FIXED PATH PULL-IN/PUSHBACK TRAJECTORIES

FOR AIRLINER GROUND TRANSPORT

By

KARTHIK KUMAR RAJENDRAN

Bachelor of Engineering in Mechanical Engineering
Anna University
Chennai, India
2010

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2019

FIXED PATH PULL-IN/PUSHBACK

TRAJECTORIES FOR AIRLINER GROUND

TRANSPORT

Thesis Approved:

Dr. James Allan Kidd

Thesis Adviser

Dr. Jerome Hausselle

Dr. Shuodao Wang

ACKNOWLEDGEMENTS

I sincerely thank to my adviser Dr. James Allan Kidd for giving me the opportunity to pursue my graduate studies under him. His extensive support and guidance steer me in the right path during the thesis. I am very much indebted for his presence and without his unbounded knowledge, constant support and patience this thesis would not have been possible. I would like to express my gratitude to my committee members Dr. Jerome Hausselle and Dr. Shuodao Wang for their support and advise.

Name: KARTHIK KUMAR RAJENDRAN

Date of Degree: MAY, 2019

Title of Study: FIXED PATH PILL-IN/PUSHBACK TRAJECTORIES FOR AIRLINER GROUND TRANSPORT

Major Field: MECHANICAL AND AEROSPACE ENGINEERING

Abstract: The process of ground movement of airplanes about an airport is currently accomplished with a combination of engine thrust and ground vehicles. Minimization of on-ground fuel usage of airlines and reduction of ground support equipment hazards are desirable goals for airlines and airports. Fuel economy, pollution and noise as well as safety improvements are sought by a variety of technologies and operational approaches. In this research, an investigation is conducted into the range of potential paths for pulling airliners into their destination gates and pushing them back for engine start on departure through fixed path nose gear tracks. The feasibility of fixed paths is analyzed for this application and improved trajectories are identified. A kinematic model is developed to generate the trajectory of Main Landing Gear, wing and tail tips of an aircraft. Benefits and risks of system integration also discussed.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

LIST OF SYMBOLS AND ABBRIVATIONS

KOZ – Keep Out Zone

Delta (δ) – Step angle along the path when aircraft travels on a curved path

Theta (θ) – Angle of aircraft axis with respect to horizontal axis

Phi (φ)– Angle of track point with respect to horizontal axis

Alpha (α) – Nose gear angle

X, Y – Position of any point (drive, trace, etc.) in cartesian coordinates

L – Distance from nose gear to center of main gear along the aircraft axis (Wheel base)

NLG – Nose Landing Gear

MLG – Main Landing Gear

CHAPTER I

INTRODUCTION

Large and expensive airliners moving in constrained and congested areas near terminal creates the potential for damage and injury [3] as well as operational delays [10]. The need for safe and efficient aircraft terminal movements has motivated a number of different interesting approaches. The current method used for moving an aircraft in a highly dynamic environment is a difficult procedure due to limited visibility and lack of monitoring equipment. One of the responsibilities of ground personnel carrying out a pushback operation is to ensure that no part of the aircraft structure will impact with any other fixed object or aircraft. Prediction of aircraft trajectory is considered as a safety measure. A fixed track towing system has been proposed [2] to offer improvements in safety and reduction in required ground support equipment and personnel. This might be an efficient way for ground maneuver by reducing fuel usage and noise pollution. One significant question is whether such an architecture could provide pull-in and push-back in typical existing airport configurations. This thesis examines the feasibility of such an architecture with respect to typical gate configurations, potential track layouts and resulting airplane ground trajectories.

Push back is an aircraft procedure in aviation where an airplane parked at a terminal gate is connected to tug via a tow bar to its nose landing gear and pushed away from the gate out to the taxiway. The aircraft then proceeds by its own engine thrust. The aircraft engines are typically not started or are left at idle thrust while at the gate. This reduces the likelihood of damage caused by

engine exhaust to the nearby aircraft, equipment and structures. Taxiing is currently controlled manually by the pilot using nosewheel steering, differential braking and engine thrust.

The primary focus of this research is to find the trajectories of various points on an aircraft during pull-in or push back when the nose gear follows a predefined path. Also, the feasibility of the predefined path is analyzed as a fixed track system for aircraft ground maneuvers. This analysis defines the space needed to push back an aircraft without impinging on Keep-Out Zones (KOZ's). KOZ's are the minimum safe distances around adjacent objects such as other aircraft, buildings, ground maneuvering vehicles. As a part of this research an improved track will be find out to minimize track length and ramp space required for this operation.

