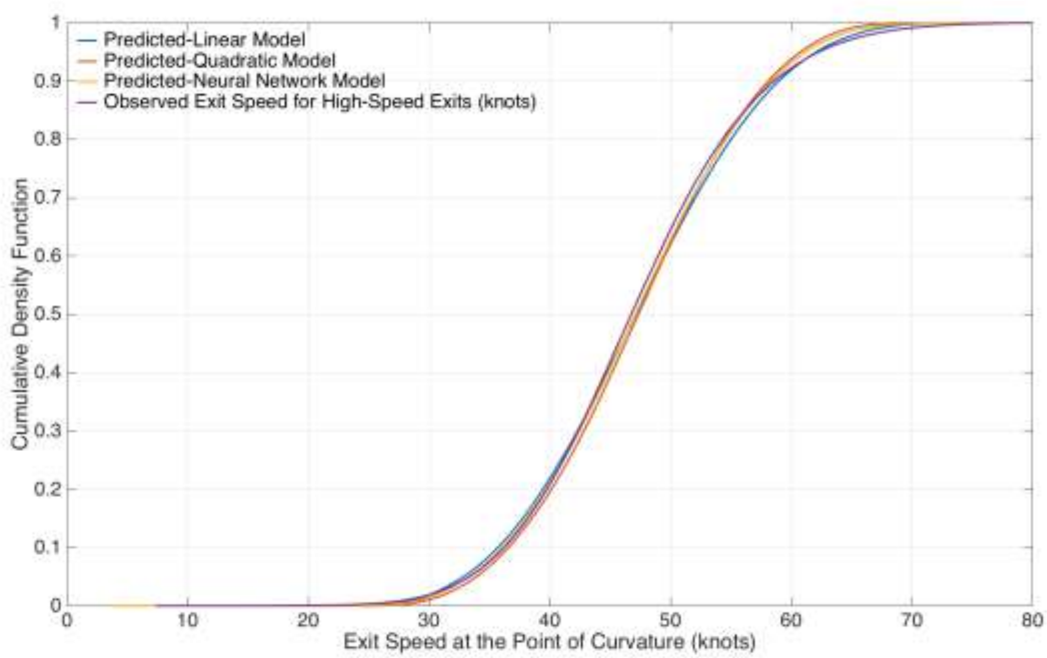


Figure 57 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for High-Speed Exits (ADG III).

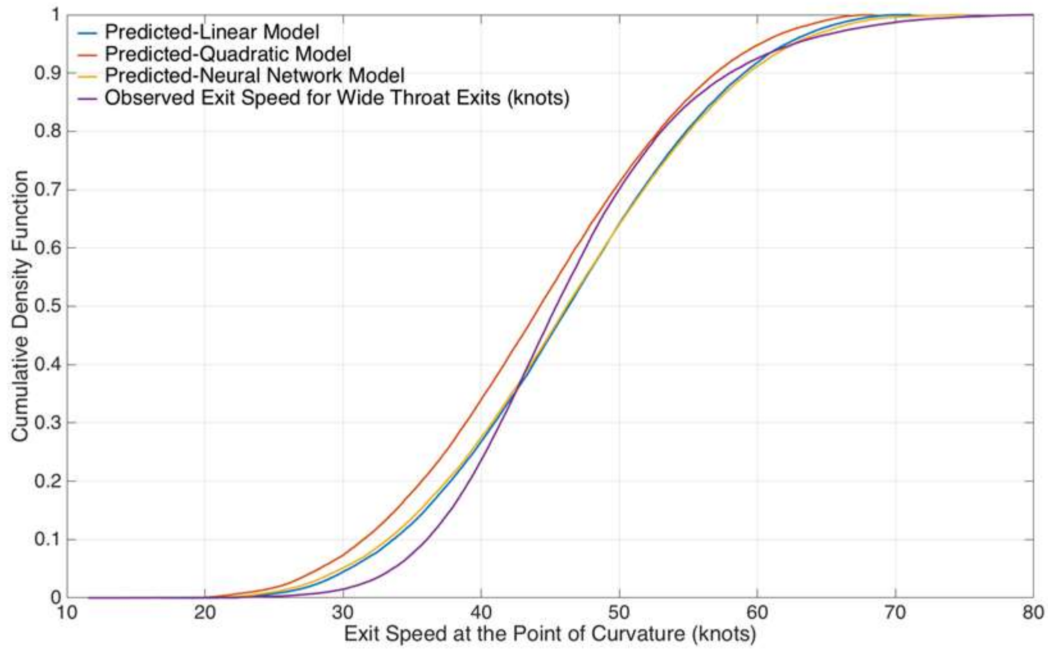


Figure 58 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Wide Throat Exits (ADG III).

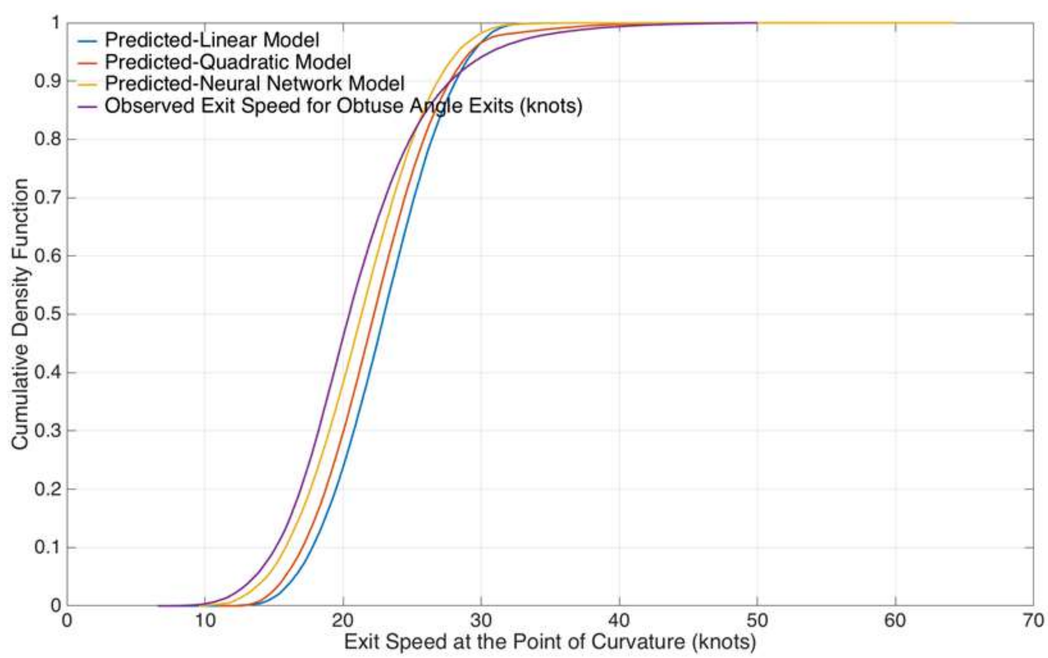


Figure 59 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Obtuse Angle Exits (ADG III).

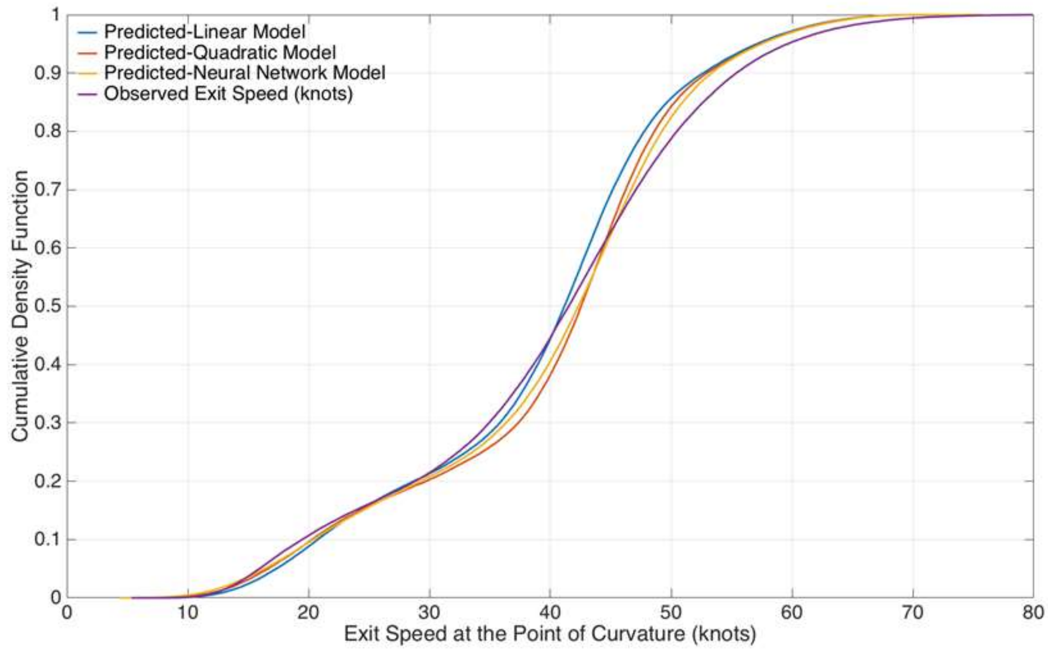


Figure 60 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for all Exits (ADG IV).

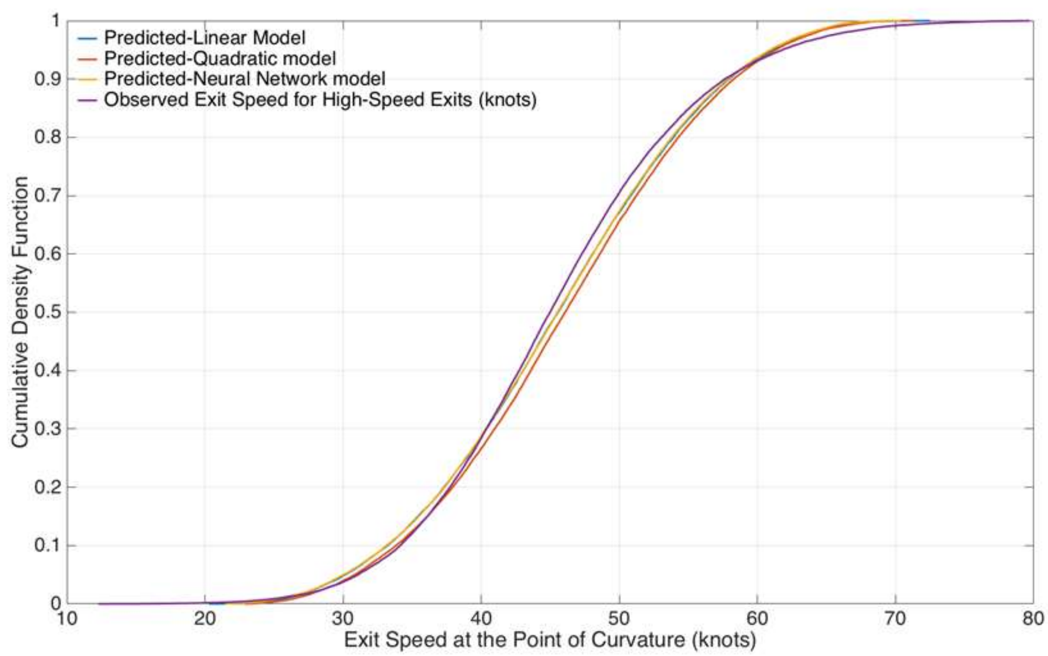


Figure 61 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for High-Speed Exits (ADG IV).

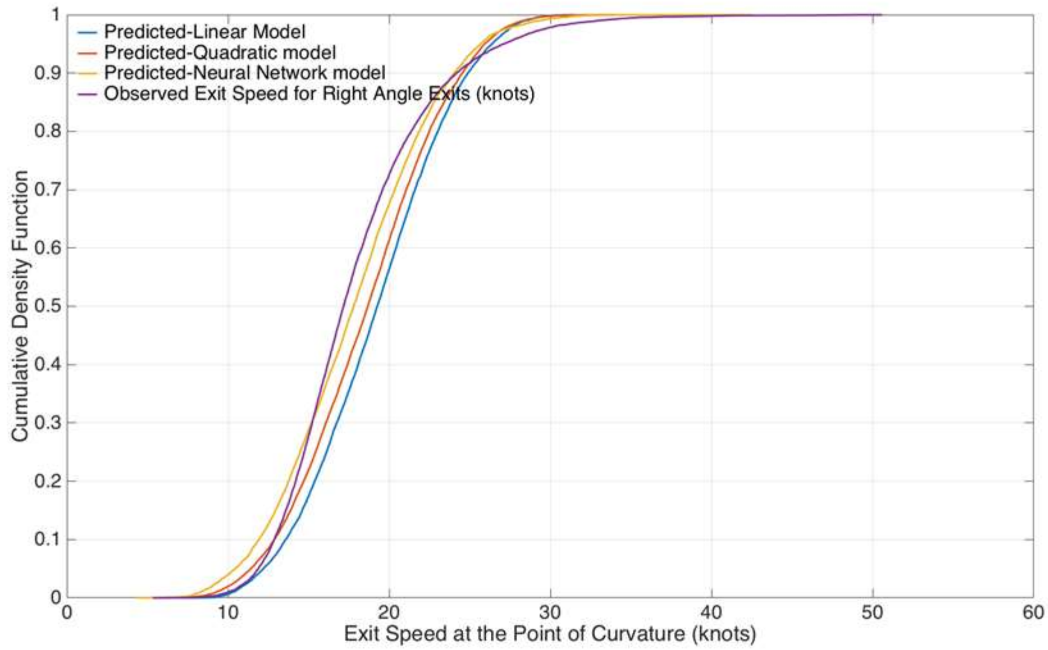


Figure 62 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Right Angle Exits (ADG IV).

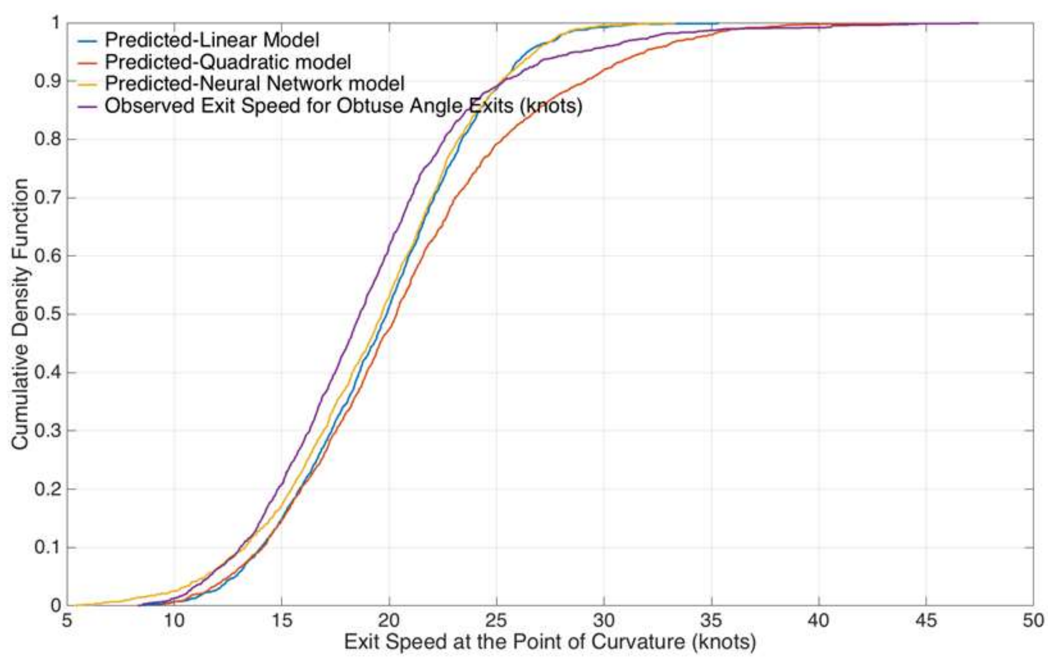


Figure 63 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Obtuse Angle Exits (ADG IV).

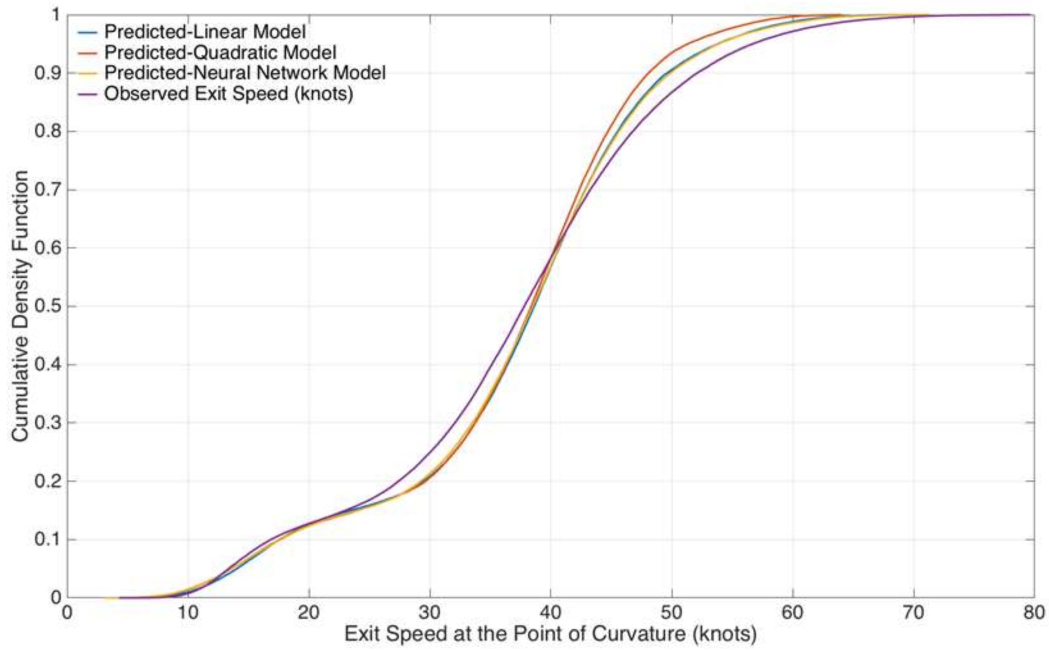


Figure 64 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for all Exits (ADG V).

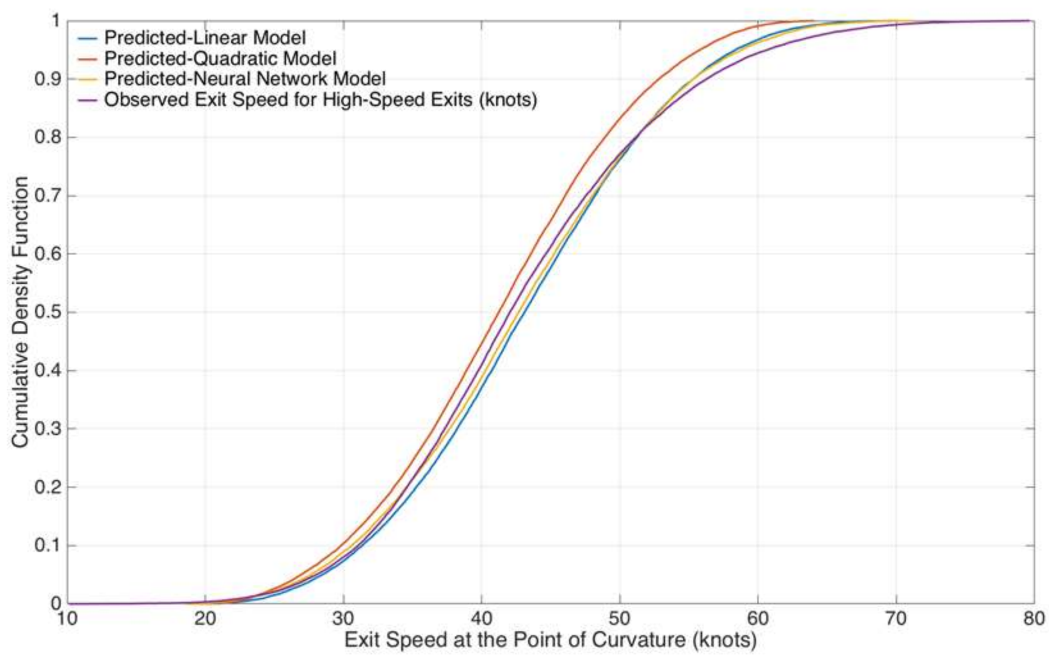


Figure 65 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for High-Speed Exits (ADG V).