Detailed discussions about the aircraft trajectory modelling, feasibility analysis of predefined path, experimental validation and determination of improved path are presented in this thesis report. To get some basic knowledge about aircraft ground maneuvers, various specifications and components involved in push back and pull -in operations are explained in this chapter.

Safety is one of the most important factors in near-terminal operations due to the potential for severe injuries to ground support personnel and the high cost of damaged aircraft. According to a survey [3], during push back at US airports in the period 1991-2011, 31.8% of 189 reported incidents/accidents involved collision of an aircraft (that is being pushed back) with stationary/moving/pushed object. With the need to increase airport capacity, reduce ground cycle times and improve ground safety, there is interest in automating portions of the ground operations.

Figure 1.1: Aircraft collision during push back [27]

Using Boeing 737-800 as an example, the fuel consumption is to be 2530 kilogram of jet fuel per hour [40]. Auxiliary Power Unit (APU) in the aircraft needs to keep running to supply air conditioning, electricity and hydraulic power during this entire pull-in/push back operation. Since the number of flights are increased for the same airport size it is necessary to reduce the taxi time. To achieve this there are new systems are to be installed as an alternative such as Electric Landing Gear [24], robotic tugs [25], and fixed path towing systems such as Aircraft Towing System (ATS) [2].

1.1 RESEARCH OBJECTIVES

The following are the objectives of current research

i) Modelling and simulation of tracking main gear, wing and tail tips of an aircraft during towing operations.

ii) Determine the feasibility of an airliner to pull-in and push back in a fixed track configuration in typical airport configurations and with typical airliners.

iii) If feasibility is shown, then seek to minimize track length and ramp space required for pull-in and push-back.

1.2 AIRCRAFT GROUND OPERATIONS

Aircraft ground operations include moving the aircraft to and from the active runway, parking at the terminal and servicing while parked. As the aircraft lands and is taxied near its assigned gate, ground personnel guide the pilots as they maneuver to final parking position at the gate. The parking operation is usually done using engine thrust. Once the aircraft is park and engines shut down, the board bridge is moved into position. Passengers are unloaded and new passenger loaded through this bridge. Meanwhile, conveyor units are placed at the aircraft baggage bays. Baggage is unloaded and loaded by using the conveyors. Fueling is accomplished via trucks or from fixed emplacements at the parking location. Then the aircraft is cleaned between passenger unloading and loading and catering supplies are loaded for new passengers.

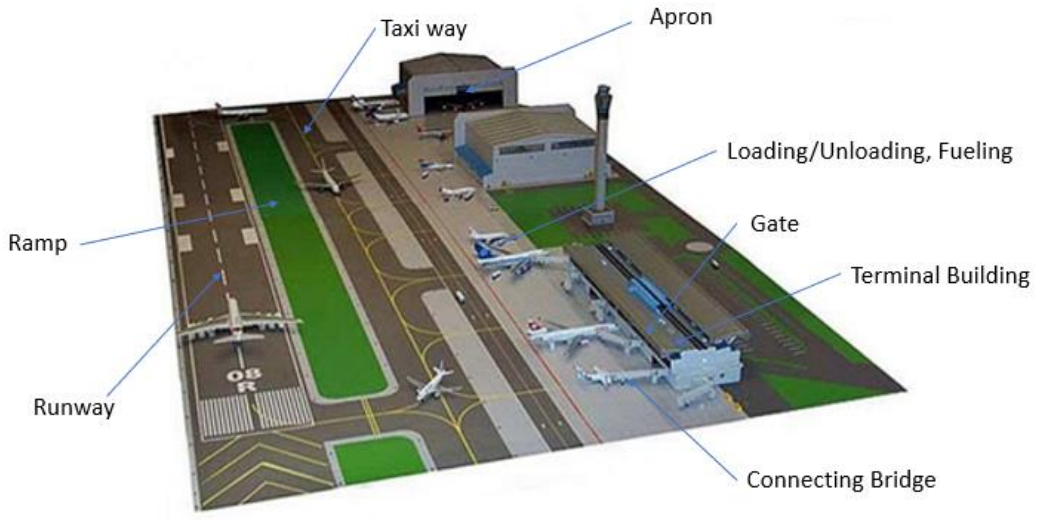


Figure 1.2: Typical Airport Layout [26]