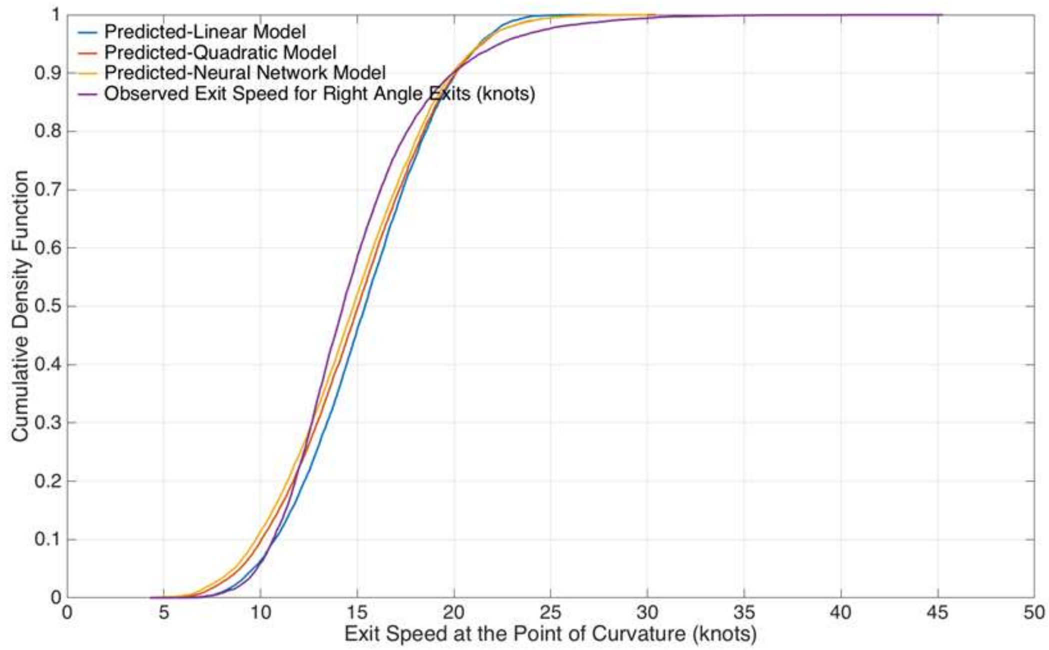


Figure 66 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Right Angle Exits (ADG V).

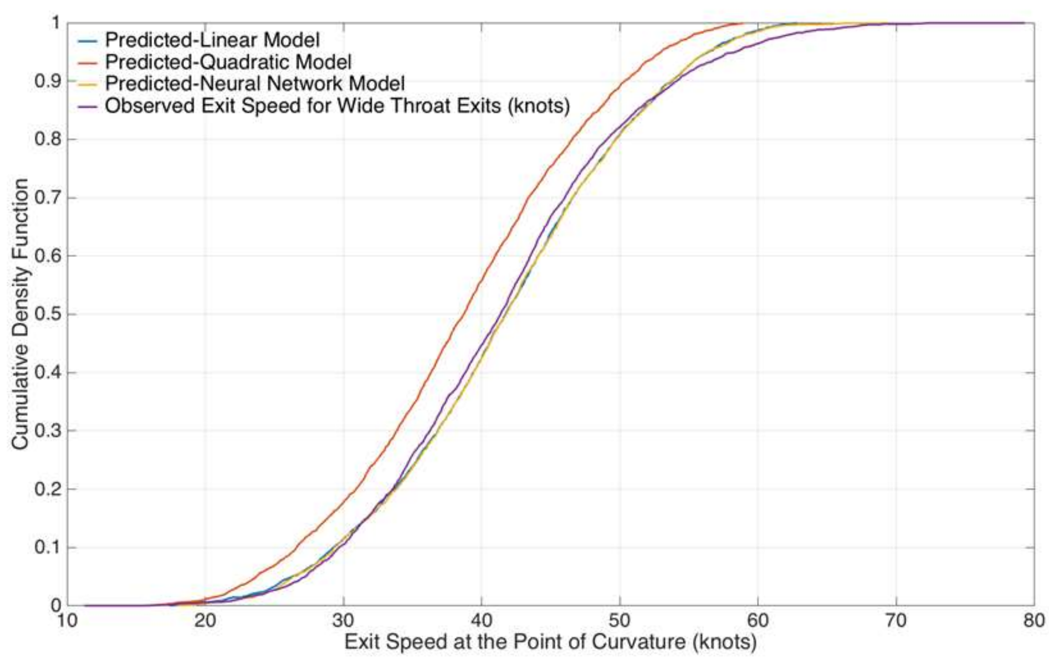


Figure 67 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Wide Throat Exits (ADG V).

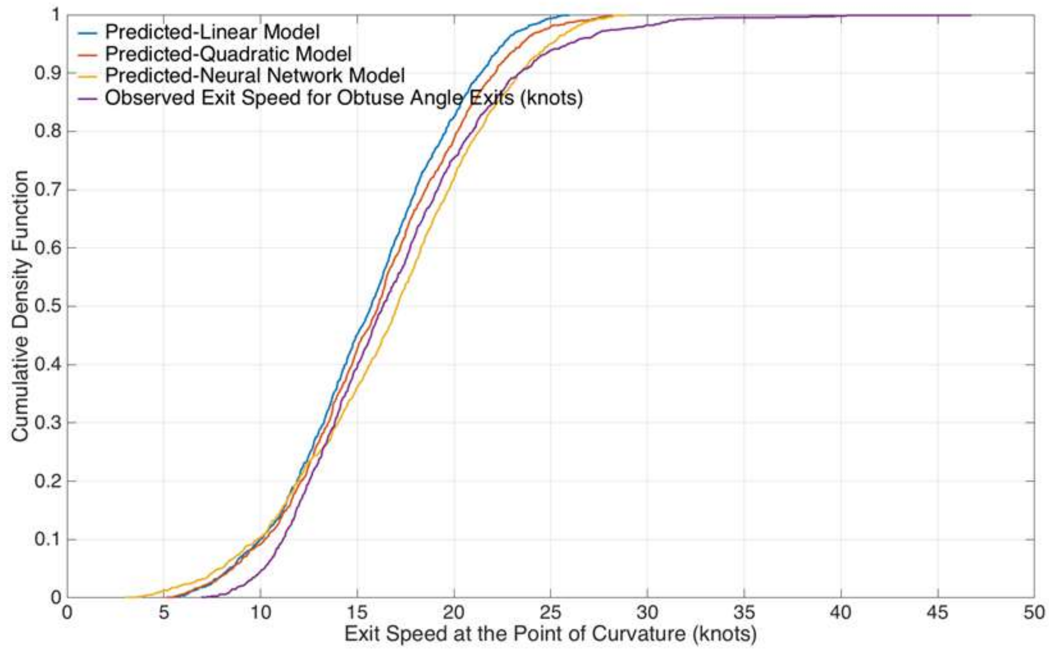


Figure 68 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Obtuse Angle Exits (ADG V).

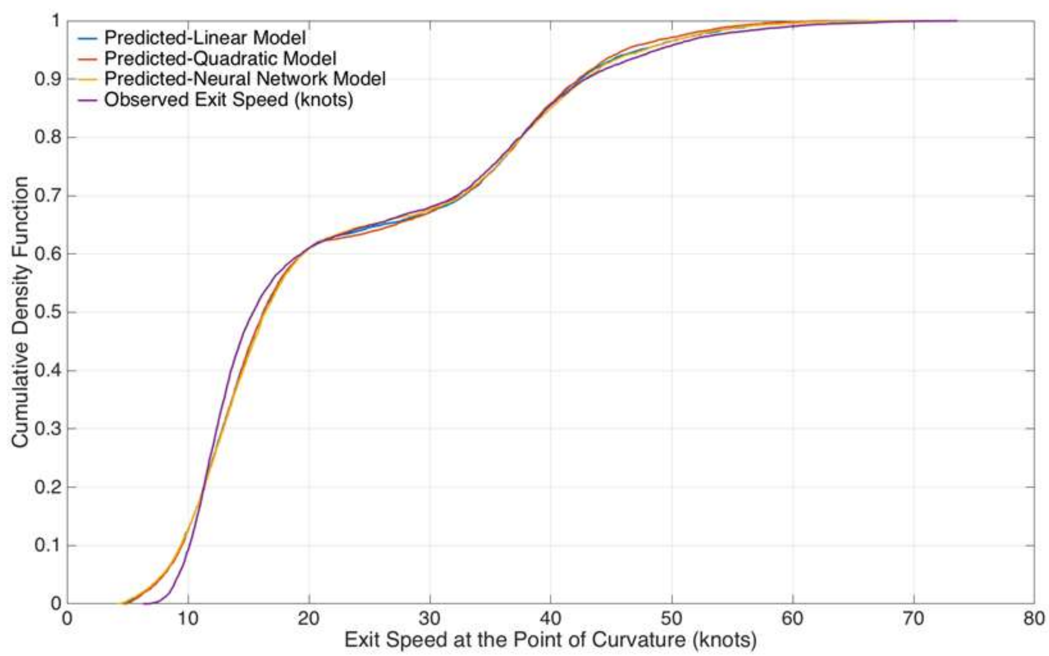


Figure 69 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for all Exits (ADG VI).

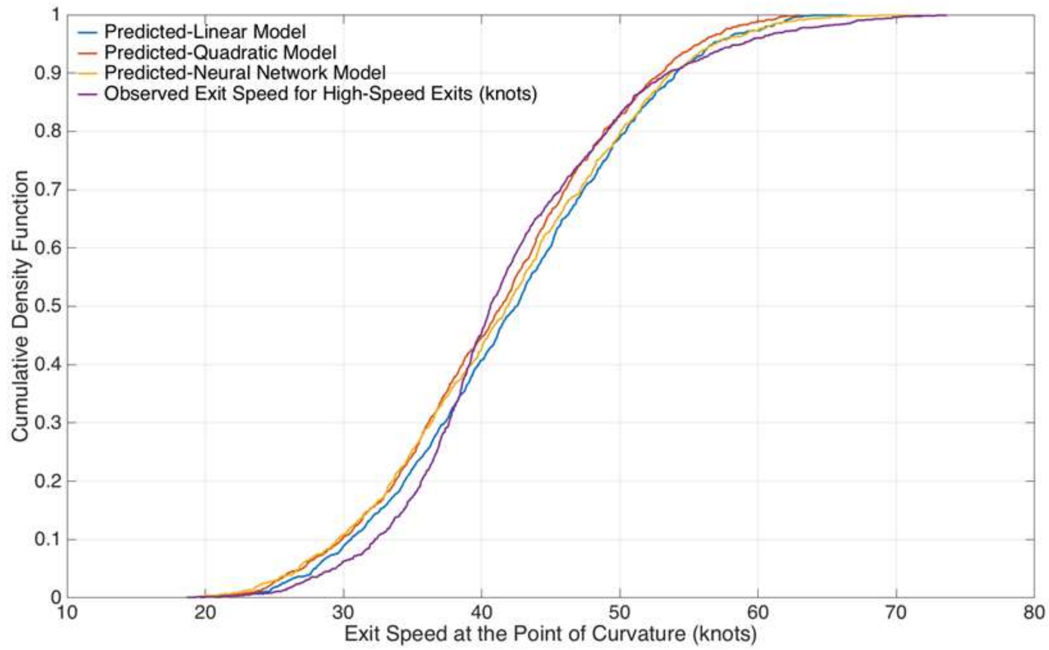


Figure 70 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for High-Speed Exits (ADG VI).

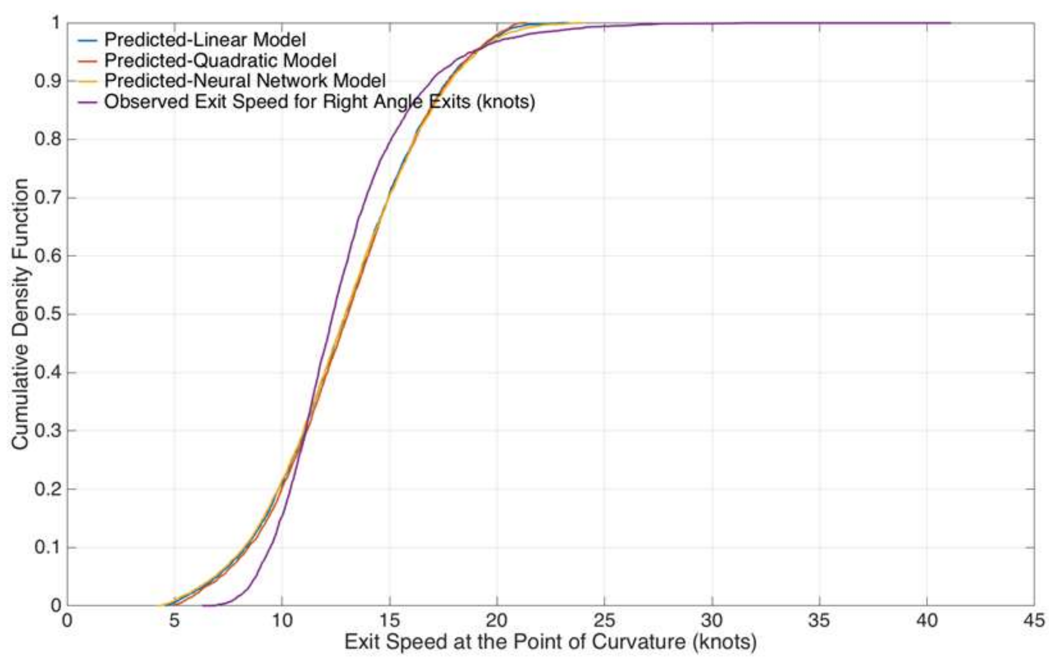


Figure 71 : Cumulative Density Function Plot of Exit Speeds at the Point of Curvature for Right Angle Exits (ADG VI).

CHAPTER 4: RECOMMENDATIONS AND CONCLUSION

4.1 Conclusion

The Runway Exit Database has information about exit geometry parameters from 37 ASDE-X airports. The total number of exits at these airports is equal to 3386. Among them, 238 exits are standard high-speed exits. A standard high-speed exit has longer path lengths, greater than 900 ft. They also have turn radius greater than 1400 ft and exit angle lower than 35^0. There are also 77 non-standard high-speed exits with exit angles lower than 35^0 and path length lower than 900 ft or radius lower than 1400 ft. Of the 77 non-standard high-speed exits, 31 are on short runways with runway lengths lower than 7500 ft. The remaining 46 non-standard high-speed exits are on long runways with runway lengths greater than 7500 ft. The short runways tend to have non-standard high-speed exits because the aircraft type using these runways typically include ADG I and ADG II in many cases. These aircraft don't need standard high-speed exits to vacate the runways. On long runways, the reason for designing non-standard high-speed exits sometimes is due to space constraint surrounding the runways. The smaller space around the runways compromises the turn radius of the non-standard high-speed exit leading to shorter path lengths. Small airports such as LGA and DCA with space constraints have many non-standard high-speed exits.

The number of right-angle exits in the database are 1693. Most of the right-angle exits are designed for aircraft belonging to TDG 6 and 7. The radius of a right-angle exit varies with the runway to closest taxiway centerline distance. The maximum turn radius required is 135 ft for TDG 6 with runway to centerline distance 400 ft. Many exits have turn radius greater than 135 ft. The right-angle exits are placed uniformly all over the runway. Whereas, the high-speed exits are located mainly at the middle of the runway.

Aircraft belonging to ADG I, ADG II and ADG III largely use high-speed exits. The large aircraft belonging ADG IV, ADG V and ADG VI use both high-speed exits and right-angle exits. As the exit speeds observed at right angle exits is lower than the high-speed exits, the exit speed distribution at the point of curvature is bimodal for big aircraft compared to the small aircraft. Accounting for such differences involves using different runway exit speed models for each ADG.

The average exit speed at the point of curvature is maximum for ADG II (43 knots) due to the smaller size of the aircraft which gives more flexibility to the pilots in maneuvering the exits at higher speeds. The average exit speeds drop progressively from ADG II to ADG VI. The average exit speed is lowest for ADG VI aircraft due to the extensive use of right-angle exits. Also, their large aircraft size constraints the pilots from taking exits at higher speeds.

4.2 Recommendation

The data used for analysis is fully segmented based on aircraft belonging to different ADG's. This implies the data is segmented into six datasets as there are six ADG's in total. However, after full segmentation based on ADG's, each dataset has observations from a variety of exits including high-speed exits, right angle exits, wide throat exits, obtuse angle exits and other non-standard exits. Therefore, the model developed using this dataset has to account for all these exit types. However, in future, the analysis can go further by segmenting the datasets into different exit types. This can help analyze the exits better.

The exit speeds at which pilots maneuver an exit depends upon the type of exit taken. Path length is an important predictor of the exit speed at the point of curvature. Radius is the next important predictor in determining the exit speed at the point of curvature. Therefore, in the future, when designing any non-standard high-speed exit, the planner should keep in mind these two factors in designing the exit. Few high-speed exits with long path lengths and sharper turn radius (exit FB on runway 04R at JFK) have lower exit speeds compared to other high-speed exits with similar path length and shallow turn radius. This implies that high-speed exits having lower $\frac{path\ length}{Radius}$ tend to have higher exit speeds.

Airports with space constraints such as LGA and DCA have many non-standard high-speed exits. Many non-standard high-speed exits at these airports have hold bar closer to runway centerline compared to the standard high-speed exits. Due to this, the average exit speed at the point of curvature for a non-standard high-speed is lower compared to the standard high-speed exit. In such cases where the hold bar is closer to the runway centerline, a wide throat exit can be used instead of a non-standard high-speed exit. This study shows the average exit speed observed at high-speed exits is almost equal to the average exit speed observed at the point of curvature for wide-throat exits. Therefore, at airports such as LGA and DCA, the non-standard high-speed exits can be replaced by the wide throat exits which can help improve the capacity of the runway.

Moreover, the aircraft belonging to ADG VI, use the right angle exits four times the high-speed exit. Which implies poor placement of high-speed exits for aircraft belonging to ADG VI category. In future, this needs to change for airports having a higher percentage of ADG VI aircraft in their fleet mix.

The right angle exits designed using the previous FAA standards had a turn radius of 250 ft. However, the new FAA standards for designing right angle exits have lower turn radius. The turn radius depends on the taxiway design group of the aircraft and the separation between the runway and taxiway centerline.

REFERENCES

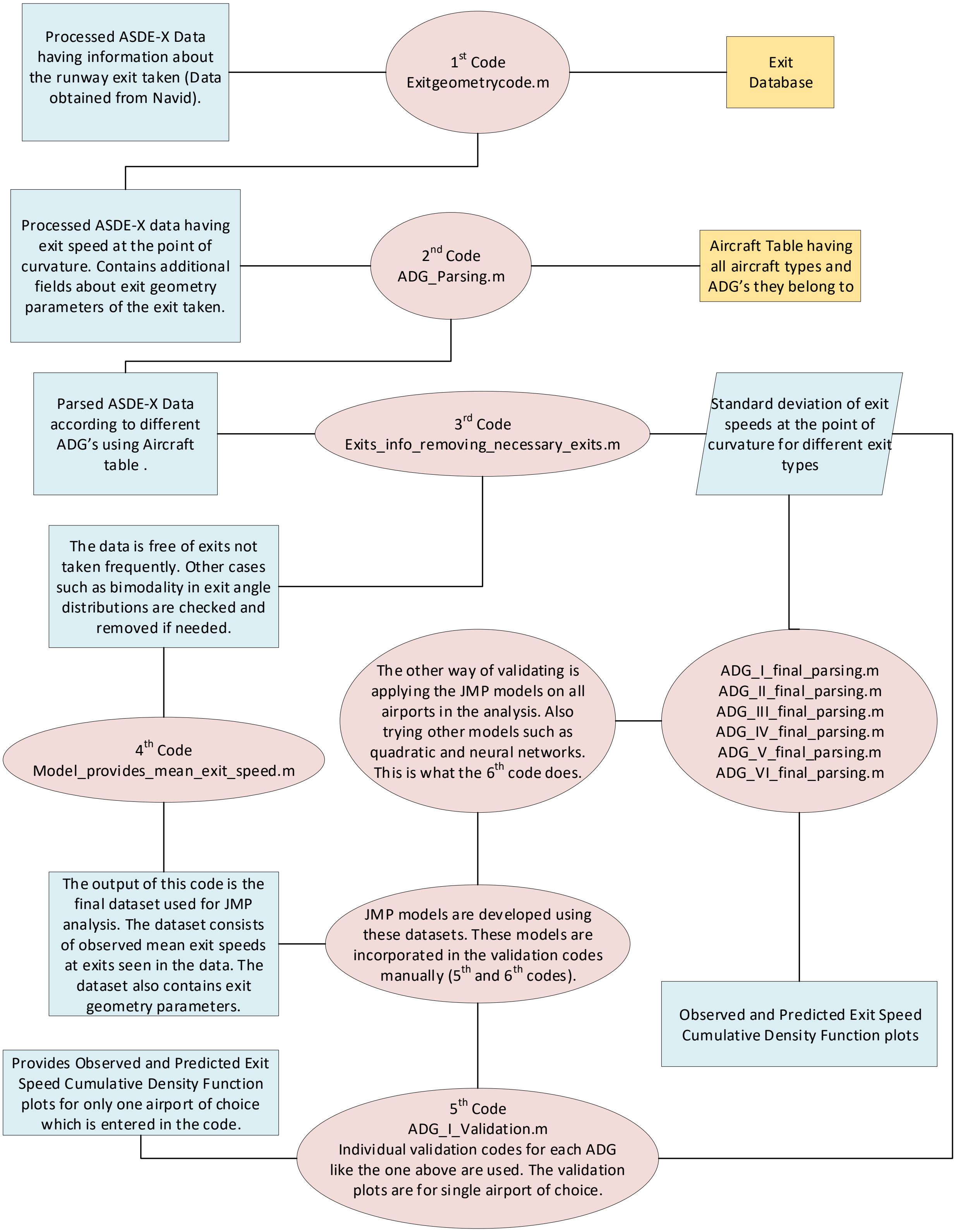
1. Xiao, Y., Fu, X., & Zhang, A. (2013). Demand uncertainty and airport capacity choice. Transportation Research Part B: Methodological, 57, 91-104.
2. Trani, A. A., Hobeika, A. G., Sherali, H. D., Kim, B. J., & Sadam, C. K. (1990). Runway Exit Designs for Capacity Improvement Demonstrations. Phase 1: Algorithmic Development, 90-32.
3. Kolos-Lakatos, T. (2013). The influence of runway occupancy time and wake vortex separation requirements on runway throughput (Doctoral dissertation, Massachusetts Institute of Technology).
4. Horonjeff, R., Finch, D.M., Belmont, D.M., and Ahlbom, G.; Exit Taxiway Location and Design, Institute of Transportation and Traffic Engineering, University of Califonia, Berkeley, August 1958.
5. Horonjeff, R., Grassi, R.C., Read, R.R., and Ahlbom, G.; A Mathematical Model for Locating Exit Taxiways, Institute of Transportation and Traffic Engineering, University of Califomia, Berkeley, December 1959.
6. Horonjeff, R., Read, R.R., and Ahlbom, G.; Exit Taxiway Locations, Institute of Transportation and Traffic Engineering, University of Califonia, Berkeley, September 1960.
7. Federal Aviation Administration, Airport Design, FAA/AC 150/5300-13, September 1989.
8. Schoen, M.L. et al, Probabilistic Computer Model of Optimal Runway Turnoffs, NASA Contractor Report 172549, April 1985.
9. Carr, H., P. Reaveley, and L.B. Smith IV, "New Turnoffs for Optimal Runway Occupancy Times", Airport Forum, No. 3, 1980, pp. 21-26.
10. Federal Aviation Administration, Airport Design, FAA/AC 150/5300-13A, February 2014.
11. Daellenbach, H. G. (1974). Dynamic programming model for optimal location of runway exits. Transportation Research, 8(3), 225-232.
12. REDIM 2.0 User's Manual

13. Agatonovic-Kustrin, S., & Beresford, R. (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. Journal of pharmaceutical and biomedical analysis, 22(5), 717-727.