Once everything is done, all the safety points are checked before the aircraft is pushed back from gate. A tug is typically used to push back the aircraft from gate to the taxiway. Engine thrust may

sometimes be used but it increases the risk of damage to adjacent aircraft, ground support equipment and terminal buildings while moving the aircraft in reverse. Typically, a tow bar is used to connect the aircraft and tug to push or pull the aircraft. The tow bar is disconnected from the aircraft after it reaches the target location. Then the aircraft can be moved on its own power to the runway for takeoff. Key steps of the towing operation include positioning the tug, connection of the tow bar, disabling the aircraft nose gear hydraulic steering system, maneuvering the tug to generate the desired aircraft trajectory, maintaining proper clearance from adjacent equipment, structures and personnel, disconnecting the tow bar and finally re-enabling the nose gear steering system. Communication between ground crew and pilots may be via wired intercoms and hand/light signals. This is the traditional method carried out in most of the airports which is inefficient in fuel usage and personnel.

Figure 1.3: Boeing 737-800 with tug [2]

Most airport terminal areas are designed to have aircraft parked in parallel at the gates [11] [13]. In these conditions, the distance between adjacent aircraft is small and push back/pull-in operations must be accomplished with great care. Small variations (0.1°) from the desired path may lead to collision with adjacent airplanes.

1.3 FIXED PATH TOWING SYSTEM CONCEPT

One concept put forward to reduce ground support equipment and personnel for terminal operations is to have a fixed track towing system that automatically engages, moves and disengages with the aircraft. The Aircraft Towing System (ATS) is a proposed fixed path, electrically driven infrastructure for moving an aircraft from gate to runway and runway to gate. The ATS approach is unique in that most of the system is underground with only the airplane interface (dolly) above the surface. The dolly has a rotating plate and clamping system to support and retain the airplane nose landing gear (NLG).

Figure 1.4: Aircraft Towing System Architecture [2]

The dolly rotating plate allows the NLG to remain angularly stationary with respect to the airplane while the dolly is propelled along its path. This eliminates the need for ground support personnel interaction with the airplane, e.g., pulling or inserting pins to disable the NLG steering when attaching to a tug. Front and rear ramps are included on the dolly to allow loading and unloading in both directions. The pilot will need maneuver the aircraft in alignment with the dolly and drive the nose landing gear onto it. The clamping plates will then compress the NLG tires and the system will be ready to pull or push the aircraft. At this point the pilot will only need to remain off the

brakes unless an emergency stop is required. The system is sized to handle aircraft from regional jets up to jumbo jets. There are no aircraft modification required.

The underground mechanism consists of a channel built in the airport taxiway requiring a modification to the ramp and taxi areas. A pull car system consist of electrical drive mechanism is installed in the channel below the dolly. Software and sensors are integrated in the ATS system to eliminate the human requirement in the ground maneuvers. A highly critical aspect of the ATS and any fixed track system is to have NLG paths that ensure all parts of the airplane remain clear of KOZ's for all operations. The question is whether there are feasible paths that provide this required KOZ clearance.

CHAPTER II

REVIEW OF LITERATURE

In this chapter a detailed study of previous work done in relation to pushback trajectories is explained. The complete literature survey is elaborated on different segments like tracking the main gear path, mathematical approach, safety, and autonomous taxiing.

During earlier times, scaled models of aircraft undercarriages were fabricated and various ground maneuvers were evaluated using them. [5]. Full-size models were used for more complex problems associated with pull-in/pushback of the aircraft [5]. Later, mathematical solutions performed on digital computers were developed for use in design and analysis of new aircraft configurations [5] The analytical models typically used either the Kinematic approach or the Dynamic approach to model the aircraft trajectories. In the kinematic approach, the trajectories are modeled based on the geometric representations without reference to any forces involved. The dynamic approach includes some modelling of forces and dynamic motion. Many factors may be included, such as aircraft mass distribution, landing gear configuration, physical characteristics of aerodrome, aircraft tire characteristics, and weather conditions [7]. Since solving dynamic model of aircraft maneuver is a complex process and the differences between the kinematic and dynamic models are likely to be small when considering the very low speed maneuvers. Because, the effects of friction and forces on dynamic model is less in low velocities compare to high velocity, so kinematic model is used in this study. Moreover, this research mainly from the perspective of modelling and analysis of aircraft trajectories for a fixed NLG path, which simplifies the solution process by kinematic

approach with respect to dynamic process. Three initial parameters are required to calculate the main gear trajectory. One is initial position of the nose gear, second is the predefined path and third is the distance between nose gear and center point of the main gear (Wheel base) along the aircraft axis.

In general, the aircraft will not travel in a fixed path while it is being pushed back with a tug. This research deals with the feasibility analysis of aircraft trajectories during pull-in/push back in a predefined path. According to Ezra Hauer [4], when two aircraft are travelling in a defined NLG path, the MLG of the aircraft with the shorter wheel base will deviate less from the NLG path than the aircraft with the larger wheel base. It is important to compute the trajectory of main gear while designing and constructing an airport. Taxiway turns are designed based on the main gear path to prevent dropping the main gear off the edge. Figure 2.1 shows an MLG that has departed the hard taxiway surface. The main gear position is determined by calculating the center point (along the axles) of the main gear along the aircraft axis. The determination of trajectory can be accomplished by using graphical means, small scale models, Tables and graphs and analytical models.

Figure 2.1: MLG off the prepared taxiway [28]

Information about the main gear track in constant radius turns can be obtained from the aircraft manufacturer in their airport operations documents produced for each type of aircraft [18]. The

guidelines for developing taxi tracks for airplane are: The aircraft wheel base should be larger than the turning radius and the nose deflection angle should be higher than the allowed angle to perform the ground maneuvers [5]. These guidelines will provide the possibility of the center point of main gear of large and small size aircraft would trace a common trajectory for a predefined path.

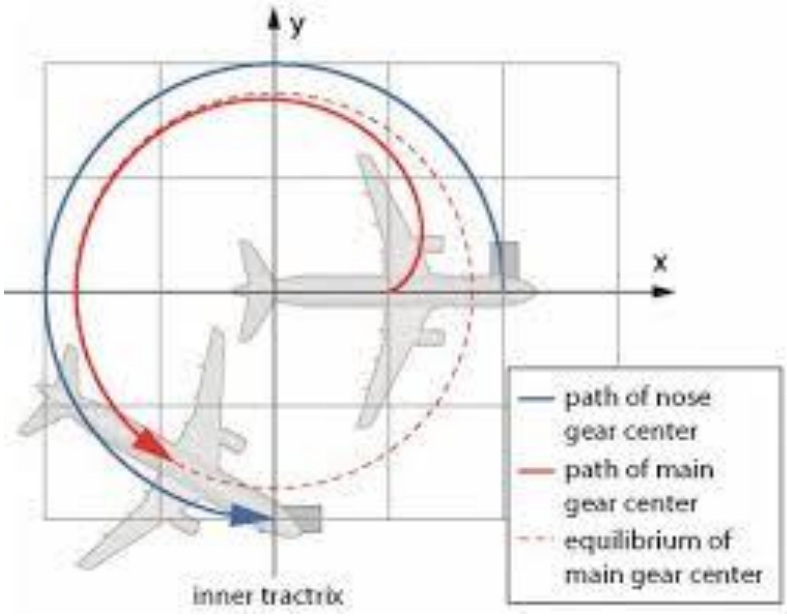


Figure 2.2: Inner tractrix of an aircraft towed along a circular path [3]