14. Joline, E.S., "Optimization of Runway Exit Configurations", Transportation Engineering Journal, American Society of Civil Engineers, Vol. 100, February 1974, pp. 85-102.

15. Aerodrome Design Manual (Part 2)

Appendix – A – FLOWCHART OF WORK



Appendix – B – Table Containing Number of Standard High-Speed Exits, Non-Standard High-Speed Exits and Right Angle Exits for all Runways at Airports Analyzed

Airport	Runway	Number of Standard High-Speed Exits	Number of Non-Standard High-Speed Exits	Number of Right Angle Exits
ATL	08L	2	0	5
ATL	08R	0	1	8
ATL	09L	1	0	10
ATL	09R	2	0	3
ATL	10	2	0	2
ATL	26L	0	3	7
ATL	26R	2	0	6
ATL	27L	2	0	4
ATL	27R	0	1	10
ATL	28	2	0	0
BDL	1	0	0	1
BDL	6	0	0	9
BDL	15	0	0	10
BDL	19	0	0	2
BDL	24	0	0	7
BDL	33	0	0	10
BOS	04L	0	0	6

BOS	04R	2	1	6
BOS	9	0	0	2
BOS	14	0	0	0
BOS	15L	0	0	0
BOS	15R	0	0	0
BOS	22L	0	1	5
BOS	22R	0	0	5
BOS	27	0	1	0
BOS	32	0	0	0
BOS	33L	1	1	2
BOS	33R	0	0	0
BWI	10	0	0	8
BWI	15L	0	0	5
BWI	15R	0	0	6
BWI	28	0	0	7
BWI	33L	0	0	5
BWI	33R	0	1	3
CLE	06L	0	0	2
CLE	06R	0	0	5
CLE	10	0	0	1
CLE	24L	0	0	10
CLE	24R	0	0	2

CLE	28	0	0	1
CLT	5	0	0	6
CLT	18C	3	0	12
CLT	18L	2	0	11
CLT	18R	2	0	2
CLT	23	1	0	4
CLT	36C	2	0	12
CLT	36L	2	0	2
CLT	36R	1	1	10
DCA	1	0	2	4
DCA	4	0	2	0
DCA	15	0	3	0
DCA	19	0	2	5
DCA	22	0	2	2
DCA	33	0	2	1
DEN	7	1	0	3
DEN	8	1	0	4
DEN	16L	2	0	6
DEN	16R	2	0	6
DEN	17L	1	0	3
DEN	17R	2	0	6
DEN	25	1	0	3

DEN	26	2	0	4
DEN	34L	4	0	6
DEN	34R	2	0	8
DEN	35L	2	0	4
DEN	35R	2	0	3
DFW	13L	0	0	2
DFW	13R	1	0	4
DFW	17C	5	0	16
DFW	17L	3	0	3
DFW	17R	3	0	23
DFW	18L	3	0	22
DFW	18R	4	0	12
DFW	31L	1	0	3
DFW	31R	2	0	5
DFW	35C	3	0	15
DFW	35L	3	0	23
DFW	35R	3	0	3
DFW	36L	3	0	13
DFW	36R	3	0	23
DTW	03L	0	0	4
DTW	03R	3	0	4
DTW	04L	0	0	2

DTW	04R	0	0	9
DTW	09L	0	0	5
DTW	09R	0	0	2
DTW	21L	1	0	4
DTW	21R	0	0	3
DTW	22L	1	0	8
DTW	22R	0	0	1
DTW	27L	0	0	2
DTW	27R	1	0	4
EWR	04L	2	0	12
EWR	04R	2	1	5
EWR	11	0	0	8
EWR	22L	3	0	2
EWR	22R	1	1	16
EWR	29	0	2	8
FLL	10L	2	2	9
FLL	10R	0	0	4
FLL	28L	1	0	5
FLL	28R	1	0	7
HNL	04L	0	0	4
HNL	04R	1	0	6
HNL	08L	0	4	4

HNL	08R	1	0	2
HNL	22L	0	1	5
HNL	22R	0	2	4
HNL	26L	1	0	2
HNL	26R	0	0	6
HOU	4	0	0	9
HOU	13L	0	0	10
HOU	13R	0	0	8
HOU	17	0	0	6
HOU	22	0	0	10
HOU	31L	0	0	6
HOU	31R	0	0	7
HOU	35	0	0	5
IAD	01C	3	0	7
IAD	01L	2	0	2
IAD	01R	3	0	2
IAD	12	2	0	3
IAD	19C	3	0	7
IAD	19L	3	0	2
IAD	19R	2	0	2
IAD	30	2	0	1
IAH	08L	2	0	2

IAH	08R	3	0	3
IAH	9	2	0	1
IAH	15L	3	1	5
IAH	15R	2	0	7
IAH	26L	3	0	5
IAH	26R	2	0	2
IAH	27	2	0	1
IAH	33L	2	0	7
IAH	33R	3	0	6
JFK	04L	0	1	15
JFK	04R	2	1	0
JFK	13L	0	0	7
JFK	13R	0	2	11
JFK	22L	1	1	0
JFK	22R	0	0	9
JFK	31L	0	1	6
JFK	31R	0	1	8
LAS	01L	0	0	9
LAS	01R	1	0	5
LAS	07L	0	0	10
LAS	07R	2	0	1
LAS	19L	1	0	4

LAS	19R	2	0	6
LAS	25L	3	0	1
LAS	25R	0	0	10
LAX	06L	0	0	2
LAX	06R	1	1	8
LAX	07L	1	1	13
LAX	07R	2	0	6
LAX	24L	0	1	6
LAX	24R	2	0	2
LAX	25L	5	0	4
LAX	25R	0	0	8
LGA	4	0	2	7
LGA	13	0	3	2
LGA	22	0	1	8
LGA	31	1	1	2
MCO	17L	0	0	2
MCO	17R	0	0	6
MCO	18L	0	0	5
MCO	18R	0	0	5
MCO	35L	0	0	7
MCO	35R	0	0	2
MCO	36L	0	0	7

MCO	36R	0	0	5
MDW	04L	0	0	7
MDW	04R	0	0	13
MDW	13C	0	0	6
MDW	13L	0	0	3
MDW	13R	0	0	4
MDW	22L	0	1	12
MDW	22R	0	0	6
MDW	31C	0	1	6
MDW	31L	0	0	7
MDW	31R	0	0	4
MEM	9	0	0	9
MEM	18C	2	0	14
MEM	18L	0	0	7
MEM	18R	2	0	4
MEM	27	1	0	10
MEM	36C	2	0	14
MEM	36L	1	0	4
MEM	36R	3	0	7
MIA	08L	0	0	5
MIA	08R	0	2	8
MIA	9	0	0	2

MIA	12	0	3	1
MIA	26L	0	1	10
MIA	26R	0	0	4
MIA	27	0	0	3
MIA	30	0	0	3
MKE	01L	1	0	5
MKE	01R	0	0	3
MKE	07L	0	0	1
MKE	07R	0	0	5
MKE	13	0	0	1
MKE	19L	0	0	2
MKE	19R	0	0	4
MKE	25L	0	0	5
MKE	25R	0	1	1
MKE	31	0	0	1
MSP	4	0	0	5
MSP	12L	0	0	3
MSP	12R	0	0	15
MSP	17	0	0	6
MSP	22	0	0	6
MSP	30L	0	0	16
MSP	30R	0	0	2

MSP	35	0	0	5
ORD	15	0	0	2
ORD	33	0	1	0
ORD	04L	0	0	0
ORD	04R	1	0	0
ORD	09L	1	0	1
ORD	09R	0	0	7
ORD	10C	3	0	10
ORD	10L	0	1	9
ORD	10R	2	0	1
ORD	22L	0	2	1
ORD	22R	0	0	1
ORD	27L	0	1	5
ORD	27R	1	0	1
ORD	28C	2	0	12
ORD	28L	2	0	1
ORD	28R	0	2	12
PHL	8	0	0	1
PHL	09L	1	0	7
PHL	09R	4	0	9
PHL	17	0	0	5
PHL	26	0	1	4

PHL	27L	2	0	6
PHL	27R	2	0	7
PHL	35	0	1	5
PHX	07L	0	0	16
PHX	07R	0	0	5
PHX	8	0	0	20
PHX	25L	0	0	9
PHX	25R	0	0	15
PHX	26	0	0	20
PVD	5	0	0	6
PVD	16	0	0	3
PVD	23	0	0	5
PVD	34	0	0	2
SAN	9	0	0	7
SAN	27	0	2	7
SDF	11	0	0	4
SDF	17L	0	0	5
SDF	17R	0	0	9
SDF	29	1	0	2
SDF	35L	0	0	9
SDF	35R	0	0	5
SEA	16C	3	0	5

SEA	16L	0	0	7
SEA	16R	3	0	1
SEA	34C	3	0	8
SEA	34L	3	0	1
SEA	34R	0	1	10
SFO	01L	0	0	10
SFO	01R	0	0	12
SFO	10L	0	0	16
SFO	10R	0	0	13
SFO	19L	0	0	12
SFO	19R	0	0	13
SFO	28L	0	0	16
SFO	28R	0	0	14
SLC	14	0	0	0
SLC	16L	3	0	5
SLC	16R	0	0	2
SLC	17	1	0	5
SLC	32	0	0	0
SLC	34L	0	0	2
SLC	34R	3	0	6
SLC	35	2	1	3
SNA	02L	0	0	8

SNA	02R	0	0	6
SNA	20L	0	0	4
SNA	20R	0	0	8
STL	6	0	0	2
STL	11	2	0	0
STL	12L	1	0	8
STL	12R	0	0	6
STL	24	0	0	3
STL	29	2	0	2
STL	30L	0	0	4
STL	30R	1	0	6

Appendix – C – Source Code

Exitgeometrycode.m

```
% code to parse navid's raw data and gives required fields

% The input for this code is Navid's Raw Data. Using this Raw Data, this
% code only selects the fields required from the raw data.

%The other input file is the airport exit database for single airport. This
%database is essentially made from the consolidated exit database in
%sharepoint. An additional field called runway length is added to this.

%The most essential field it selects is the exit name. Based on the exit
%name selected, this code goes through the exit database and extracts the
%corresponding exit geometry parameters

%The final output is a consolidated excel file having each operation
%associated with exit geometry parameters and exit speeds at the point of
%curvature.

clc
clear

%Enter the airport code as input for temp variable

temp = 'SEA';

%adding path to the exit database
addpath('/Users/atsl/Documents/ATSL_Git/ATSL_Git/REDIM/Runway_Exit_SpeedScripts/Exit Database');

Airport_ExitData = importdata( 'SEA.xls' );

%Runway entries extracted from the SEA exit database. Note: They are not
%unique. They are all the entries in the exit database.
Runway_entries_extracted = Airport_ExitData.textdata( 2:end,2 );

%Runway exits extracted from the SEA exit database. These correspond to the
%extracted runway entries.
Runway_exits_extracted = Airport_ExitData.textdata( 2:end,3 );

%finding the number of runway entries which is equal to number of exits in
%SEA
Number_of_exits = length( Runway_entries_extracted );


%adding path to Raw data from Navid
addpath('/Users/atsl/Documents/REDIM/Raw Data from Navid/SEA');

%% Removing the redundant file names in the directory

directory= dir('/Users/atsl/Documents/REDIM/Raw Data from Navid/SEA'); % saving the directory information in a struct file

Index_of_deleted_files=[]; % index of deleted files

%finding the number of entries for a month
Number_of_entries_directory = length( directory );

%Removing entries with names '.' , '._' and '..'%

random = 1;

for i=1 : Number_of_entries_directory

    if( strcmp( char( directory( i ).name( 1 ) ),'.' ) == 1 )
```

```
        Index_of_deleted_files=[ Index_of_deleted_files,i ];

    end

end

directory( Index_of_deleted_files )=[];

%updating the number of entries in the directory

Number_of_entries_directory = length( directory );

%creating a directory to save the data for further analysis
mkdir ( '/Users/atsl/Documents/REDIM/Data', temp );

%% Loading the multiple files using the outer loop

%creating a new file for SEA data

% measuring the progress by introducing a variable number of flights
Number_of_flights_processed = 0;

for m = 1 : Number_of_entries_directory
    %% loading the files. Obtaining the file names from the directory

    load( directory(m).name );

    Index_of_incomplete_entries = []; % Index of incomplete entries to delete them

    Length_Data = length( Airport_Report_File_Pure ); % the length of a single mat file (for one day) in the raw data sent by Navid

    Index_of_incomplete_entries =  find([Airport_Report_File_Pure.Flag_For_unreasonable_results] ~= 0);

    Airport_Report_File_Pure( Index_of_incomplete_entries ) = [];

    %after removing the incomplete entries, the necessary fields are extracted
    %using the following for loop.

    %%  before using the for loop to extract necessary fields, updating the
    % length of the data file

    Length_Data = length( Airport_Report_File_Pure );



    for i = 1: Length_Data

        check_i = [];

        %temporary vaiable to measure the bleed rate along the turnoff
        temporary_IndexExitHoldbar = Airport_Report_File_Pure( i ).IndexExitHoldBar;

        temporary_Index_Turn_Manuever = Airport_Report_File_Pure( i ).Index_Turn_Manuover;

        %temporary variable introduced to measure the change in angle from the
        %runway centreline.
        temporary_Index_Exit = Airport_Report_File_Pure( i ).IndexExit;

        %Number of indexes between PC and Runway Exit
        turn_to_exit_indexes = temporary_Index_Exit - temporary_Index_Turn_Manuever;

        %Number of indexes between Runway exit to holdbar
        turn_exit_holdbar = temporary_IndexExitHoldbar - temporary_Index_Exit;

        if  ( ( turn_to_exit_indexes > 0 ) && ( turn_exit_holdbar > 0 ) )

            Data( i ).Operational_Runway = Airport_Report_File_Pure( i ).Operational_Runway;

            Data( i ).Aircraft_Type = Airport_Report_File_Pure( i ).Aircraft_Type;
```

```
        Data( i ).Exit_Name = Airport_Report_File_Pure( i ).Exit_Name;

        temporary_Index_Turn_Manuever = Airport_Report_File_Pure( i ).Index_Turn_Manuover;

        Data( i ).Exit_Speed_knots = Airport_Report_File_Pure( i ).Exiting_Speed_Knots;

        %temporary variable introduced to measure the change in angle from the
        %runway centreline.
        temporary_Index_Exit = Airport_Report_File_Pure( i ).IndexExit;

        %Calculating the time taken to maneuver
        Data( i ).time_to_maneuver = Airport_Report_File_Pure( i ).Time_Seconds_After_Midnight( temporary_IndexExitHoldbar ) -
Airport_Report_File_Pure( i ).Time_Seconds_After_Midnight( temporary_Index_Turn_Manuever );

        %plotting the

        %Calculating the bleed rate (m/s2)
        Data( i ).Bleed_rate_m_ss = ( Airport_Report_File_Pure( i ).Speed_Smoothed_m_s( temporary_IndexExitHoldbar ) -
Airport_Report_File_Pure( i ).Speed_Smoothed_m_s( temporary_Index_Turn_Manuever ) ) / Data( i ).time_to_maneuver;


        %Storing all the lat coordinates in a temporary array for easy access
        latitudes = Airport_Report_File_Pure( i ).Latitude_all_Points;

        %Storing all the long coSEAinates in a temporary array for easy access
        longitudes = Airport_Report_File_Pure( i ).Longitude_all_Points;

        %temporary variable introduced to store the turn index. Makes code
        %easier by not influencing the actual variable.
        temporary = temporary_Index_Turn_Manuever;


        % for loop to measure the azimuths of all consecutive points from the PC
        % to the runway exit.
        for z = 1 : turn_to_exit_indexes

            angle ( z ) = azimuth ( latitudes ( temporary ), longitudes ( temporary ) ,  latitudes ( temporary + 1 ), longitudes ( temporary
+ 1 ) );

            temporary = temporary +1;

        end

        %array having values of difference in consecutive array values
        difference_in_angles = diff( angle );

        %Sum of all the differences to see cumulative change in angle from PC to
        %the runway exit.
        Sum_differences_in_angle = sum( difference_in_angles );

        Data( i ).Change_in_azimuth_PC_Polygon_Exit = Sum_differences_in_angle;

        %Number of indexes between Runway exit to holdbar
        turn_exit_holdbar = temporary_IndexExitHoldbar - temporary_Index_Exit;

        temporary_2 = temporary_Index_Exit;

        %for loop to measure the azimuths of all consecutive points from runway
        %exit edge to the holdbar
        for y = 1 : turn_exit_holdbar

            angle_2 ( y ) = azimuth ( latitudes ( temporary_2 ), longitudes ( temporary_2 ) ,  latitudes ( temporary_2 + 1 ), longitudes (
temporary_2 + 1 ) );

            temporary_2 = temporary_2 + 1;

        end

        %array having values of difference in consecutive array values
```

```
        difference_in_angles_2 = diff( angle_2 );

        %Sum of all the differences to see cumulative change in angle from
        %runway exit edge to the holdbar.
        Sum_differences_in_angle_2 = sum( difference_in_angles_2 );

        %emptying the angle array
        angle_2 = [];

        Data( i ).Change_in_azimuth_exit_holdbar = Sum_differences_in_angle_2;



        Data( i ).Total_change_heading = abs ( Data( i ).Change_in_azimuth_PC_Polygon_Exit + Data( i
).Change_in_azimuth_exit_holdbar ) ;

        if ( Data( i ).Total_change_heading > 200 )

          Data( i ).Total_change_heading = 360 - Data( i ).Total_change_heading;

        end


        for j = 1 : Number_of_exits

          if( ( strcmp ( Runway_entries_extracted( j ) , ( Data( i ).Operational_Runway ) ) == 1 ) && ( strcmp ( Runway_exits_extracted{
j }, Data(i).Exit_Name) == 1))

            %Angle made by the first consecutive points with the runway centreline.
            first_consecutive_points_angle_with_runway_centreline = abs( angle ( 1 ) - Airport_ExitData.data( j, 7 ) );

            %making angle an empty array to avoid previous operations affecting the
            %current ones.
            angle =[];


            if ( first_consecutive_points_angle_with_runway_centreline > 200 )

              first_consecutive_points_angle_with_runway_centreline = 360 -
first_consecutive_points_angle_with_runway_centreline ;

            end

            Data( i ).Total_change_heading = Data( i ).Total_change_heading +
first_consecutive_points_angle_with_runway_centreline;

            %adding exit angle parameter
            Data( i ).ExitAngle = Airport_ExitData.data( j, 2 );

            %adding path length parameter
            Data( i ).Path_Length_ft = Airport_ExitData.data( j, 3 );

            %adding holdbar distance parameter
            Data( i ).Holdbar_distance_ft = Airport_ExitData.data( j, 9 );

            %adding closest taxiway distance parameter
            Data( i ).Closest_Taxiway_Distance_ft = Airport_ExitData.data( j, 11 );

            %adding taxiway width parameter
            Data( i ).Taxi_Width_ft = Airport_ExitData.data( j,13 );

            %adding distance from threshold parameter
            Data( i ).Distance_from_Threshold_ft = Airport_ExitData.data( j, 1 );

            %adding Radius parameter
            Data( i ).Radius_ft = Airport_ExitData.data( j,12 );

            %adding remaining distance ( Remaining distance = Runway
            %length - Distance of exit from threshold along runway
```

```
            %centreline )
            Data( i ).Remaining_Distance_ft = Airport_ExitData.data( j, 14 );

            % Measuring Runway Angle
            Data( i ).RunwayAngle = Airport_ExitData.data( j, 7);

            %defining a temporary left or right turn exit to calculate
            %the exit engle of the operation
            temp_leftorRight = Airport_ExitData.data( j, 6);


          end % end of loop which checks the data entries with the exit database


        end %end of for loop involved in extracting data. This for loop creates one recSEA. One recSEA in each file.


        Number_of_flights_processed = Number_of_flights_processed + 1;

        %establishing the validity of the record by adding a variable.

        Data(i).Validity = 1;

        if (Data( i ).ExitAngle > 0)

          Data(i).Validity = 1;

        else

          Data(i).Validity = 0;

        end

    else

        Data(i).Validity = 0;


    end % end of if statement checking whether indexes of turn manuever, runway polygon exit and holdbar are in SEAer.


end %this loop parses the data for one whole month


if ( length ( Data ) > 0 )

    %Removing records with unreasonable results
    Invalid_entries = find([Data.Validity] == 0);

    Data(Invalid_entries) = [];


    %Obtaining the directory of the files
    dir='/Users/atsl/Documents/REDIM/Data';

    filename = directory(m).name ;

    filename_add=[dir '/' temp '/' filename];

    save(filename_add,'Data');

    filename_consolidated = [ '/Users/atsl/Documents/REDIM/Data/consolidated' '/' filename];

    save(filename_consolidated,'Data');

    Data=struct([]);
    % saving the mat files of each day operations in the file called Data.

    disp ( 'Number_of_flights_processed =' ) ; disp ( Number_of_flights_processed );
```

```
    end

end

% the following code deals with making a consolidated table having all the
% operations at airport of interest. This consolidated table will be the
% input file to the next code :  ADG_parsing.

% adding path to the data just parsed
addpath ('/Users/atsl/Documents/REDIM/Data/SEA');

% loading the first mat file in the directory.
load ( directory( 1 ).name );

%Creating a consolidated table. Using for loop, all the data is eventually
%added into this table.
Consolidated_table = struct2table( Data );

for i = 1 : Number_of_entries_directory

    load ( directory( i ).name );

    temp_table = struct2table( Data );

    Consolidated_table = outerjoin ( Consolidated_table, temp_table, 'MergeKeys', true);

end


%creating a new directory to save the consolidated data
mkdir ( '/Users/atsl/Documents/REDIM/Final Data', temp );

%Saving the consolidated table in a new directory
dir = '/Users/atsl/Documents/REDIM/Final Data';

%temporary file name
filename = [ dir '/' temp '/Final Data'];

filename_add = [filename '.csv'];

writetable ( Consolidated_table , filename_add );
```