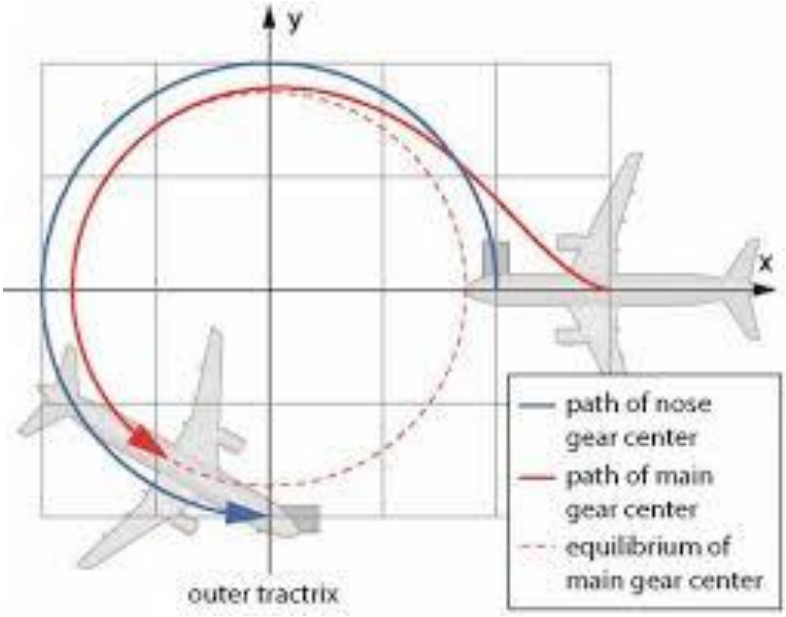


Figure 2.3: Outer tractrix of an aircraft towed along a circular path [3]

The basic concept of tracing the main gear is taken by referring the detailed analysis of the path traversed by the Ackerman drive of a tricycle or rear wheel of bus. The path of the rear wheels of a bus or long vehicle in general, is a tractrix or equi-tangential curve, which is different than the traces formed by the front wheels [6]. It is easier to track the rear wheel when the vehicle travels in a straight path than when the vehicle travels in a curved path. This curved path creates a curved tractrix which can be described in to inner tractrix and outer tractrix. These outer and inner cases are interesting for modelling of pushback trajectories. The right tractrix must be chosen to produce the desired movement along a path. Figures 2.2 and 2.3 show the inner and outer tractrix of an aircraft towed on a circular NLG path.

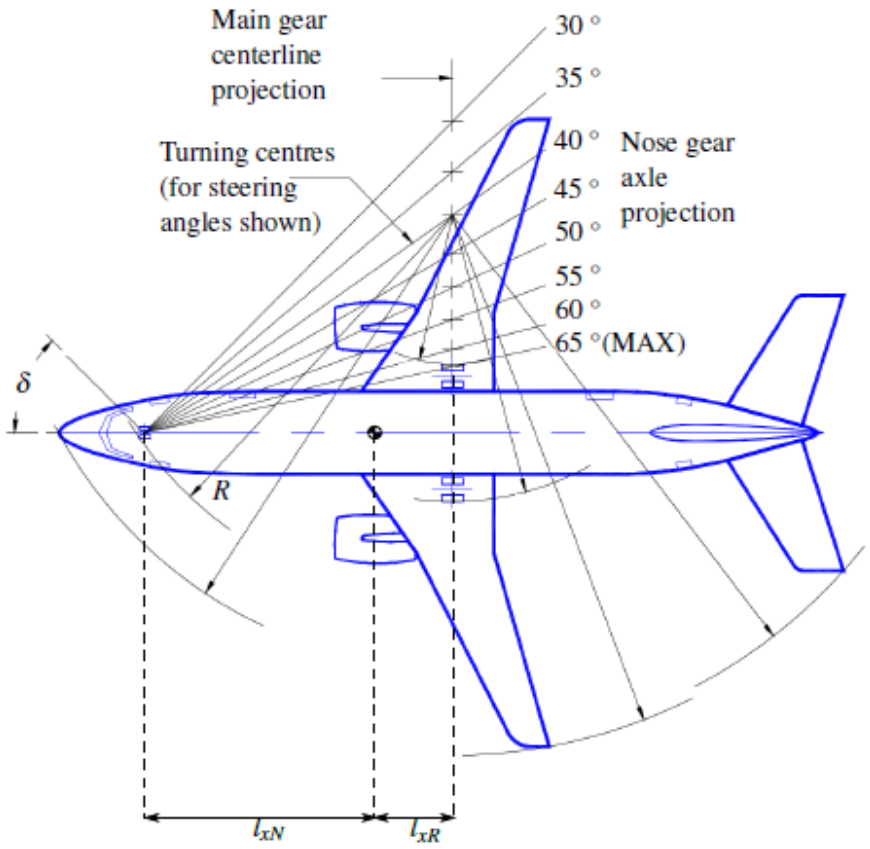


Figure 2.4: Ackerman tricycle and turn radii for Boeing 737-400 [8]

The Ackerman drive method assumes that the lines perpendicular to wheel axles meet at one point, denoted as the Instantaneous Center of Curvature when the vehicle is turning. As the rear wheels are fixed in orientation with respect to the airplane axis, this center point must be on a line extended from the rear axles (perpendicular to the airplane longitudinal axis) [8].

In some airports, more complex maneuvers are performed, the aircraft needs to make a U-turn for take-off in a desired direction (opposite direction from its current position) and there might be space restrictions to push back and pull-in to change the aircraft's direction. During this condition the aircraft need to be pulled out of the constrained space and make a U-turn to achieve its final orientation. Most of the time ,the U-turn occurs near the runway and the aircraft needs more swept area than it needs during 90° turn. There are two methods involved in this U-turn maneuver. One is the aircraft do maneuver when taxiing along the edge of the runway. Another is that taxiing laterally along from center line of runway to the edge of the runway [7] and figure 2.5 shows an aircraft U turn maneuver. In this situation the NLG may need turn more than its allowed turning angle when the aircraft tuns on its own. To overcome this limitation, the NLG is unpinned to rotate at desired angle when the aircraft towed by tug.

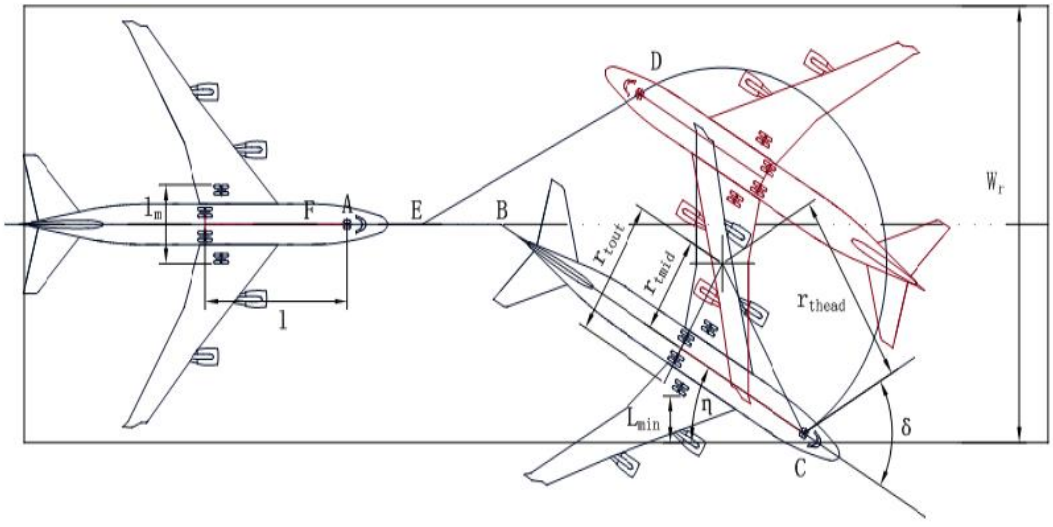


Figure 2.5: Aircraft U turn maneuver [7]

Students at Purdue University designed an autonomous taxiing system for a design competition conducted by Federal Aviation Administration in 2010 [10]. The "Automated NextGen Taxi System" (ANTS) was designed to fill a gap between current pilot-based, engine-propelled taxi and a fully automatic taxi system. The objective was an autonomous towing system for airports to increase efficiency of push-back/pull-in operations while maintaining safety. It is composed of two systems: Data Management System and the robotic tractors or Towing Support Vehicles. This Towing Support Vehicle is designed to autonomously attach and detach the aircraft into the vehicle. This vehicle is a towbarless tug. Tow bars require ground personnel to adjust the tow bar height, connect tow bar to the nose wheel and lock the connecting pin in place [10]. These actions consume valuable departure time. The towbarless design is used with the goal of reducing this time. Note that towbarless systems still take time to approach, align, and secure the NLG for towing and to release after tow is complete. It took around 62 seconds for a towbarless tug (expediter 600) to align and assemble with A380 for push back operation [28]. This process can be done only when the nose gear wheel and tug are aligned accurately. Meanwhile the pilot can stop the Towing Support Vehicle when there is an emergency.

CHAPTER III

MODEL DEVELOPMENT AND VALIDATION

Detailed description about mathematical development of model, experimental procedure and validation are discussed in this chapter. A number of airports and airplanes are studied to develop the analytical model. After finalizing the initial details, such as paths and aircraft size, a mathematical model is developed to estimate the trajectory of the aircraft. This model is then implemented as a computer code to provide means of user input and simulation result output.