ADG_Parsing.m

```
%% Script to segregate the data into ADG category

% This code takes the input from exit geometry code.

% The input to this code is the Final Data excel sheet which has records of
% all the observations that happened at the airport. These observations
% need to be segregated into different ADG groups. This helps do further
% analysis over each ADG group.

% Other input for this code is the Aircraft Table which has information
% over which ADG group does an aircraft belong to

clc
clear

temp = 'LAS';
%adding the path to the final data
addpath('/Users/atsl/Documents/REDIM/Final Data/LAS');
Operations_data = importdata('Final Data.csv');

%Calculating the number of operations to use in for loop

Number_of_Operations = length(Operations_data.textdata) ;

% Extracting all the aircrafts in the operations to find the unique
```

```
% aircrafts

for i = 2 : Number_of_Operations

    Aircrafts{i-1,1} = Operations_data.textdata{i,2};

end

%importing the aircraft database having AAC and ADG information

addpath ('/Users/atsl/Documents/ATSL_Git/ATSL_Git/REDIM/Runway_Exit_SpeedScripts');

ADG_data = importdata('Aircraft_Table.xlsx');

Number_of_Aircraft_types = length(ADG_data.textdata);

% Extracting the data of AAC for all different aicrafts

for i = 2 : Number_of_Aircraft_types

    ADG{i-1,1} = ADG_data.textdata{i,8};

end

%% Finding all Aircrafts belonging to different AAC and categorising them
% into different arrays.

%finding the index of aircrafts with ADG = 1

Index_1=find( strcmp( '1', ADG))+1;

% Number of Aircrafts in category 1

Number_Aircrafts_Category_1 = length( Index_1 );

for i = 1 : Number_Aircrafts_Category_1

ADG_1 {i,1}= ADG_data.textdata{ Index_1(i),1};

end

%finding the index of aircrafts with ADG = 2

Index_2=find( strcmp( '2', ADG))+1;

% Number of Aircrafts in category 2

Number_Aircrafts_Category_2 = length( Index_2 );

for i = 1 : Number_Aircrafts_Category_2

ADG_2 {i,1}= ADG_data.textdata{ Index_2(i),1};

end

%finding the index of aircrafts with ADG = 3

Index_3=find( strcmp( '3', ADG))+1;

% Number of Aircrafts in category 3

Number_Aircrafts_Category_3 = length( Index_3 );

for i = 1 : Number_Aircrafts_Category_3

ADG_3 {i,1}= ADG_data.textdata{ Index_3(i),1};

end

%finding the index of aircrafts with ADG =4
```

```
Index_4=find( strcmp( '4', ADG))+1;

% Number of Aircrafts in category 4

Number_Aircrafts_Category_4 = length( Index_4 );

for i = 1 : Number_Aircrafts_Category_4

ADG_4 {i,1}= ADG_data.textdata{ Index_4(i),1};

end

%finding the index of aircrafts with ADG =5

Index_5=find( strcmp( '5', ADG))+1;

% Number of Aircrafts in category 5

Number_Aircrafts_Category_5 = length( Index_5 );

for i = 1 : Number_Aircrafts_Category_5

ADG_5 {i,1}= ADG_data.textdata{ Index_5(i),1};

end

%finding the index of aircrafts with ADG =6

Index_6=find( strcmp( '6', ADG))+1;

% Number of Aircrafts in category 5

Number_Aircrafts_Category_6 = length( Index_6 );

for i = 1 : Number_Aircrafts_Category_6

ADG_6 {i,1}= ADG_data.textdata{ Index_6(i),1};

end
%% Matching the aircraft categories obtained with the main data.

Index_Data_1=[];
Index_Data_2=[];
Index_Data_3=[];
Index_Data_4=[];
Index_Data_5=[];
Index_Data_6=[];

%loop used to find the index of aircrafts belonging to ADG I
for i =1: Number_Aircrafts_Category_1

   for j = 1 : (Number_of_Operations - 1 )

      if ( strcmp( ADG_1{i,1},Aircrafts{j,1} )==1 )

         Index_Data_1=[ Index_Data_1; j ];
      end
   end
end

% Matching the aircrafts from ADG_2

for i=1: Number_Aircrafts_Category_2

   for j = 1 : (Number_of_Operations - 1 )

      if ( strcmp( ADG_2{i,1},Aircrafts{j,1} ) ==1 )

         Index_Data_2=[ Index_Data_2; j ];
```

```
        end
    end
end

% Matching the aircrafts from ADG_3

for i=1: Number_Aircrafts_Category_3

    for j = 1 : (Number_of_Operations - 1 )

        if ( strcmp( ADG_3{i,1},Aircrafts{j,1} )==1 )

            Index_Data_3=[ Index_Data_3; j ];
        end
    end
end

% Matching the aircrafts from ADG_4

for i = 1: Number_Aircrafts_Category_4

    for j = 1 : (Number_of_Operations - 1 )

        if ( strcmp( ADG_4{i,1},Aircrafts{j,1} )==1 )

            Index_Data_4=[ Index_Data_4; j ];
        end
    end
end

% Matching the aircrafts from ADG_5

for i = 1: Number_Aircrafts_Category_5

    for j = 1 : (Number_of_Operations - 1 )

        if ( strcmp( ADG_5{i,1},Aircrafts{j,1} )==1 )

            Index_Data_5=[ Index_Data_5; j ];
        end
    end
end

% Matching the aircrafts from ADG_6

for i = 1: Number_Aircrafts_Category_6

    for j = 1 : (Number_of_Operations - 1 )

        if ( strcmp( ADG_6{i,1},Aircrafts{j,1} )==1 )

            Index_Data_6=[ Index_Data_6; j ];
        end
    end
end

%% Creating Data Tables for each of these categories

%Creating a directory to save the data


Number_Aircrafts_ADG_1_in_Data = length ( Index_Data_1 );

for i = 1 : Number_Aircrafts_ADG_1_in_Data

Data_ADG_1(i).Operational_Runway = Operations_data.textdata { (Index_Data_1( i,1 )) + 1 , 1};

Data_ADG_1(i).Aircraft_Type = Operations_data.textdata { (Index_Data_1( i,1 )) + 1 , 2};

Data_ADG_1(i).Exit_Name = Operations_data.textdata { (Index_Data_1( i,1 )) + 1 , 3};
```

```
Data_ADG_1(i).Exit_Speed_kts = Operations_data.data ( (Index_Data_1( i,1 )) , 1);
Data_ADG_1(i).Bleed_rate_m_ss = Operations_data.data ( (Index_Data_1( i,1 )) , 3);
Data_ADG_1(i).time_to_manuever_from_PC_to_holdbar_sec= Operations_data.data ( (Index_Data_1( i,1 )) , 2);
Data_ADG_1(i).Change_in_azimuth_PC_Polygon_Exit = Operations_data.data( (Index_Data_1( i,1 )) , 4);
Data_ADG_1(i).Change_azimuth_exit_holdbar = Operations_data.data ( (Index_Data_1( i,1 )) , 5);
Data_ADG_1(i).Total_change_heading= Operations_data.data ( (Index_Data_1( i,1 )) , 6);
Data_ADG_1(i).ExitAngle = Operations_data.data ( (Index_Data_1( i,1 )) , 7);
Data_ADG_1(i).Path_length_ft = Operations_data.data ( (Index_Data_1( i,1 )) , 8);
Data_ADG_1(i).Holdbar_distance_ft = Operations_data.data ( (Index_Data_1( i,1 )) , 9);
Data_ADG_1(i).Closest_taxiway_ft = Operations_data.data ( (Index_Data_1( i,1 )) , 10);
Data_ADG_1(i).Taxiway_width_ft = Operations_data.data ( (Index_Data_1( i,1 )) , 11);
Data_ADG_1(i).Distance_from_threshold_ft = Operations_data.data ( (Index_Data_1( i,1 )) , 12);
Data_ADG_1(i).Radius_ft = Operations_data.data ( (Index_Data_1( i,1 )) , 13);
Data_ADG_1(i).Remaining_Distance_ft = Operations_data.data ( (Index_Data_1( i,1 )) , 14);
Data_ADG_1(i).RunwayAngle = Operations_data.data ( ( Index_Data_1( i,1 )) , 15);
Data_ADG_1(i).Airport = temp;

end
% Creating a struct file for ADG_2 data.
Number_Aircrafts_ADG_2_in_Data = length ( Index_Data_2 );
for i = 1 : Number_Aircrafts_ADG_2_in_Data
Data_ADG_2(i).Operational_Runway = Operations_data.textdata { (Index_Data_2( i,1 )) + 1 , 1};
Data_ADG_2(i).Aircraft_Type = Operations_data.textdata { (Index_Data_2( i,1 )) + 1 , 2};
Data_ADG_2(i).Exit_Name = Operations_data.textdata { (Index_Data_2( i,1 )) + 1 , 3};
Data_ADG_2(i).Exit_Speed_kts = Operations_data.data ( (Index_Data_2( i,1 )) , 1);
Data_ADG_2(i).Bleed_rate_m_ss = Operations_data.data ( (Index_Data_2( i,1 )) , 3);
Data_ADG_2(i).time_to_manuever_from_PC_to_holdbar_sec= Operations_data.data ( (Index_Data_2( i,1 )) , 2);
Data_ADG_2(i).Change_in_azimuth_PC_Polygon_Exit = Operations_data.data( (Index_Data_2( i,1 )) , 4);
Data_ADG_2(i).Change_azimuth_exit_holdbar = Operations_data.data ( (Index_Data_2( i,1 )) , 5);
Data_ADG_2(i).Total_change_heading= Operations_data.data ( (Index_Data_2( i,1 )) , 6);
Data_ADG_2(i).ExitAngle = Operations_data.data ( (Index_Data_2( i,1 )) , 7);
Data_ADG_2(i).Path_length_ft = Operations_data.data ( (Index_Data_2( i,1 )) , 8);
Data_ADG_2(i).Holdbar_distance_ft = Operations_data.data ( (Index_Data_2( i,1 )) , 9);
Data_ADG_2(i).Closest_taxiway_ft = Operations_data.data ( (Index_Data_2( i,1 )) , 10);
Data_ADG_2(i).Taxiway_width_ft = Operations_data.data ( (Index_Data_2( i,1 )) , 11);
```

```
Data_ADG_2(i).Distance_from_threshold_ft = Operations_data.data ( (Index_Data_2( i,1 )) , 12);
Data_ADG_2(i).Radius_ft = Operations_data.data ( (Index_Data_2( i,1 )) , 13);
Data_ADG_2(i).Remaining_Distance_ft = Operations_data.data ( (Index_Data_2( i,1 )) , 14);
Data_ADG_2(i).RunwayAngle = Operations_data.data ( ( Index_Data_2( i,1 )) , 15);

Data_ADG_2(i).Airport = temp;

end
% Creating a database having recLASs of ADG 3.

Number_Aircrafts_ADG_3_in_Data = length ( Index_Data_3 );
if (Number_Aircrafts_ADG_3_in_Data >0)
  for i = 1 : Number_Aircrafts_ADG_3_in_Data
Data_ADG_3(i).Operational_Runway = Operations_data.textdata { (Index_Data_3( i,1 )) + 1 , 1};
Data_ADG_3(i).Aircraft_Type = Operations_data.textdata { (Index_Data_3( i,1 )) + 1 , 2};
Data_ADG_3(i).Exit_Name = Operations_data.textdata { (Index_Data_3( i,1 )) + 1 , 3};
Data_ADG_3(i).Exit_Speed_kts = Operations_data.data ( (Index_Data_3( i,1 )) , 1);
Data_ADG_3(i).Bleed_rate_m_ss = Operations_data.data ( (Index_Data_3( i,1 )) , 3);
Data_ADG_3(i).time_to_manuever_from_PC_to_holdbar_sec= Operations_data.data ( (Index_Data_3( i,1 )) , 2);
Data_ADG_3(i).Change_in_azimuth_PC_Polygon_Exit = Operations_data.data( (Index_Data_3( i,1 )) , 4);
Data_ADG_3(i).Change_azimuth_exit_holdbar = Operations_data.data ( (Index_Data_3( i,1 )) , 5);
Data_ADG_3(i).Total_change_heading= Operations_data.data ( (Index_Data_3( i,1 )) , 6);
Data_ADG_3(i).ExitAngle = Operations_data.data ( (Index_Data_3( i,1 )) , 7);
Data_ADG_3(i).Path_length_ft = Operations_data.data ( (Index_Data_3( i,1 )) , 8);
Data_ADG_3(i).Holdbar_distance_ft = Operations_data.data ( (Index_Data_3( i,1 )) , 9);
Data_ADG_3(i).Closest_taxiway_ft = Operations_data.data ( (Index_Data_3( i,1 )) , 10);
Data_ADG_3(i).Taxiway_width_ft = Operations_data.data ( (Index_Data_3( i,1 )) , 11);
Data_ADG_3(i).Distance_from_threshold_ft = Operations_data.data ( (Index_Data_3( i,1 )) , 12);
Data_ADG_3(i).Radius_ft = Operations_data.data ( (Index_Data_3( i,1 )) , 13);
Data_ADG_3(i).Remaining_Distance_ft = Operations_data.data ( (Index_Data_3( i,1 )) , 14);
Data_ADG_3(i).RunwayAngle = Operations_data.data ( ( Index_Data_3( i,1 )) , 15);

Data_ADG_3(i).Airport = temp;

  end

end
```

```
% Creating a database of all the recLASs with aircrafts belonging to ADG_4
Number_Aircrafts_ADG_4_in_Data = length ( Index_Data_4 );
if (Number_Aircrafts_ADG_4_in_Data >0)
   for i = 1 : Number_Aircrafts_ADG_4_in_Data
Data_ADG_4(i).Operational_Runway = Operations_data.textdata { (Index_Data_4( i,1 )) + 1 , 1};
Data_ADG_4(i).Aircraft_Type = Operations_data.textdata { (Index_Data_4( i,1 )) + 1 , 2};
Data_ADG_4(i).Exit_Name = Operations_data.textdata { (Index_Data_4( i,1 )) + 1 , 3};
Data_ADG_4(i).Exit_Speed_kts = Operations_data.data ( (Index_Data_4( i,1 )) , 1);
Data_ADG_4(i).Bleed_rate_m_ss = Operations_data.data ( (Index_Data_4( i,1 )) , 3);
Data_ADG_4(i).time_to_manuever_from_PC_to_holdbar_sec= Operations_data.data ( (Index_Data_4( i,1 )) , 2);
Data_ADG_4(i).Change_in_azimuth_PC_Polygon_Exit = Operations_data.data( (Index_Data_4( i,1 )) , 4);
Data_ADG_4(i).Change_azimuth_exit_holdbar = Operations_data.data ( (Index_Data_4( i,1 )) , 5);
Data_ADG_4(i).Total_change_heading= Operations_data.data ( (Index_Data_4( i,1 )) , 6);
Data_ADG_4(i).ExitAngle = Operations_data.data ( (Index_Data_4( i,1 )) , 7);
Data_ADG_4(i).Path_length_ft = Operations_data.data ( (Index_Data_4( i,1 )) , 8);
Data_ADG_4(i).Holdbar_distance_ft = Operations_data.data ( (Index_Data_4( i,1 )) , 9);
Data_ADG_4(i).Closest_taxiway_ft = Operations_data.data ( (Index_Data_4( i,1 )) , 10);
Data_ADG_4(i).Taxiway_width_ft = Operations_data.data ( (Index_Data_4( i,1 )) , 11);
Data_ADG_4(i).Distance_from_threshold_ft = Operations_data.data ( (Index_Data_4( i,1 )) , 12);
Data_ADG_4(i).Radius_ft = Operations_data.data ( (Index_Data_4( i,1 )) , 13);
Data_ADG_4(i).Remaining_Distance_ft = Operations_data.data ( (Index_Data_4( i,1 )) , 14);
Data_ADG_4(i).RunwayAngle = Operations_data.data ( ( Index_Data_4( i,1 )) , 15);
Data_ADG_4(i).Airport = temp;

   end

end
% Creating a database of all the recLASs having ADG_5 aircrafts
Number_Aircrafts_ADG_5_in_Data = length ( Index_Data_5 );
if (Number_Aircrafts_ADG_5_in_Data >0)
   for i = 1 : Number_Aircrafts_ADG_5_in_Data
Data_ADG_5(i).Operational_Runway = Operations_data.textdata { (Index_Data_5( i,1 )) + 1 , 1};
Data_ADG_5(i).Aircraft_Type = Operations_data.textdata { (Index_Data_5( i,1 )) + 1 , 2};
Data_ADG_5(i).Exit_Name = Operations_data.textdata { (Index_Data_5( i,1 )) + 1 , 3};
Data_ADG_5(i).Exit_Speed_kts = Operations_data.data ( (Index_Data_5( i,1 )) , 1);
Data_ADG_5(i).Bleed_rate_m_ss = Operations_data.data ( (Index_Data_5( i,1 )) , 3);
```

```
Data_ADG_5(i).time_to_manuever_from_PC_to_holdbar_sec= Operations_data.data ( (Index_Data_5( i,1 )) , 2);
Data_ADG_5(i).Change_in_azimuth_PC_Polygon_Exit = Operations_data.data( (Index_Data_5( i,1 )) , 4);
Data_ADG_5(i).Change_azimuth_exit_holdbar = Operations_data.data ( (Index_Data_5( i,1 )) , 5);
Data_ADG_5(i).Total_change_heading= Operations_data.data ( (Index_Data_5( i,1 )) , 6);
Data_ADG_5(i).ExitAngle = Operations_data.data ( (Index_Data_5( i,1 )) , 7);
Data_ADG_5(i).Path_length_ft = Operations_data.data ( (Index_Data_5( i,1 )) , 8);
Data_ADG_5(i).Holdbar_distance_ft = Operations_data.data ( (Index_Data_5( i,1 )) , 9);
Data_ADG_5(i).Closest_taxiway_ft = Operations_data.data ( (Index_Data_5( i,1 )) , 10);
Data_ADG_5(i).Taxiway_width_ft = Operations_data.data ( (Index_Data_5( i,1 )) , 11);
Data_ADG_5(i).Distance_from_threshold_ft = Operations_data.data ( (Index_Data_5( i,1 )) , 12);
Data_ADG_5(i).Radius_ft = Operations_data.data ( (Index_Data_5( i,1 )) , 13);
Data_ADG_5(i).Remaining_Distance_ft = Operations_data.data ( (Index_Data_5( i,1 )) , 14);
Data_ADG_5(i).RunwayAngle = Operations_data.data ( ( Index_Data_5( i,1 )) , 15);
Data_ADG_5(i).Airport = temp;

end
end
% Creating a database of ADG_6 aircraft recLASs
Number_Aircrafts_ADG_6_in_Data = length ( Index_Data_6 );
if (Number_Aircrafts_ADG_6_in_Data >0)
for i = 1 : Number_Aircrafts_ADG_6_in_Data
Data_ADG_6(i).Operational_Runway = Operations_data.textdata { (Index_Data_6( i,1 )) + 1 , 1};
Data_ADG_6(i).Aircraft_Type = Operations_data.textdata { (Index_Data_6( i,1 )) + 1 , 2};
Data_ADG_6(i).Exit_Name = Operations_data.textdata { (Index_Data_6( i,1 )) + 1 , 3};
Data_ADG_6(i).Exit_Speed_kts = Operations_data.data ( (Index_Data_6( i,1 )) , 1);
Data_ADG_6(i).Bleed_rate_m_ss = Operations_data.data ( (Index_Data_6( i,1 )) , 3);
Data_ADG_6(i).time_to_manuever_from_PC_to_holdbar_sec= Operations_data.data ( (Index_Data_6( i,1 )) , 2);
Data_ADG_6(i).Change_in_azimuth_PC_Polygon_Exit = Operations_data.data( (Index_Data_6( i,1 )) , 4);
Data_ADG_6(i).Change_azimuth_exit_holdbar = Operations_data.data ( (Index_Data_6( i,1 )) , 5);
Data_ADG_6(i).Total_change_heading= Operations_data.data ( (Index_Data_6( i,1 )) , 6);
Data_ADG_6(i).ExitAngle = Operations_data.data ( (Index_Data_6( i,1 )) , 7);
Data_ADG_6(i).Path_length_ft = Operations_data.data ( (Index_Data_6( i,1 )) , 8);
Data_ADG_6(i).Holdbar_distance_ft = Operations_data.data ( (Index_Data_6( i,1 )) , 9);
Data_ADG_6(i).Closest_taxiway_ft = Operations_data.data ( (Index_Data_6( i,1 )) , 10);
Data_ADG_6(i).Taxiway_width_ft = Operations_data.data ( (Index_Data_6( i,1 )) , 11);
Data_ADG_6(i).Distance_from_threshold_ft = Operations_data.data ( (Index_Data_6( i,1 )) , 12);
```

```
Data_ADG_6(i).Radius_ft = Operations_data.data ( (Index_Data_6( i,1 )) , 13);

Data_ADG_6(i).Remaining_Distance_ft = Operations_data.data ( (Index_Data_6( i,1 )) , 14);

Data_ADG_6(i).RunwayAngle = Operations_data.data ( ( Index_Data_6( i,1 )) , 15);

Data_ADG_6(i).Airport = temp;

end

end

if ( Number_Aircrafts_ADG_1_in_Data > 0 )

mkdir ( '/Users/atsl/Documents/REDIM/Validation/ADG_I Consolidated' );

dir = '/Users/atsl/Documents/REDIM/Validation/ADG_I Consolidated';

% giving a temporary file name
filename = [dir '/' temp];


%adding the format for the filename.
filename_add = [filename '.mat'];

%saving the table
save( filename_add , 'Data_ADG_1');

end

if ( Number_Aircrafts_ADG_2_in_Data > 0 )

mkdir ( '/Users/atsl/Documents/REDIM/Validation/ADG_II Consolidated' );

dir = '/Users/atsl/Documents/REDIM/Validation/ADG_II Consolidated';

%giving a temporary file name
filename = [dir '/' temp];

%adding the format for the filename.
filename_add = [filename '.mat'];

%saving the stuct
save( filename_add , 'Data_ADG_2');

end

if ( Number_Aircrafts_ADG_3_in_Data > 0 )

mkdir ( '/Users/atsl/Documents/REDIM/Validation/ADG_III Consolidated' );

dir = '/Users/atsl/Documents/REDIM/Validation/ADG_III Consolidated';

%giving a temporary file name
filename = [dir '/' temp];

%adding the format for the filename.
filename_add = [filename '.mat'];

%saving the struct
save( filename_add , 'Data_ADG_3');

end

if ( Number_Aircrafts_ADG_4_in_Data > 0 )

mkdir ( '/Users/atsl/Documents/REDIM/Validation/ADG_IV Consolidated' );

dir = '/Users/atsl/Documents/REDIM/Validation/ADG_IV Consolidated';
```

```
%giving a temporary file name
filename = [dir '/' temp];

%adding the format for the filename.
filename_add = [filename '.mat'];

%saving the struct
save( filename_add , 'Data_ADG_4');

end

if ( Number_Aircrafts_ADG_5_in_Data > 0 )

mkdir ( '/Users/atsl/Documents/REDIM/Validation/ADG_V Consolidated' );

dir = '/Users/atsl/Documents/REDIM/Validation/ADG_V Consolidated';

%giving a temporary file name
filename = [dir '/' temp];

%adding the format for the filename.
filename_add = [filename '.mat'];

%saving the struct
save( filename_add , 'Data_ADG_5');

end

if ( Number_Aircrafts_ADG_6_in_Data > 0 )

mkdir ( '/Users/atsl/Documents/REDIM/Validation/ADG_VI Consolidated' );

dir = '/Users/atsl/Documents/REDIM/Validation/ADG_VI Consolidated';

%giving a temporary file name
filename = [dir '/' temp];

%adding the format for the filename.
filename_add = [filename '.mat'];

%saving the struct
save( filename_add , 'Data_ADG_6');

end

%% end
```

Exits_info_removing_unnecessary_exits.m

```
%code to remove the exits having low number of observations. Even this code
%checks for bimodality in the exit angle distributions. However, more
%importantly, this code removes exits having low number of distributions.
%Even this code gives an exit database having record of each unique exit
%taken.

% Important job of this code - Standard Deviation is extracted for
% different exit types.

%this code does following things:
%1. Create an exit database having a record of all unique exits taken at
%all the airports.

%2. Finding the exits having low count and finding exits having potential bimodal
%distributions for exit angles.

%3. Cleaning the data. Removing the exits with low counts and saving the
%original mat files.

%4. Consolidates all the mat files to create a consolidated data table and
%save it in the specific ADG analysis folder.
```

```
% Normality test is 2 when there are not many observations


clc
clear


%% STEP -1

%use control-f and replace to ADG_I if you want the analysis to be done
%for ADG_I


addpath('/Users/atsl/Documents/REDIM/ADG_I Consolidated');

directory =  dir ('/Users/atsl/Documents/REDIM/ADG_I Consolidated');

Index_deleted_files = [];


% threshold to determine whether the number of obervations for an exit are
% low
threshold_minimum_observations = 15;


for i = 1 : length (directory)

    if ( strcmp ( char ( directory ( i ).name(1) ) , '.' ) == 1 )

        Index_deleted_files = [ Index_deleted_files, i ];

    end

end

directory ( Index_deleted_files ) = [];

Number_Airports = length ( directory );

m = 1; % Exit Data counter

normal_unique_exit_speed_index = []; % array to store m values corresponding to
%observations having a minimum of 100 observations and have normally
%distributed exit speeds.

n = 1; % Flag counter

l = 1; % counter for low observations

Exit_Data = struct([]);

exits_normal_exit_speed_distribution = struct ([]);

for i = 1 : Number_Airports

    Raw_Data = importdata ( directory ( i ).name );

    Data = struct2table ( Raw_Data );

    temp_runways = table2array ( Data ( :, 1) );

    %finding unique runways
    Unique_Runways = unique ( temp_runways );

    Number_Runway = length ( Unique_Runways );

    for j = 1 : Number_Runway

        Index_Runway = find ( strcmp ( Unique_Runways { j } , temp_runways ) );
```

```
temp_Runway_Data = Raw_Data ( Index_Runway );

Runway_Data = struct2table ( Raw_Data ( Index_Runway ) );

Exits_One_Runway = table2cell ( unique ( Runway_Data ( : , 3 ) ) );

% Number of unique exits taken on one runway
Number_Exits_One_Runway = length ( Exits_One_Runway );

temp_exits = table2array ( Runway_Data ( : , 3 ) );

for k = 1 : Number_Exits_One_Runway

  % finding the index of the exit
  Index_exit = find ( strcmp ( Exits_One_Runway { k }, temp_exits ) );

  Number_observations_one_exit = length ( Index_exit );

  Exit_specific_Data = temp_Runway_Data ( Index_exit );

  path_length_specific_exit = Exit_specific_Data (1).Path_length_ft;

  ratio_remaining_length_runway_length = Exit_specific_Data (1).Remaining_Distance_ft/( Exit_specific_Data
(1).Remaining_Distance_ft + Exit_specific_Data (1).Distance_from_threshold_ft);

  temp_Exit_Specific_Data = struct2table ( Exit_specific_Data );

  temp_total_change_heading = table2array ( temp_Exit_Specific_Data ( : , 9 ) );

  mean_total_change_heading = mean ( temp_total_change_heading );

  Exit_Speeds_one_exit_kts = table2array ( temp_Exit_Specific_Data ( : , 4 ) );

  mean_speed_kts = mean ( Exit_Speeds_one_exit_kts );

  standard_deviation_exit_speed_kts = std ( Exit_Speeds_one_exit_kts );

  total_Change_heading = table2array ( temp_Exit_Specific_Data ( : , 9 ) );


  Exit_Data ( m ).Airport = char ( directory ( i ). name (1:3) );

  Exit_Data ( m ).Runway_End = Unique_Runways { j };

  Exit_Data ( m ).Exit_Name = Exits_One_Runway { k };

  Exit_Data ( m ).path_length_ft = Exit_specific_Data (1).Path_length_ft ;

  Exit_Data ( m ).ratio_remaining_length_runway = ratio_remaining_length_runway_length;

  Exit_Data ( m ).mean_Exit_Speed_kts = mean_speed_kts;

  Exit_Data ( m ).STD_Exit_Speed_kts = standard_deviation_exit_speed_kts;

  normalized_exit_speed_kts = ( Exit_Speeds_one_exit_kts - mean_speed_kts ) / ( standard_deviation_exit_speed_kts );

  Exit_Data ( m ).Number_Of_Observations = length ( Index_exit );

  addpath ('/Users/atsl/Documents/ATSL_Git/ATSL_Git/REDIM/Runway_Exit_SpeedScripts/function');

  %Bimodal distribution check. %bimodality check for the exit angles.Takes zero if it is a
  %bimodal distribution.

  if length ( total_Change_heading ) > threshold_minimum_observations


    Exit_Data ( m ).Bimodality_Check_exit_angle = HartigansDipTest ( total_Change_heading );

  else
```

```
            Exit_Data ( m ).Bimodality_Check_exit_angle = 2;

        end
        %takes zero if the null hypothesis, data is normally
        %distributed is valid

        if length ( normalized_exit_speed_kts ) > threshold_minimum_observations

            Exit_Data ( m ).normality_Check_exit_speed_kts = kstest ( normalized_exit_speed_kts );

        else

            Exit_Data ( m ).normality_Check_exit_speed_kts = 2;

        end

        if (  Exit_Data ( m ).normality_Check_exit_speed_kts == 0 )

            exits_normal_exit_speed_distribution = [ exits_normal_exit_speed_distribution ,  Exit_specific_Data ];

        end

        if ( Exit_Data ( m ).normality_Check_exit_speed_kts == 0 & Exit_Data ( m ).Number_Of_Observations > 100 )

            normal_unique_exit_speed_index = [ normal_unique_exit_speed_index, m ];

        end


        if length ( total_Change_heading ) > threshold_minimum_observations

            Exit_Data ( m ).normality_Check_exit_angle = kstest ( total_Change_heading );

        else

            Exit_Data ( m ).normality_Check_exit_angle = 2;

        end


        Exit_Data ( m ).Mean_Total_Change_Azimuth = mean ( Exit_Speeds_one_exit_kts );

        Exit_Data ( m ).Exit_Angle = unique ( table2array ( temp_Exit_Specific_Data ( : , 10 ) ) );



        m = m + 1;

        disp ( m)

      end


   end

end

normal_unique_exit_speed_data = Exit_Data ( normal_unique_exit_speed_index );

% finding the standard deviation of different types of exits

% for high speed exits

normal_unique_exit_speed_table = struct2table ( normal_unique_exit_speed_data );

exit_angle_normal_unique_exit_speed_table = table2array ( normal_unique_exit_speed_table ( : , 13 ) );

exit_path_normal_unique_exit_speed_table = table2array ( normal_unique_exit_speed_table ( : , 4 ) );
```

```
Index_high_speed_exit = find ( exit_angle_normal_unique_exit_speed_table < 36 & exit_path_normal_unique_exit_speed_table >
1000 );

High_speed_unique_data = normal_unique_exit_speed_data ( Index_high_speed_exit );

High_speed_unique_table = struct2table ( High_speed_unique_data );

ratio_remaining_length_runway = table2array ( High_speed_unique_table ( : , 5 ) );

standard_deviation_trial_high_speed = table2array (High_speed_unique_table ( : , 7 ));

Index_remaining_length_runway_greater_than_05 = find ( ratio_remaining_length_runway > 0.5 );

if ( isempty ( Index_remaining_length_runway_greater_than_05 ) == 0 )

  standard_deviation_high_speed_greater_than_05 = max ( standard_deviation_trial_high_speed (
Index_remaining_length_runway_greater_than_05 ) );

end

Index_remaining_length_runway_between_05_035 = find ( ratio_remaining_length_runway > 0.35 & ratio_remaining_length_runway
< 0.5 );

if ( isempty ( Index_remaining_length_runway_between_05_035 ) == 0 )

  standard_deviation_high_speed_between_05_035 = max ( standard_deviation_trial_high_speed (
Index_remaining_length_runway_between_05_035 ) );

end

Index_remaining_length_runway_less_than_035 = find ( ratio_remaining_length_runway < 0.35 );

if ( isempty ( Index_remaining_length_runway_less_than_035 ) == 0 )

  standard_deviation_high_speed_less_than_035 = max ( standard_deviation_trial_high_speed (
Index_remaining_length_runway_less_than_035 ) );

end

if ( isempty (Index_remaining_length_runway_greater_than_05) == 1 )

  standard_deviation_high_speed_greater_than_05= standard_deviation_high_speed_between_05_035;

end


% for right angle exits

Index_right_angle_exit = find ( exit_angle_normal_unique_exit_speed_table < 95 & exit_angle_normal_unique_exit_speed_table >
85 );

Right_angle_unique_data = normal_unique_exit_speed_data ( Index_right_angle_exit );

Right_angle_unique_table = struct2table ( Right_angle_unique_data );

%removing C6-B (CLT ), G (DCA) and N1 (DCA) exits from the data

trial_right_angle_exit = table2array (Right_angle_unique_table ( : , 3 ));

trial_runway_end = table2array (Right_angle_unique_table ( : , 2 ));

Index_C6_B = find ( strcmp ( trial_right_angle_exit, 'C6-B' ) ==1  & strcmp ( trial_runway_end, '18L' ) ==1 );

Index_N1 = find ( strcmp ( trial_right_angle_exit, 'N1' ) ==1  & strcmp ( trial_runway_end, '01' ) ==1 );

Index_G = find ( strcmp ( trial_right_angle_exit, 'G' ) ==1  & strcmp ( trial_runway_end, '19' ) ==1 );

Index_JJ_12 = find ( strcmp ( trial_right_angle_exit, 'JJ' ) ==1  & strcmp ( trial_runway_end, '12' ) ==1 );
```

```
Index_JJ_30 = find ( strcmp ( trial_right_angle_exit, 'JJ' ) ==1  & strcmp ( trial_runway_end, '30' ) ==1 );

Index_T1 = find ( strcmp ( trial_right_angle_exit, 'T1' ) ==1  & strcmp ( trial_runway_end, '27' ) ==1 );

Index_right_remove = [ Index_C6_B; Index_N1; Index_G; Index_JJ_12; Index_JJ_30; Index_T1];

Right_angle_unique_data ( Index_right_remove ) =[];

Right_angle_unique_table = struct2table ( Right_angle_unique_data );

standard_deviation_exit_speeds_right_angle = table2array ( Right_angle_unique_table ( : , 7 ) );

standard_deviation_exit_speeds_right_angle ( standard_deviation_exit_speeds_right_angle > 7 ) = [];

standard_deviation_right_angle_exits = max ( standard_deviation_exit_speeds_right_angle );

% obtuse exits

index_obtuse_exits = find ( exit_angle_normal_unique_exit_speed_table > 100 );

Exit_data_table = struct2table ( Exit_Data );

Exit_data_angle = table2array ( Exit_data_table ( : , 13 ) );

normality_check_exit_data = table2array ( Exit_data_table ( : , 10 ) );

if ( isempty ( index_obtuse_exits ) == 1 )
   % implies there are no obtuse angle exits with normal exit speed
   % distribution and number of observations greater than 100.
   disp ('There are no obtuse angle exits with normal exit speed distribution and number of observations greater than 100')

   Index_obtuse = find ( Exit_data_angle > 95 & normality_check_exit_data  == 0 );

   Obtuse_exit_data = ( Exit_Data ( Index_obtuse ) );

   Obtuse_exit_table = struct2table ( Obtuse_exit_data );

   trial_obtuse_exit_angle = table2array ( Obtuse_exit_table ( : , 13 ) );

   %finding index of exits with angles greater than 130
   trial_1_obtuse_index = find ( trial_obtuse_exit_angle > 129 );

   trial_1_obtuse_observations = table2array ( Obtuse_exit_table (: , 8 ) );

   trial_1_obtuse_observations_count =  trial_1_obtuse_observations ( trial_1_obtuse_index );

   [ trial_1_obtuse_maximum, trial_1_obtuse_index_maximum ] = max ( trial_1_obtuse_observations_count ) ;

   Standard_deviation_obtuse = Obtuse_exit_data ( trial_1_obtuse_index ( trial_1_obtuse_index_maximum ) ).STD_Exit_Speed_kts;

end

if ( isempty ( index_obtuse_exits ) == 0 )

   disp ( ' There are few obtuse angle exits with normal exit speed distribution and number of observations greater than 100')


   Obtuse_exit_data = ( normal_unique_exit_speed_data ( index_obtuse_exits ) );

   Obtuse_exit_table = struct2table ( Obtuse_exit_data );

   trial_obtuse_exit_angle = table2array ( Obtuse_exit_table ( : , 13 ) );

   %finding index of exits with angles greater than 130
   trial_1_obtuse_index = find ( trial_obtuse_exit_angle > 129 );

   trial_1_obtuse_observations = table2array ( Obtuse_exit_table (: , 8 ) );

   trial_1_obtuse_observations_count =  trial_1_obtuse_observations ( trial_1_obtuse_index );
```

```
   [ trial_1_obtuse_maximum, trial_1_obtuse_index_maximum ] = max ( trial_1_obtuse_observations_count ) ;

   Standard_deviation_obtuse = Obtuse_exit_data ( trial_1_obtuse_index ( trial_1_obtuse_index_maximum ) ).STD_Exit_Speed_kts;

end

% wide throat exits

Index_wide_throat_exit = find ( exit_angle_normal_unique_exit_speed_table >= 45 & exit_path_normal_unique_exit_speed_table >
800 );

if isempty ( Index_wide_throat_exit ) == 0

disp ( ' There are few wide throat exits with number of observations greater than 100 and normal exit speed distribution');

Wide_throat_unique_data = normal_unique_exit_speed_data ( Index_wide_throat_exit );

Wide_throat_unique_table = struct2table ( Wide_throat_unique_data );

trial_wide_throat_std = table2array ( Wide_throat_unique_table ( :,7) );

Standard_deviation_wide_throat = max ( trial_wide_throat_std );

end

exit_data_path_length_ft = table2array ( Exit_data_table (:,4) );

if isempty ( Index_wide_throat_exit ) == 1

disp ( ' There are no wide throat exits with number of observations greater than 100 and normal exit speed distribution');


Index_wide_throat_exit = find ( Exit_data_angle > 50 & exit_data_path_length_ft  > 800 & normality_check_exit_data == 0 );

Wide_throat_unique_data = Exit_Data ( Index_wide_throat_exit );

Wide_throat_unique_table = struct2table ( Wide_throat_unique_data );

trial_wide_throat_std = table2array ( Wide_throat_unique_table ( :,7) );

trial_wide_throat_observations = table2array ( Wide_throat_unique_table ( :,8) );

[trial_maximum_wide_throat_observations, index_maximum_wide_throat_observations] = max ( trial_wide_throat_observations );

Standard_deviation_wide_throat = trial_wide_throat_std (index_maximum_wide_throat_observations);

end


% Non-standard exits

Index_non_standard_exit = find ( Exit_data_angle > 29 & Exit_data_angle < 86 & exit_data_path_length_ft  < 900 &
normality_check_exit_data == 0 );

Non_standard_unique_data = Exit_Data ( Index_non_standard_exit );

% Removing the wide-throat exits from the non-standard exits

Non_standard_unique_table = struct2table ( Non_standard_unique_data );

Non_standard_exit_angle = table2array ( Non_standard_unique_table ( :, 13 ) );

Non_standard_exit_path_length = table2array ( Non_standard_unique_table ( :, 4 ) );


% removing the wide throat exits from the non-standard exits
Index_non_standard_remove = find ( Non_standard_exit_path_length > 800 & Non_standard_exit_angle > 50 );

Non_standard_unique_data ( Index_non_standard_remove ) = [];
```

```
Non_standard_unique_table = struct2table ( Non_standard_unique_data );

trial_non_standard_observations = table2array ( Non_standard_unique_table ( :,8) );

trial_non_standard_std = table2array ( Non_standard_unique_table ( : , 7) );

[trial_maximum_non_standard_observations, index_non_standard_observations] = max ( trial_non_standard_observations );

Standard_deviation_non_standard_exit = trial_non_standard_std (index_non_standard_observations);


%end of STEP - 1


%% STEP - 2


%for loop to find the potential bimodal cases and low counts for specific exits
Total_exits = length ( Exit_Data );

for m = 1 : Total_exits


  % Recording potential bimodal cases
  if ( Exit_Data ( m ).Number_Of_Observations > 1000 & Exit_Data ( m ).Bimodality_Check_exit_angle > 0.03 )

    Flag_Bimodality ( n ).Airport =  Exit_Data ( m ).Airport;

    Flag_Bimodality ( n ).Runway_End = Exit_Data ( m ).Runway_End;

    Flag_Bimodality ( n ).Exit_Name =  Exit_Data ( m ).Exit_Name;

    n=n+1;

  end

  %% Recording exits with low counts
    if ( Exit_Data ( m ).Number_Of_Observations < threshold_minimum_observations )

        Low_Count ( l ).Airport = Exit_Data ( m ).Airport;

        Low_Count ( l ).Runway_End = Exit_Data ( m ).Runway_End;

        Low_Count ( l ).Exit_Name =  Exit_Data ( m ).Exit_Name;

        l = l + 1;

    else

       Low_Count = [];
    end

end
% end of STEP-2


%start of STEP-3
%% Cleaning the data, essentially cleaning data having exits with low number of observations
if ( isempty ( Low_Count ) == 0 )

for i = 1 : Number_Airports

  %%Removing low count exits
  Low_count_table = struct2table ( Low_Count );

  temp_airports = table2array( Low_count_table ( : , 1 ) );

  %check which airports have low exit counts
  Airport_index = find ( strcmp ( char ( directory ( i ).name (1:3) ) , temp_airports ) );
```

```
    Airport_specific_low_count = Low_Count ( Airport_index ) ;

    Raw_Data = importdata ( directory ( i ).name );

    Number_Observations = length ( Raw_Data );

    Number_exits_deleted = length ( Airport_specific_low_count );

    index_delete = [];

    for k = 1 : Number_exits_deleted

      for j = 1 : Number_Observations

          if ( ( strcmp ( Airport_specific_low_count ( k ).Runway_End , Raw_Data ( j ). Operational_Runway ) ==1 & ( strcmp (
Airport_specific_low_count ( k ).Exit_Name , Raw_Data ( j ). Exit_Name ) ==1 ) ) )

          index_delete = [ index_delete ; j ];

        end

      end

    end

    Raw_Data ( index_delete ) = [];

    Airport_being_saved = directory (i).name;

    filename_consolidated = [ '/Users/atsl/Documents/REDIM/ADG_I Consolidated' '/' Airport_being_saved];

    save ( filename_consolidated , 'Raw_Data' );

end

end

    %% end of cleaning data with low exit counts and saved them in the original directory

    %% for now this part of the code is not used as bimodal cases are not found very often. They are found only in ADG III.
    % Usually this part of the code is not used.
    %{

%%Adjusting the Bimodal cases

  i = 7;

  Runway_End = '27L';

  Exit_Name = 'A1-L';

  Index = [];

  Raw_Data = importdata ( directory ( i ).name);

  Mean = 33.27

  standard_deviation = 13.42;

  Range_lower = Mean - 2.5*standard_deviation;

  Range_higher = Mean + 2.5*standard_deviation;

   for j = 1 : Number_Observations

     if ( strcmp ( Raw_Data ( j ).Operational_Runway , '27L' ) ==1 & strcmp ( Raw_Data ( j ).Exit_Name , 'A1-L' ) == 1 )

        if ( Raw_Data ( j ).Total_change_heading > Range_higher | Raw_Data ( j ).Total_change_heading < Range_lower )

        Index = [ Index ; j ];
```

```
        end

      end

   end

Raw_Data ( Index ) = [];

save ( 'ORD.mat','Raw_Data');

%}

 %% Creating the consolidated data having got rid of exits with low records. The end
 % product is a stuct having all the records of a specific ADG
m = 1;

addpath('/Users/atsl/Documents/REDIM/ADG_I Consolidated');

directory =  dir ('/Users/atsl/Documents/REDIM/ADG_I Consolidated');

for i = 1 : Number_Airports

    if ( strcmp ( char ( directory ( i ).name(1) ) , '.' ) == 1 )

        Index_deleted_files = [ Index_deleted_files, i ];

    end

end

directory ( Index_deleted_files ) = [];

Number_Airports = length ( directory );

for i = 1 : Number_Airports


if ( i ==1 )

consolidated_table = struct2table ( importdata ( directory ( i ).name ) );

end

if ( i>1 )

temp_table = struct2table ( importdata ( directory ( i ).name ) );

consolidated_table = outerjoin ( consolidated_table, temp_table, 'MergeKeys', true);

end

disp ( i )

end

consolidated_struct = table2struct ( consolidated_table );

% checking if any of the standard deviations are empty. This usually
% happens with the ADG_I

check_high_speed_1 = isempty ( Index_remaining_length_runway_greater_than_05 ) + isempty ( Index_remaining_length_runway_less_than_035 );

if  ( check_high_speed_1 == 2 )

    standard_deviation_high_speed_greater_than_05 = standard_deviation_high_speed_between_05_035;

    standard_deviation_high_speed_less_than_035 = standard_deviation_high_speed_between_05_035;
```

```
end

check_wide_throat = isempty ( Index_wide_throat_exit );

if ( check_wide_throat == 1 )

   if ( Standard_deviation_non_standard_exit > 8 )

   Standard_deviation_wide_throat = Standard_deviation_non_standard_exit;

   else

   Standard_deviation_wide_throat = standard_deviation_high_speed_between_05_035;

   end

end


save ( '/Users/atsl/Documents/REDIM/ADG_I_Analysis/consolidated_data' , 'consolidated_struct' , '-v7.3');

save ( '/Users/atsl/Documents/REDIM/ADG_I_Analysis/normal_exit_speed_data' , 'exits_normal_exit_speed_distribution' );

save (
'/Users/atsl/Documents/REDIM/ADG_I_Analysis/Standard_deviation_for_high_speed_exits_with_remaining_length_ratio_below_0.35.mat', 'standard_deviation_high_speed_less_than_035');

save (
'/Users/atsl/Documents/REDIM/ADG_I_Analysis/Standard_deviation_for_high_speed_exits_with_remaining_length_ratio_between_0.35_and_0.5.mat', 'standard_deviation_high_speed_between_05_035' );

save (
'/Users/atsl/Documents/REDIM/ADG_I_Analysis/Standard_deviation_for_high_speed_exits_with_remaining_length_ratio_greater_than_0.5.mat', 'standard_deviation_high_speed_greater_than_05' );

save ( '/Users/atsl/Documents/REDIM/ADG_I_Analysis/Standard_deviation_for_right_angle_exits.mat', 'standard_deviation_right_angle_exits' );

save ( '/Users/atsl/Documents/REDIM/ADG_I_Analysis/Standard_deviation_for_obtuse_angle_exits.mat', 'Standard_deviation_obtuse' );

save ( '/Users/atsl/Documents/REDIM/ADG_I_Analysis/Standard_deviation_for_wide_throat_exits.mat', 'Standard_deviation_wide_throat' );

save ( '/Users/atsl/Documents/REDIM/ADG_I_Analysis/Standard_deviation_for_non_standard exits.mat', 'Standard_deviation_non_standard_exit' );

% this excel sheet is copied to jmp and further analysis is done on this
% excel sheet. The linear model developed from jmp is used in the next code

Excel_file_directory = '/Users/atsl/Documents/REDIM/ADG_II_Analysis/';

filename_excel = [ Excel_file_directory 'consolidated_data.csv'];

writetable (consolidated_table, filename_excel);
```

Model_provides_mean_exit_speed.m

```
 clc
clear

addpath ('/Users/atsl/Documents/REDIM/ADG_VI Consolidated');

directory = dir ( '/Users/atsl/Documents/REDIM/ADG_VI Consolidated');

addpath ('/Users/atsl/Documents/ATSL_Git/ATSL_Git/REDIM/Runway_Exit_SpeedScripts/function');
```

```
[directory, Number_Airports] = Removes_unnecessary_files ( directory );

counter = 1;

for b = 1 : Number_Airports

  load (directory(b).name);

  Airport_table = struct2table ( Data_ADG_6 );

operational_runway = table2array ( Airport_table ( : , 1 ) ) ;

exit_name = table2array ( Airport_table ( : , 3 ) );

Exit_angle_records = table2array ( Airport_table( : , 10 ) );

 % Removing exit data having N1 exit on runway 01 at DCA
index_N = find ( strcmp ( operational_runway, '01' ) == 1 & strcmp ( exit_name, 'N1' ) ==1);

index_G = find ( strcmp ( operational_runway, '19' ) == 1 & strcmp ( exit_name, 'G' ) ==1);

% Removing exit data having C6-B exit on runway 18L at CLT
index_C6_B = find ( strcmp ( operational_runway, '18L' ) == 1 & strcmp ( exit_name, 'C6-B' ) ==1);

Index_JJ_12 = find ( strcmp ( exit_name, 'JJ' ) ==1  & strcmp ( operational_runway, '12' ) ==1 );

Index_JJ_30 = find ( strcmp ( exit_name, 'JJ' ) ==1  & strcmp ( operational_runway, '30' ) ==1 );

Index_T1 = find ( strcmp ( exit_name, 'T1' ) ==1  & strcmp ( operational_runway, '27' ) ==1 );

Index_DB_R = find ( strcmp ( exit_name, 'DB-R' ) ==1  & strcmp ( operational_runway, '13L' ) ==1 );

Index_D_R = find ( strcmp ( exit_name, 'D-R' ) ==1  & strcmp ( operational_runway, '13L' ) ==1);

Index_B6 = find ( strcmp ( exit_name, 'B6' ) ==1  & strcmp ( operational_runway, '10L' ) ==1);

index_right_angle = find ( Exit_angle_records >=85 & Exit_angle_records <= 95 );

number_right_angle_exits = length ( index_right_angle );

index_delete_records = [];

% Removing the data with exit speeds greater than 50 knots for right angle
% exits.
for i = 1 : number_right_angle_exits

   if ( Data_ADG_6 ( index_right_angle ( i ) ).Exit_Speed_kts > 50 )

      trial_index = index_right_angle ( i );

      index_delete_records = [ index_delete_records ; trial_index ];

   end

end

index_delete_records = [ index_delete_records; index_N; index_C6_B; Index_JJ_12; Index_JJ_30; Index_T1; Index_DB_R;
Index_D_R];

Data_ADG_6 ( index_delete_records ) = [];

Airport_table = struct2table( Data_ADG_6 );

Observed_exit_speed_kts = table2array ( Airport_table (:,4) );

Path_length_ft = table2array ( Airport_table (:,11) );
```

```
Radius_ft = table2array ( Airport_table (:,16) );
Remaining_Distance_ft = table2array ( Airport_table (:,17) );
Distance_from_threshold_ft = table2array ( Airport_table (:, 15) );
operational_runway = table2array ( Airport_table ( : , 1 ) ) ;
exit_name = table2array ( Airport_table ( : , 3 ) );
Exit_angle_records = table2array ( Airport_table( : , 10 ) );
Runway_length_ft = Remaining_Distance_ft + Distance_from_threshold_ft;
Remaining_length_ratio = Remaining_Distance_ft./Runway_length_ft;
Holdbar_distance_ft = table2array (Airport_table (:,12));
Closest_Taxiway_Distance_ft = table2array(Airport_table (:,13));
Taxiway_Width_ft = table2array(Airport_table (:,14));
Exit_Speeds_Kts = table2array (Airport_table (:,4));
Airports = table2array(Airport_table (:,19));
Unique_Airports = unique (Airports);
Number_Airports = length (Unique_Airports);

for i = 1 : Number_Airports
  Index_Airport = find ( strcmp (Unique_Airports{i}, Airports));
  Runways_Airport = operational_runway (Index_Airport);
  Unique_Runways_Airport = unique (Runways_Airport);
  Number_Runways_Airport = length (Unique_Runways_Airport);
  for j = 1 : Number_Runways_Airport
    Index_Runway = find ( strcmp (Runways_Airport, Unique_Runways_Airport {j}));
    Index_Runway_Actual = Index_Airport(Index_Runway);
    Exit_Name_Runway = exit_name (Index_Runway_Actual);
    Unique_Exits_Runway = unique (Exit_Name_Runway);
    Number_Exits_Runway = length (Unique_Exits_Runway);
    for l = 1 : Number_Exits_Runway
    Index_Exit = find (strcmp(Unique_Exits_Runway{l}, Exit_Name_Runway));
    Index_Exit_Actual =  Index_Runway_Actual ( Index_Exit );
    Exit_Observed_Speeds_mean = mean (Observed_exit_speed_kts (Index_Exit_Actual));
    Path_length_exit_ft = unique(Path_length_ft(Index_Exit_Actual));
    Radius_exit_ft = unique(Radius_ft(Index_Exit_Actual));
    Holdbar_distance_exit_ft = unique(Holdbar_distance_ft(Index_Exit_Actual));
    Remaining_distance_exit_ft = unique(Remaining_Distance_ft(Index_Exit_Actual));
```

```
        Runway_length_exit_ft = unique(Runway_length_ft(Index_Exit_Actual));
        Closest_Taxiway_Distance_exit_ft = unique(Closest_Taxiway_Distance_ft(Index_Exit_Actual));
        Taxiway_Width_exit_ft = unique(Taxiway_Width_ft(Index_Exit_Actual));
        Exit_Angle = unique(Exit_angle_records(Index_Exit_Actual));
        Exit_Data(counter).Index = Index_Exit_Actual;
        Exit_Data(counter).Airport = Unique_Airports{i};
        Exit_Data(counter).Runway_End = Unique_Runways_Airport {j};
        Exit_Data(counter).Exit_Name = Unique_Exits_Runway{l};
        Exit_Data(counter).Number_Observations = length(Index_Exit_Actual);
        Exit_Data(counter).Observed_Speeds_kts = Exit_Observed_Speeds_mean;
        Exit_Data(counter).Exit_Angle = Exit_Angle;
        Exit_Data(counter).Path_length_exit_ft = Path_length_exit_ft;
        Exit_Data(counter).Radius_exit_ft = Radius_exit_ft;
        Exit_Data(counter).Holdbar_distance_exit_ft = Holdbar_distance_exit_ft;
        Exit_Data(counter).Remaining_distance_exit_ft = Remaining_distance_exit_ft;
        Exit_Data(counter).Runway_length_exit_ft = Runway_length_exit_ft;
        Exit_Data(counter).Closest_Taxiway_Distance_ft = Closest_Taxiway_Distance_exit_ft ;
        Exit_Data(counter).Taxiway_Width_ft = Taxiway_Width_exit_ft;
        Exit_Observed_Speeds_mean = [];
        Index_Exit_Actual = [];
        Index_Exit = [];
        counter = counter + 1;
        end
        Index_Runway_Actual = [];
        Index_Runway = [];
    end
    Index_Airport = [];
end
Exit_Table = struct2table (Exit_Data);
Exit_Table_Observed_Speeds_kts = table2array (Exit_Table (:,6));
Index_High_Speeds = find (Exit_Table_Observed_Speeds_kts > 60);
Exit_Table_Number_Observations = table2array ( Exit_Table (:,5) );
Index_less_observations = find ( Exit_Table_Number_Observations < 10);
Index_delete = [Index_High_Speeds; Index_less_observations];
Exit_Removed_Data = Exit_Data;
```

```
Exit_Removed_Data(Index_delete) = [];

Exit_Removed_Table = struct2table ( Exit_Removed_Data );

Exit_Removed_Table(:,1) = [];

end

Excel_file_directory = '/Users/atsl/Documents/REDIM/ADG_VI_Analysis/';

filename_excel = [ Excel_file_directory 'Exit Data.csv'];

writetable (Exit_Removed_Table, filename_excel);
```

Validation for ADG_I model. Scripts for other ADG's are similar. These scripts are on the Git Hub.

```
%code to validate the ADG_I model

% Use the Jmp formula obtained for validation.
%Choose an airport. Make sure that the airport is already analysed before
%using it over here. Every airport that goes through first two codes can be
%validated using this code.

clc
clear

threshold_standard_deviation = 2;

addpath ('/Users/atsl/Documents/REDIM/Validation/ADG_I Consolidated');
directory = dir('/Users/atsl/Documents/REDIM/Validation/ADG_I Consolidated');

Index_of_deleted_files=[]; % index of deleted files

%finding the number of entries for a month
Number_of_entries_directory = length( directory );

%Removing entries with names '.' , '._' and '..'%

random = 1;

for i=1 : Number_of_entries_directory

  if( strcmp( char( directory( i ).name( 1 ) ),'.' ) == 1 )

    Index_of_deleted_files=[ Index_of_deleted_files,i ];

  end

end

directory( Index_of_deleted_files )=[];

%updating the number of entries in the directory

Number_of_entries_directory = length( directory );

for i = 1 : Number_of_entries_directory

Airports{i} = char ( directory ( i ).name(1:3) );

end

Airport_observed_predicted_speed_check = 'JFK';

Index_match_Airport = find ( strcmp ( Airports, Airport_observed_predicted_speed_check) );
```

```
addpath ('/Users/atsl/Documents/REDIM/ADG_I_Analysis');

load('Standard_deviation_for_high_speed_exits_with_remaining_length_ratio_below_0.35.mat')
load('Standard_deviation_for_high_speed_exits_with_remaining_length_ratio_between_0.35_and_0.5.mat')
load('Standard_deviation_for_high_speed_exits_with_remaining_length_ratio_greater_than_0.5.mat')
load('Standard_deviation_for_non_standard exits.mat')
load('Standard_deviation_for_obtuse_angle_exits.mat')
load('Standard_deviation_for_right_angle_exits.mat')
load('Standard_deviation_for_wide_throat_exits.mat')


  load ( directory (Index_match_Airport) .name);

  Data_ADG_I_table = struct2table (Data_ADG_1);

  Path_length_ft = table2array ( Data_ADG_I_table ( : , 11 ) );

  Radius_ft = table2array ( Data_ADG_I_table ( : , 16 ) );

  Remaining_Distance_ft = table2array ( Data_ADG_I_table (:,17) );

Distance_from_threshold_ft = table2array ( Data_ADG_I_table (:, 15) );

Observed_Exit_Speeds_kts = table2array ( Data_ADG_I_table (:, 4) );

Runway_length_ft = Remaining_Distance_ft + Distance_from_threshold_ft;

Remaining_length_ratio = Remaining_Distance_ft./Runway_length_ft;

Holdbar_distance_ft = table2array(Data_ADG_I_table(:,12));

predicted_exit_speed_mean_kts = 28.7 + 0.0259*Path_length_ft - 1.262*(Path_length_ft./Radius_ft) - 0.0500*Holdbar_distance_ft;

Exit_angle_degrees = table2array ( Data_ADG_I_table (:, 10 ) );

High_speed_sigma_remaining_length_ratio_less_than_035 = standard_deviation_high_speed_less_than_035;

High_speed_sigma_remaining_length_ratio_between_05_035 = standard_deviation_high_speed_between_05_035;

High_speed_sigma_remaining_length_ratio_greater_than_05 = standard_deviation_high_speed_greater_than_05;

Right_Angle_speed_sigma = standard_deviation_right_angle_exits;

non_standard_sigma = Standard_deviation_non_standard_exit;

Obtuse_Angle_speed_sigma = Standard_deviation_obtuse;

if isempty (Standard_deviation_wide_throat ) ==0

Wide_throat_exit_sigma = Standard_deviation_wide_throat;

end

if isempty (Standard_deviation_wide_throat ) ==1

Wide_throat_exit_sigma = Standard_deviation_non_standard_exit;

end

Index_30 = find ( Exit_angle_degrees <36 & Path_length_ft > 1000 );

Index_Wide_throat = find ( Exit_angle_degrees > 50 & Path_length_ft > 800 );

Number_wide_throat_exits = length ( Index_Wide_throat );
% changing the predicted speed formula for the wide throat exits


Number_high_speed_exits = length ( Index_30 );
```

```
Total_observations = length ( Data_ADG_1 );

random_numbers = randn ( 2*Total_observations, 1 );

random_numbers ( find ( abs ( random_numbers ) > threshold_standard_deviation  ) ) = [];


%estimating the speeds for high speed exits

for i = 1 : Number_high_speed_exits

   if ( Remaining_length_ratio ( Index_30 ( i ) ) < 0.35 )

   Predicted_exit_speed_kts ( Index_30 ( i ),1 ) = predicted_exit_speed_mean_kts ( Index_30 ( i ) ) + random_numbers ( Index_30 ( i ) )*High_speed_sigma_remaining_length_ratio_less_than_035;

   while Predicted_exit_speed_kts ( Index_30 ( i ) ) < 0

   Predicted_exit_speed_kts ( Index_30 ( i ),1 ) =  predicted_exit_speed_mean_kts ( Index_30 ( i ) ) + randn ( 1,1 )*High_speed_sigma_remaining_length_ratio_less_than_035;

   end

   end


    if ( Remaining_length_ratio ( Index_30 ( i ) ) > 0.5 )

   Predicted_exit_speed_kts ( Index_30 ( i ),1 ) = predicted_exit_speed_mean_kts ( Index_30 ( i ) ) + random_numbers ( Index_30 ( i ) )*High_speed_sigma_remaining_length_ratio_greater_than_05;

   while Predicted_exit_speed_kts ( Index_30 ( i ) ) < 0

   Predicted_exit_speed_kts ( Index_30 ( i ),1 ) =  predicted_exit_speed_mean_kts ( Index_30 ( i ) ) + randn ( 1,1 )*High_speed_sigma_remaining_length_ratio_greater_than_05;

   end

   end


    if ( Remaining_length_ratio ( Index_30 ( i ) ) <= 0.5  & Remaining_length_ratio (Index_30 ( i ) ) >= 0.35 )

   Predicted_exit_speed_kts ( Index_30 ( i ),1 ) = predicted_exit_speed_mean_kts ( Index_30 ( i ) ) + random_numbers ( Index_30 ( i ) )*High_speed_sigma_remaining_length_ratio_between_05_035;

   while Predicted_exit_speed_kts ( Index_30 ( i ) ) < 0

   Predicted_exit_speed_kts ( Index_30 ( i ),1 ) =  predicted_exit_speed_mean_kts ( Index_30 ( i ) ) + randn ( 1,1 )*High_speed_sigma_remaining_length_ratio_between_05_035;

   end

   end

end

%estimating the speed for right angle exits

Index_90 = find ( Exit_angle_degrees < 95 & Exit_angle_degrees > 85 );

Number_right_angle_exits = length ( Index_90 );

for i = 1 : Number_right_angle_exits

   Predicted_exit_speed_kts ( Index_90 ( i ),1 ) = predicted_exit_speed_mean_kts ( Index_90 ( i ) ) + random_numbers ( Index_90 ( i ) )*standard_deviation_right_angle_exits;

   while Predicted_exit_speed_kts ( Index_90 ( i ) ) < 0
```

```
   Predicted_exit_speed_kts ( Index_90 ( i ),1 ) =  predicted_exit_speed_mean_kts ( Index_90 ( i ) ) + randn ( 1,1 )*standard_deviation_right_angle_exits;

   end

end

%estimating the speed for obtuse exits

Index_95_150 = find ( Exit_angle_degrees > 100 );

Number_obtuse_exits = length ( Index_95_150 );

for i = 1 : Number_obtuse_exits

   Predicted_exit_speed_kts ( Index_95_150( i ),1 ) = predicted_exit_speed_mean_kts ( Index_95_150( i ) ) + random_numbers ( Index_95_150( i ) )*Standard_deviation_obtuse;

   while Predicted_exit_speed_kts ( Index_95_150( i ) ) < 0

   Predicted_exit_speed_kts ( Index_95_150( i ),1 ) =  predicted_exit_speed_mean_kts ( Index_95_150( i ) ) + randn ( 1,1 )*Standard_deviation_obtuse;

   end

end

%estimating the speed of wide throat exits

for i = 1 : Number_wide_throat_exits

   Predicted_exit_speed_kts ( Index_Wide_throat( i ),1 ) = predicted_exit_speed_mean_kts ( Index_Wide_throat( i ) ) + random_numbers ( Index_Wide_throat( i ) )*Wide_throat_exit_sigma;

   while Predicted_exit_speed_kts ( Index_Wide_throat( i ) ) < 0

   Predicted_exit_speed_kts ( Index_Wide_throat( i ),1 ) =  predicted_exit_speed_mean_kts ( Index_Wide_throat( i ) ) + randn ( 1,1 )*Wide_throat_exit_sigma;

   end

end

Index = [ Index_30 ; Index_90 ; Index_95_150 ; Index_Wide_throat ];

All_Index = transpose ( 1 : 1 : Total_observations );

% non-standard exits

Index_non_standard = find ( ismember ( All_Index, Index ) == 0 );

Number_non_standard_exits = length ( Index_non_standard );

for i = 1 : Number_non_standard_exits

   Predicted_exit_speed_kts ( Index_non_standard( i ),1 ) = predicted_exit_speed_mean_kts ( Index_non_standard( i ) ) + random_numbers ( Index_non_standard( i ) )*Standard_deviation_non_standard_exit;

   while Predicted_exit_speed_kts ( Index_non_standard( i ) ) < 0

   Predicted_exit_speed_kts ( Index_non_standard( i ),1 ) =  predicted_exit_speed_mean_kts ( Index_non_standard( i ) ) + randn ( 1,1 )*Standard_deviation_non_standard_exit;

   end

end

%creating a folder to save the ADG I validation plots
```

```
mkdir ( '/Users/atsl/Documents/REDIM/Validation/ADG_I Analysis' );

figure('units','normalized','outerposition',[0 0 1 1]);

cdfplot ( Observed_Exit_Speeds_kts );

hold on;

cdfplot ( Predicted_exit_speed_kts );

xlabel (' Exit Speed at the Point of Curvature (knots) ');

ylabel (' Cumulative Density Function ');

set ( gca, 'FontSize', 24 );

leg = legend ( [' Observed Exit Speed (knots) - ' Airport_observed_predicted_speed_check] , [ ' Predicted Exit Speed (knots) - ' Airport_observed_predicted_speed_check ] , 'Location' , 'NorthWest');

leg.FontSize = 32;

legend boxoff;

title ('');

set(findall(gca, 'Type', 'Line'),'LineWidth',3);

path = '/Users/atsl/Documents/REDIM/Validation/ADG_IV Analysis';

filename = [ 'All Exits -' Airport_observed_predicted_speed_check];

saveas ( gca, fullfile ( path, filename ), 'fig' );

% plotting the high speed exits

High_speed_predicted_kts = Predicted_exit_speed_kts (Index_30);

High_speed_observed_kts = Observed_Exit_Speeds_kts (Index_30);

if ( length (High_speed_predicted_kts> 0) )

figure('units','normalized','outerposition',[0 0 1 1]);

cdfplot ( High_speed_observed_kts );

hold on;

cdfplot ( High_speed_predicted_kts );

xlabel (' Exit Speed at the point of curvature (knots) ');

ylabel (' Cumulative Density Function ');

set ( gca, 'FontSize', 24 );

leg = legend ( [' Observed Exit Speed for High Speed Exits (knots) - ' Airport_observed_predicted_speed_check] , [ ' Predicted Exit Speed for High Speed Exits (knots) - ' Airport_observed_predicted_speed_check ] , 'Location' , 'NorthWest');

leg.FontSize = 32;

legend boxoff;

title ('');

set(findall(gca, 'Type', 'Line'),'LineWidth',3);

path = '/Users/atsl/Documents/REDIM/Validation/ADG_I Analysis';

filename = [ 'High Speed Exits -' Airport_observed_predicted_speed_check];
```

```
saveas ( gca, fullfile ( path, filename ), 'fig' );

end

% removing right angle exits with exit speeds greater than 50 knots

Right_angle_predicted_kts = Predicted_exit_speed_kts (Index_90);

Right_angle_observed_kts = Observed_Exit_Speeds_kts (Index_90);

Index_Right_angle_greater_50 = find ( Right_angle_observed_kts > 50 );

Right_angle_predicted_kts ( Index_Right_angle_greater_50 ) = [];

Right_angle_observed_kts ( Index_Right_angle_greater_50 ) = [];


%plotting right angle exits
if (length (Right_angle_predicted_kts)>0)

figure('units','normalized','outerposition',[0 0 1 1]);


cdfplot ( Right_angle_observed_kts );

hold on;

cdfplot ( Right_angle_predicted_kts );

xlabel (' Exit Speed at the point of curvature (knots) ');

ylabel (' Cumulative Density Function ');

set ( gca, 'FontSize', 24 );

leg = legend ( [' Observed Exit Speed for Right angle Exits (knots) - ' Airport_observed_predicted_speed_check] , [ ' Predicted Exit
Speed for Right angle Exits (knots) - ' Airport_observed_predicted_speed_check ] , 'Location' , 'NorthWest');

leg.FontSize = 32;

legend boxoff;

title ('');

set(findall(gca, 'Type', 'Line'),'LineWidth',3);

path = '/Users/atsl/Documents/REDIM/Validation/ADG_I Analysis';

filename = [ 'Right angle Exits -' Airport_observed_predicted_speed_check];

saveas ( gca, fullfile ( path, filename ), 'fig' );

end


% plotting the Wide_throat exits

if (length (Index_Wide_throat) > 0 )

Wide_throat_predicted_kts = Predicted_exit_speed_kts (Index_Wide_throat);

Wide_throat_observed_kts = Observed_Exit_Speeds_kts (Index_Wide_throat);


figure('units','normalized','outerposition',[0 0 1 1]);

cdfplot ( Wide_throat_observed_kts );

hold on;
```

```
cdfplot ( Wide_throat_predicted_kts );

xlabel (' Exit Speed at the point of curvature (knots) ');

ylabel (' Cumulative Density Function ');

set ( gca, 'FontSize', 24 );

leg = legend ( [' Observed Exit Speed for Wide throat Exits (knots) - ' Airport_observed_predicted_speed_check] , [ ' Predicted Exit Speed for Wide throat Exits (knots) - ' Airport_observed_predicted_speed_check ] , 'Location' , 'NorthWest');

leg.FontSize = 32;

legend boxoff;

title ('');

set(findall(gca, 'Type', 'Line'),'LineWidth',3);

path = '/Users/atsl/Documents/REDIM/Validation/ADG_I Analysis';

filename = [ 'Wide_throat Exits -' Airport_observed_predicted_speed_check];

saveas ( gca, fullfile ( path, filename ), 'fig' );

end

% plotting the non_standard exits

non_standard_predicted_kts = Predicted_exit_speed_kts (Index_non_standard);

non_standard_observed_kts = Observed_Exit_Speeds_kts (Index_non_standard);

if ( length (non_standard_predicted_kts) > 0 )
figure('units','normalized','outerposition',[0 0 1 1]);

cdfplot ( non_standard_observed_kts );

hold on;

cdfplot ( non_standard_predicted_kts );

xlabel (' Exit Speed at the point of curvature (knots) ');

ylabel (' Cumulative Density Function ');

set ( gca, 'FontSize', 24 );

leg = legend ( [' Observed Exit Speed for non standard Exits (knots) - ' Airport_observed_predicted_speed_check] , [ ' Predicted Exit Speed for non standard Exits (knots) - ' Airport_observed_predicted_speed_check ] , 'Location' , 'NorthWest');

leg.FontSize = 32;

legend boxoff;

title ('');

set(findall(gca, 'Type', 'Line'),'LineWidth',3);

path = '/Users/atsl/Documents/REDIM/Validation/ADG_I Analysis';

filename = [ 'non_standard Exits -' Airport_observed_predicted_speed_check];

saveas ( gca, fullfile ( path, filename ), 'fig' );

end


%plotting obtuse angle exit speeds
```

```
obtuse_predicted_kts = Predicted_exit_speed_kts (Index_95_150);

obtuse_observed_kts = Observed_Exit_Speeds_kts (Index_95_150);

Index_Obtuse_angle_greater_50 = find ( obtuse_observed_kts > 50 );

obtuse_predicted_kts ( Index_Obtuse_angle_greater_50 ) = [];

obtuse_observed_kts ( Index_Obtuse_angle_greater_50 ) = [];

if ( length ( obtuse_predicted_kts ) > 0 )

figure('units','normalized','outerposition',[0 0 1 1]);

cdfplot ( obtuse_observed_kts );

hold on;

cdfplot ( obtuse_predicted_kts );

xlabel (' Exit Speed at the point of curvature (knots) ');

ylabel (' Cumulative Density Function ');

set ( gca, 'FontSize', 24 );

leg = legend ( [' Observed Exit Speed for Obtuse Exits (knots) - ' Airport_observed_predicted_speed_check] , [ ' Predicted Exit Speed for Obtuse Exits (knots) - ' Airport_observed_predicted_speed_check ] , 'Location' , 'NorthWest');

leg.FontSize = 32;

legend boxoff;

title ('');

set(findall(gca, 'Type', 'Line'),'LineWidth',3);

path = '/Users/atsl/Documents/REDIM/Validation/ADG_I Analysis';

filename = [ 'obtuse Exits -' Airport_observed_predicted_speed_check];

saveas ( gca, fullfile ( path, filename ), 'fig' );

end
```

Validation of ADG 1 with all the airports analyzed. This code also tries other methods. Scripts for other ADG's are on Git Hub.

ADG_I_final_parsing.m

```
 % code to fit the model and predict the exit speed.

%This code uses linear equation, quadratic equation and tries neural model
%and plots them in one CDF. This model does this for all the airports
%analyzed using the consolidated struct file obtained from the third code.
clc
clear

addpath ('/Users/atsl/Documents/REDIM/ADG_I_Analysis');

load ('consolidated_data.mat' );
load('Standard_deviation_for_high_speed_exits_with_remaining_length_ratio_below_0.35.mat')
load('Standard_deviation_for_high_speed_exits_with_remaining_length_ratio_between_0.35_and_0.5.mat')
load('Standard_deviation_for_high_speed_exits_with_remaining_length_ratio_greater_than_0.5.mat')
load('Standard_deviation_for_non_standard exits.mat')
load('Standard_deviation_for_obtuse_angle_exits.mat')
load('Standard_deviation_for_right_angle_exits.mat')
load('Standard_deviation_for_wide_throat_exits.mat')
```

```
threshold_standard_deviation = 2;

consolidated_table = struct2table ( consolidated_struct );

operational_runway = table2array ( consolidated_table ( : , 1 ) ) ;

exit_name = table2array ( consolidated_table ( : , 3 ) );

Exit_angle_records = table2array ( consolidated_table( : , 10 ) );

 % Removing exit data having N1 exit on runway 01 at DCA
index_N = find ( strcmp ( operational_runway, '01' ) == 1 & strcmp ( exit_name, 'N1' ) ==1);

% Removing exit data having C6-B exit on runway 18L at CLT
index_C6_B = find ( strcmp ( operational_runway, '18L' ) == 1 & strcmp ( exit_name, 'C6-B' ) ==1);

index_right_angle = find ( Exit_angle_records >=85 & Exit_angle_records <= 90 );

number_right_angle_exits = length ( index_right_angle );

index_delete_records = [];

% Removing the data with exit speeds greater than 50 knots for right angle
% exits.
for i = 1 : number_right_angle_exits

   if ( consolidated_struct ( index_right_angle ( i ) ).Exit_Speed_kts > 50 )

      trial_index = index_right_angle ( i );

      index_delete_records = [ index_delete_records ; trial_index ];

   end

end

index_delete_records = [ index_delete_records; index_N; index_C6_B ];

consolidated_struct ( index_delete_records ) = [];

consolidated_table = struct2table( consolidated_struct );

Observed_exit_speed_kts = table2array ( consolidated_table (:,4) );

Path_length_ft = table2array ( consolidated_table (:,11) );

Radius_ft = table2array ( consolidated_table (:,16) );

Remaining_Distance_ft = table2array ( consolidated_table (:,17) );

Distance_from_threshold_ft = table2array ( consolidated_table (:, 15) );

Runway_length_ft = Remaining_Distance_ft + Distance_from_threshold_ft;

Remaining_length_ratio = Remaining_Distance_ft./Runway_length_ft;


%predicted mean exit speed equation obtained from jmp
predicted_exit_speed_mean_kts = 20.129582 + 0.0197865*Path_length_ft - 2.213349*(Path_length_ft./Radius_ft);

Exit_angle_degrees = table2array ( consolidated_table (:, 10 ) );

High_speed_sigma_remaining_length_ratio_less_than_035 = standard_deviation_high_speed_less_than_035;

High_speed_sigma_remaining_length_ratio_between_05_035 = standard_deviation_high_speed_between_05_035;

High_speed_sigma_remaining_length_ratio_greater_than_05 = standard_deviation_high_speed_greater_than_05;

Right_Angle_speed_sigma = standard_deviation_right_angle_exits;
```

```
non_standard_sigma = Standard_deviation_non_standard_exit;

Obtuse_Angle_speed_sigma = Standard_deviation_obtuse;

if isempty (Standard_deviation_wide_throat ) ==0

Wide_throat_exit_sigma = Standard_deviation_wide_throat;

end

if isempty (Standard_deviation_wide_throat ) ==1

Wide_throat_exit_sigma = Standard_deviation_non_standard_exit;

end

Index_30 = find ( Exit_angle_degrees <36 & Path_length_ft > 1000 );

Index_Wide_throat = find ( Exit_angle_degrees > 50 & Path_length_ft > 800 );

Number_wide_throat_exits = length ( Index_Wide_throat );
% changing the predicted speed formula for the wide throat exits


Number_high_speed_exits = length ( Index_30 );

Total_observations = length ( consolidated_struct );

random_numbers = randn ( 2*Total_observations, 1 );

random_numbers ( find ( abs ( random_numbers ) > threshold_standard_deviation  ) ) = [];



%estimating the speeds for high speed exits

for i = 1 : Number_high_speed_exits

   if ( Remaining_length_ratio ( Index_30 ( i ) ) < 0.35 )

   Predicted_exit_speed_kts ( Index_30 ( i ),1 ) = predicted_exit_speed_mean_kts ( Index_30 ( i ) ) + random_numbers ( Index_30 ( i ) )*High_speed_sigma_remaining_length_ratio_less_than_035;

   while Predicted_exit_speed_kts ( Index_30 ( i ) ) < 0

   Predicted_exit_speed_kts ( Index_30 ( i ),1 ) = predicted_exit_speed_mean_kts ( Index_30 ( i ) ) + randn ( 1,1 )*High_speed_sigma_remaining_length_ratio_less_than_035;

   end

   end


    if ( Remaining_length_ratio ( Index_30 ( i ) ) > 0.5 )

   Predicted_exit_speed_kts ( Index_30 ( i ),1 ) = predicted_exit_speed_mean_kts ( Index_30 ( i ) ) + random_numbers ( Index_30 ( i ) )*High_speed_sigma_remaining_length_ratio_greater_than_05;

   while Predicted_exit_speed_kts ( Index_30 ( i ) ) < 0

   Predicted_exit_speed_kts ( Index_30 ( i ),1 ) = predicted_exit_speed_mean_kts ( Index_30 ( i ) ) + randn ( 1,1 )*High_speed_sigma_remaining_length_ratio_greater_than_05;

   end

   end


    if ( Remaining_length_ratio ( Index_30 ( i ) ) <= 0.5  & Remaining_length_ratio (Index_30 ( i ) ) >= 0.35 )
```

```
   Predicted_exit_speed_kts ( Index_30 ( i ),1 ) = predicted_exit_speed_mean_kts ( Index_30 ( i ) ) + random_numbers ( Index_30 (
i ) )*High_speed_sigma_remaining_length_ratio_between_05_035;

   while Predicted_exit_speed_kts ( Index_30 ( i ) ) < 0

   Predicted_exit_speed_kts ( Index_30 ( i ),1 ) =  predicted_exit_speed_mean_kts ( Index_30 ( i ) ) + randn ( 1,1
)*High_speed_sigma_remaining_length_ratio_between_05_035;

   end

   end

end

%estimating the speed for right angle exits

Index_90 = find ( Exit_angle_degrees < 95 & Exit_angle_degrees > 85 );

Number_right_angle_exits = length ( Index_90 );

for i = 1 : Number_right_angle_exits

   Predicted_exit_speed_kts ( Index_90 ( i ),1 ) = predicted_exit_speed_mean_kts ( Index_90 ( i ) ) + random_numbers ( Index_90 (
i ) )*standard_deviation_right_angle_exits;

   while Predicted_exit_speed_kts ( Index_90 ( i ) ) < 0

   Predicted_exit_speed_kts ( Index_90 ( i ),1 ) =  predicted_exit_speed_mean_kts ( Index_90 ( i ) ) + randn ( 1,1
)*standard_deviation_right_angle_exits;

   end

end

%estimating the speed for obtuse exits

Index_95_150 = find ( Exit_angle_degrees > 100 );

Number_obtuse_exits = length ( Index_95_150 );

for i = 1 : Number_obtuse_exits

   Predicted_exit_speed_kts ( Index_95_150( i ),1 ) = predicted_exit_speed_mean_kts ( Index_95_150( i ) ) + random_numbers (
Index_95_150( i ) )*Standard_deviation_obtuse;

   while Predicted_exit_speed_kts ( Index_95_150( i ) ) < 0

   Predicted_exit_speed_kts ( Index_95_150( i ),1 ) =  predicted_exit_speed_mean_kts ( Index_95_150( i ) ) + randn ( 1,1
)*Standard_deviation_obtuse;

   end

end

%estimating the speed of wide throat exits

for i = 1 : Number_wide_throat_exits

   Predicted_exit_speed_kts ( Index_Wide_throat( i ),1 ) = predicted_exit_speed_mean_kts ( Index_Wide_throat( i ) ) +
random_numbers ( Index_Wide_throat( i ) )*Wide_throat_exit_sigma;

   while Predicted_exit_speed_kts ( Index_Wide_throat( i ) ) < 0

   Predicted_exit_speed_kts ( Index_Wide_throat( i ),1 ) =  predicted_exit_speed_mean_kts ( Index_Wide_throat( i ) ) + randn ( 1,1
)*Wide_throat_exit_sigma;

   end

end
```

```
Index = [ Index_30 ; Index_90 ; Index_95_150 ; Index_Wide_throat ];

All_Index = transpose ( 1 : 1 : Total_observations );

% non-standard exits

Index_non_standard = find ( ismember ( All_Index, Index ) == 0 );

Number_non_standard_exits = length ( Index_non_standard );

for i = 1 : Number_non_standard_exits

   Predicted_exit_speed_kts ( Index_non_standard( i ),1 ) = predicted_exit_speed_mean_kts ( Index_non_standard( i ) ) + random_numbers ( Index_non_standard( i ) )*Standard_deviation_non_standard_exit;

   while Predicted_exit_speed_kts ( Index_non_standard( i ) ) < 0

   Predicted_exit_speed_kts ( Index_non_standard( i ),1 ) =  predicted_exit_speed_mean_kts ( Index_non_standard( i ) ) + randn ( 1,1 )*Standard_deviation_non_standard_exit;

   end

end



path_length_square = Path_length_ft.*Path_length_ft;

predicted_exit_speed_mean_2_kts = 4.681963 + 0.0514336*Path_length_ft - 1.689e-05*path_length_square;

for i = 1 : Number_high_speed_exits

   if ( Remaining_length_ratio ( Index_30 ( i ) ) < 0.35 )

   predicted_exit_speed_2_kts ( Index_30 ( i ),1 ) = predicted_exit_speed_mean_2_kts ( Index_30 ( i ) ) + random_numbers ( Index_30 ( i ) )*High_speed_sigma_remaining_length_ratio_less_than_035;

   while predicted_exit_speed_2_kts ( Index_30 ( i ) ) < 0

   predicted_exit_speed_2_kts ( Index_30 ( i ),1 ) =  predicted_exit_speed_mean_2_kts ( Index_30 ( i ) ) + randn ( 1,1 )*High_speed_sigma_remaining_length_ratio_less_than_035;

   end

   end


    if ( Remaining_length_ratio ( Index_30 ( i ) ) > 0.5 )

   predicted_exit_speed_2_kts ( Index_30 ( i ),1 ) = predicted_exit_speed_mean_2_kts ( Index_30 ( i ) ) + random_numbers ( Index_30 ( i ) )*High_speed_sigma_remaining_length_ratio_greater_than_05;

   while predicted_exit_speed_2_kts ( Index_30 ( i ) ) < 0

   predicted_exit_speed_2_kts ( Index_30 ( i ),1 ) =  predicted_exit_speed_mean_2_kts ( Index_30 ( i ) ) + randn ( 1,1 )*High_speed_sigma_remaining_length_ratio_greater_than_05;

   end

   end


    if ( Remaining_length_ratio ( Index_30 ( i ) ) <= 0.5  & Remaining_length_ratio (Index_30 ( i ) ) >= 0.35 )

   predicted_exit_speed_2_kts ( Index_30 ( i ),1 ) = predicted_exit_speed_mean_2_kts ( Index_30 ( i ) ) + random_numbers ( Index_30 ( i ) )*High_speed_sigma_remaining_length_ratio_between_05_035;

   while predicted_exit_speed_2_kts ( Index_30 ( i ) ) < 0
```

```
    predicted_exit_speed_2_kts ( Index_30 ( i ),1 ) =  predicted_exit_speed_mean_2_kts ( Index_30 ( i ) ) + randn ( 1,1 )*High_speed_sigma_remaining_length_ratio_between_05_035;

    end

    end

end


Index_90 = find ( Exit_angle_degrees < 95 & Exit_angle_degrees > 85 );

Number_right_angle_exits = length ( Index_90 );

for i = 1 : Number_right_angle_exits

    predicted_exit_speed_2_kts ( Index_90 ( i ),1 ) = predicted_exit_speed_mean_2_kts ( Index_90 ( i ) ) + random_numbers ( Index_90 ( i ) )*standard_deviation_right_angle_exits;

    while predicted_exit_speed_2_kts ( Index_90 ( i ) ) < 0

    predicted_exit_speed_2_kts ( Index_90 ( i ),1 ) =  predicted_exit_speed_mean_2_kts ( Index_90 ( i ) ) + randn ( 1,1 )*standard_deviation_right_angle_exits;

    end

end

for i = 1 : Number_obtuse_exits

    predicted_exit_speed_2_kts ( Index_95_150( i ),1 ) = predicted_exit_speed_mean_2_kts ( Index_95_150( i ) ) + random_numbers ( Index_95_150( i ) )*Standard_deviation_obtuse;

    while predicted_exit_speed_2_kts ( Index_95_150( i ) ) < 0

    predicted_exit_speed_2_kts ( Index_95_150( i ),1 ) =  predicted_exit_speed_mean_2_kts ( Index_95_150( i ) ) + randn ( 1,1 )*Standard_deviation_obtuse;

    end

end

%estimating the speed of wide throat exits

for i = 1 : Number_wide_throat_exits

    predicted_exit_speed_2_kts ( Index_Wide_throat( i ),1 ) = predicted_exit_speed_mean_2_kts ( Index_Wide_throat( i ) ) + random_numbers ( Index_Wide_throat( i ) )*Wide_throat_exit_sigma;

    while predicted_exit_speed_2_kts ( Index_Wide_throat( i ) ) < 0

    predicted_exit_speed_2_kts ( Index_Wide_throat( i ),1 ) =  predicted_exit_speed_mean_2_kts ( Index_Wide_throat( i ) ) + randn ( 1,1 )*Wide_throat_exit_sigma;

    end

end


% non-standard exits


for i = 1 : Number_non_standard_exits

    predicted_exit_speed_2_kts ( Index_non_standard( i ),1 ) = predicted_exit_speed_mean_2_kts ( Index_non_standard( i ) ) + random_numbers ( Index_non_standard( i ) )*Standard_deviation_non_standard_exit;

    while predicted_exit_speed_2_kts ( Index_non_standard( i ) ) < 0
```

```
    predicted_exit_speed_2_kts ( Index_non_standard( i ),1 ) =  predicted_exit_speed_mean_2_kts ( Index_non_standard( i ) ) + randn
( 1,1 )*Standard_deviation_non_standard_exit;

    end

end


% adding neural network model

Neural_Input ( 1 , : ) = transpose ( Path_length_ft );

Neural_Input ( 2, : ) = transpose ( Radius_ft );

Neural_Input ( 3, : ) = transpose ( Exit_angle_degrees );

network = fitnet (10);

trial_Observed_exit_speed_kts = transpose ( Observed_exit_speed_kts );

network = train ( network, Neural_Input, trial_Observed_exit_speed_kts );

view (network);

predicted_exit_speed_mean_3_kts = network ( Neural_Input );

for i = 1 : Number_high_speed_exits

    if ( Remaining_length_ratio ( Index_30 ( i ) ) < 0.35 )

    predicted_exit_speed_3_kts ( Index_30 ( i ),1 ) = predicted_exit_speed_mean_3_kts ( Index_30 ( i ) ) + random_numbers ( Index_30
( i ) )*High_speed_sigma_remaining_length_ratio_less_than_035;

    while predicted_exit_speed_3_kts ( Index_30 ( i ) ) < 0

    predicted_exit_speed_3_kts ( Index_30 ( i ),1 ) =  predicted_exit_speed_mean_3_kts ( Index_30 ( i ) ) + randn ( 1,1
)*High_speed_sigma_remaining_length_ratio_less_than_035;

    end

    end


     if ( Remaining_length_ratio ( Index_30 ( i ) ) > 0.5 )

    predicted_exit_speed_3_kts ( Index_30 ( i ),1 ) = predicted_exit_speed_mean_3_kts ( Index_30 ( i ) ) + random_numbers ( Index_30
( i ) )*High_speed_sigma_remaining_length_ratio_greater_than_05;

    while predicted_exit_speed_3_kts ( Index_30 ( i ) ) < 0

    predicted_exit_speed_3_kts ( Index_30 ( i ),1 ) =  predicted_exit_speed_mean_3_kts ( Index_30 ( i ) ) + randn ( 1,1
)*High_speed_sigma_remaining_length_ratio_greater_than_05;

    end

    end


     if ( Remaining_length_ratio ( Index_30 ( i ) ) <= 0.5  & Remaining_length_ratio (Index_30 ( i ) ) >= 0.35 )

    predicted_exit_speed_3_kts ( Index_30 ( i ),1 ) = predicted_exit_speed_mean_3_kts ( Index_30 ( i ) ) + random_numbers ( Index_30
( i ) )*High_speed_sigma_remaining_length_ratio_between_05_035;

    while predicted_exit_speed_3_kts ( Index_30 ( i ) ) < 0

    predicted_exit_speed_3_kts ( Index_30 ( i ),1 ) =  predicted_exit_speed_mean_3_kts ( Index_30 ( i ) ) + randn ( 1,1
)*High_speed_sigma_remaining_length_ratio_between_05_035;

    end
```

```
    end

end

Index_90 = find ( Exit_angle_degrees < 95 & Exit_angle_degrees > 85 );

Number_right_angle_exits = length ( Index_90 );

for i = 1 : Number_right_angle_exits

    predicted_exit_speed_3_kts ( Index_90 ( i ),1 ) = predicted_exit_speed_mean_3_kts ( Index_90 ( i ) ) + random_numbers ( Index_90 ( i ) )*standard_deviation_right_angle_exits;

    while predicted_exit_speed_3_kts ( Index_90 ( i ) ) < 0

    predicted_exit_speed_3_kts ( Index_90 ( i ),1 ) =  predicted_exit_speed_mean_3_kts ( Index_90 ( i ) ) + randn ( 1,1 )*standard_deviation_right_angle_exits;

    end

end

for i = 1 : Number_obtuse_exits

    predicted_exit_speed_3_kts ( Index_95_150( i ),1 ) = predicted_exit_speed_mean_3_kts ( Index_95_150( i ) ) + random_numbers ( Index_95_150( i ) )*Standard_deviation_obtuse;

    while predicted_exit_speed_3_kts ( Index_95_150( i ) ) < 0

    predicted_exit_speed_3_kts ( Index_95_150( i ),1 ) =  predicted_exit_speed_mean_3_kts ( Index_95_150( i ) ) + randn ( 1,1 )*Standard_deviation_obtuse;

    end

end

%estimating the speed of wide throat exits

for i = 1 : Number_wide_throat_exits

    predicted_exit_speed_3_kts ( Index_Wide_throat( i ),1 ) = predicted_exit_speed_mean_3_kts ( Index_Wide_throat( i ) ) + random_numbers ( Index_Wide_throat( i ) )*Wide_throat_exit_sigma;

    while predicted_exit_speed_3_kts ( Index_Wide_throat( i ) ) < 0

    predicted_exit_speed_3_kts ( Index_Wide_throat( i ),1 ) =  predicted_exit_speed_mean_3_kts ( Index_Wide_throat( i ) ) + randn ( 1,1 )*Wide_throat_exit_sigma;

    end

end

% non-standard exits

for i = 1 : Number_non_standard_exits

    predicted_exit_speed_3_kts ( Index_non_standard( i ),1 ) = predicted_exit_speed_mean_3_kts ( Index_non_standard( i ) ) + random_numbers ( Index_non_standard( i ) )*Standard_deviation_non_standard_exit;

    while predicted_exit_speed_3_kts ( Index_non_standard( i ) ) < 0

    predicted_exit_speed_3_kts ( Index_non_standard( i ),1 ) =  predicted_exit_speed_mean_3_kts ( Index_non_standard( i ) ) + randn ( 1,1 )*Standard_deviation_non_standard_exit;

    end

end
```

```
figure('units','normalized','outerposition',[0 0 1 1]);

cdfplot ( Observed_exit_speed_kts );

hold on;

cdfplot ( Predicted_exit_speed_kts );

hold on;

cdfplot ( predicted_exit_speed_2_kts );

hold on;

cdfplot ( predicted_exit_speed_3_kts );


xlabel (' Exit Speed at the point of curvature (knots) ');

ylabel (' Cumulative Density Function ');

set ( gca, 'FontSize', 24 );

leg = legend ( ' Observed Exit Speed (knots) ' , ' Predicted Exit Speed (knots) - Linear Model', ' Predicted Exit Speed (knots) - Quadratic
Model' , ' Predicted Exit Speed (knots) - Neural Network Model' , 'Location' , 'NorthWest');

leg.FontSize = 30;

legend boxoff;

title ('');

set(findall(gca, 'Type', 'Line'),'LineWidth',3);

savefig ( 'All_Exits.fig' );

Figure = getframe(gcf);

imwrite(Figure.cdata, 'All_Exits.png');

figure('units','normalized','outerposition',[0 0 1 1]);

% high speed records

High_speed_records = Predicted_exit_speed_kts ( Index_30 );

High_speed_records_2 = predicted_exit_speed_2_kts ( Index_30 );

High_speed_records_3 = predicted_exit_speed_3_kts ( Index_30 );

Observed_High_speed_records = Observed_exit_speed_kts ( Index_30 );

cdfplot ( High_speed_records );

hold on ;

cdfplot ( High_speed_records_2 );

hold on;

cdfplot ( High_speed_records_3 );

hold on;

cdfplot ( Observed_High_speed_records );

xlabel (' Exit Speed at the point of curvature (knots) ');
```

```
ylabel (' Cumulative Density Function ');

set ( gca, 'FontSize', 24 );

leg = legend ( ' Predicted exit speed for 30 degree exits (knots) - Linear model ' , ' Predicted exit speed for 30 degree exits (knots) - Quadratic model' , ' Predicted exit speed for 30 degree exits (knots) - Neural Network model' , ' Observed exit speed for 30 degree exits (knots) ' , 'Location' , 'NorthWest');

leg.FontSize = 30;

legend boxoff;

title ('');

set(findall(gca, 'Type', 'Line'),'LineWidth',3);

savefig ( 'High_Speed_Exits.fig' );

Figure = getframe(gcf);

imwrite(Figure.cdata, 'High_Speed_Exits.png');

figure('units','normalized','outerposition',[0 0 1 1]);

%Testing for right angle exits

Right_Angle_records = Predicted_exit_speed_kts ( Index_90 );

Right_Angle_records_2 = predicted_exit_speed_2_kts ( Index_90 );

Right_Angle_records_3 = predicted_exit_speed_3_kts ( Index_90 );

Observed_Right_Angle_records = Observed_exit_speed_kts ( Index_90 );

cdfplot ( Right_Angle_records );

hold on;

cdfplot ( Right_Angle_records_2 );

hold on;

cdfplot ( Right_Angle_records_3 );

hold on;

cdfplot ( Observed_Right_Angle_records );

xlabel (' Exit Speed at the point of curvature (knots) ');

ylabel (' Cumulative Density Function ');

set ( gca, 'FontSize', 24 );

leg= legend ( ' Predicted exit speed for right angle exits (knots) - Linear model' ,' Predicted exit speed for right angle exits (knots) - Quadratic model ' , ' Predicted exit speed for right angle exits (knots) - Neural Network model ' ,' Observed exit speed for right angle exits (knots) ' , 'Location' , 'NorthWest' );

leg.FontSize = 30;

legend boxoff;

title ('');

set(findall(gca, 'Type', 'Line'),'LineWidth',3);

savefig ( 'Right_Angle_Exits.fig' );

Figure = getframe(gcf);
```

```
imwrite(Figure.cdata, 'Right_Angle_Exits.png');

figure('units','normalized','outerposition',[0 0 1 1]);

%Testing for obtuse angle exits

Obtuse_Angle_records = Predicted_exit_speed_kts ( Index_95_150 );

Obtuse_Angle_records_2 = predicted_exit_speed_2_kts ( Index_95_150 );

Obtuse_Angle_records_3 = predicted_exit_speed_3_kts ( Index_95_150 );

Observed_Obtuse_Angle_records = Observed_exit_speed_kts ( Index_95_150 );

Index_obtuse = find ( Observed_Obtuse_Angle_records > 50 );

Obtuse_Angle_records ( Index_obtuse ) = [];

Obtuse_Angle_records_2 ( Index_obtuse ) = [];

Obtuse_Angle_records_3 ( Index_obtuse ) = [];

Observed_Obtuse_Angle_records ( Index_obtuse ) = [];

cdfplot ( Obtuse_Angle_records );

hold on;

cdfplot ( Obtuse_Angle_records_2 );

hold on;

cdfplot ( Obtuse_Angle_records_3 );

hold on;

cdfplot ( Observed_Obtuse_Angle_records );

xlabel (' Exit Speed at the point of curvature (knots) ');

ylabel (' Cumulative Density Function ');

set ( gca, 'FontSize', 24 );

leg = legend (' Predicted exit speed for obtuse angle exits (knots) - Linear model ' , ' Predicted exit speed for obtuse angle exits (knots)
- Quadratic model ', ' Predicted exit speed for obtuse angle exits (knots) - Neural Network model ', ' Observed exit speed for obtuse
angle exits (knots) ' , 'Location' , 'NorthWest' );

leg.FontSize = 32;

legend boxoff;

title ('');

set(findall(gca, 'Type', 'Line'),'LineWidth',3);

savefig ( 'Obtuse_Angle_Exits.fig' );

Figure = getframe(gcf);

imwrite(Figure.cdata, 'Obtuse_Angle_Exits.png');

% wide throat exits

if ( length ( Index_Wide_throat > 0 ) )

figure('units','normalized','outerposition',[0 0 1 1]);

Wide_throat_records = Predicted_exit_speed_kts ( Index_Wide_throat );
```

```
Wide_throat_records_2 = predicted_exit_speed_2_kts ( Index_Wide_throat );

Wide_throat_records_3 = predicted_exit_speed_3_kts ( Index_Wide_throat );

Observed_Wide_throat_records = Observed_exit_speed_kts ( Index_Wide_throat );

cdfplot ( Wide_throat_records );

hold on;

cdfplot ( Wide_throat_records_2 );

hold on;

cdfplot ( Wide_throat_records_3 );

hold on;

cdfplot ( Observed_Wide_throat_records );

xlabel (' Exit Speed at the point of curvature (knots) ');

ylabel (' Cumulative Density Function ');

set ( gca, 'FontSize', 24 );

leg = legend (' Predicted exit speed for Wide throat exits (knots)- Linear model',' Predicted exit speed for Wide throat exits (knots) - Quadratic model ' , ' Predicted exit speed for Wide throat exits (knots) - Neural Network model ' ,' Observed exit speed for Wide throat exits (knots) ' , 'Location' , 'NorthWest' );

leg.FontSize = 32;

legend boxoff;

title ('');

set(findall(gca, 'Type', 'Line'),'LineWidth',3);

savefig ( 'Wide_throat_Exits.fig' );

Figure = getframe(gcf);

imwrite(Figure.cdata, 'Wide_throat_Exits.png');

end
```