3.1 AIRPORT TERMINAL SURVEY

To perform this modelling there are several airport terminal geometries and airplane service types are examined to develop a "standard" terminal model. These international airports varied form simple straight terminal to more complex arrangements. Airport terminals examined thus far include Dallas-Fort Worth (DFW), Alexandria International Airport (AEX), Tampa International Airport (TPA), Orlando International Airport (MCO), Miami International Airport (MIA) and John F Kennedy International Airport (JKF). The geometries were extracted using Google Earth™ [11] images and scales. The model to be developed from this examination, will define the gate spacing, jetway locations, taxiway lines and other constraints on the airliner path. With KOZ's defined, the remaining available airport surface area is considered for the fixed path system. Figure 3.1 shows the layout of Alexandria International Airport (AEX) with a simple arrangement of gates along a straight terminal with ample ramp area adjacent to them. Note the KOZ's markings, pull-in lines and stop marks at the gates. According the airport website ERJ 135, ERJ 145 and CRJ 2 [12] are

the aircraft models that are currently operated in this airport. The terminal appears capable of supporting larger aircraft with little or no modification of the gates.

Figure 3.1: Alexandria International Airport [11]

Figure 3.2 shows the north side of Terminal D at the Dallas Fort Worth International airport (DFW) with more complex geometry and a mix of small, medium and large airliners. Note the path markers for aircraft taxi as well as lane markers for service vehicles. The lower right-hand corner shows a particularly challenging location for pushback.

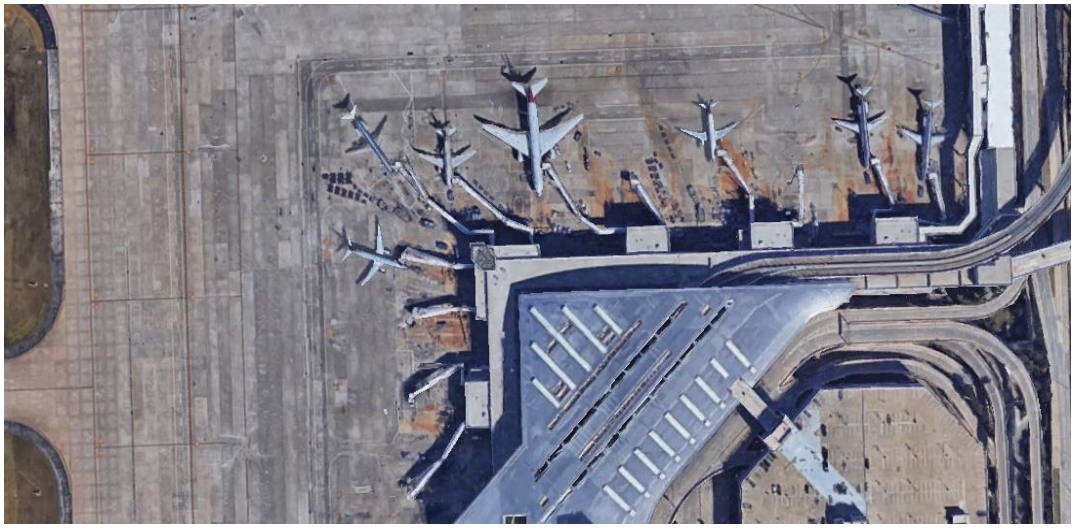


Figure 3.2: Dallas-Fort Worth International Airport [13]

The airports considered above serve a wide variety of aircraft. This study focuses on turbine engine airliners from regional jets up to wide-body jumbos. Airport terminal service areas are often constructed to serve some range of sizes with considerable overlap, e.g., regional jets through double aisle, double aisle through jumbo. As an initial step the Alexandria International Airport (AEX) is considered to do analytical calculations. This airport has the capacity to handle several aircraft sizes. A reasonable aircraft size should be selected to conduct the analytical calculations and experiments. Following aircraft are examined in terms of categories, passenger capacity and production summary.

Aircraft	Categories	Passenger Capacity	Number of aircrafts produced
ERJ 135	Medium Regional	37	1219
CRJ 200	Medium Regional	50	185
CRJ 700	Medium Regional	70	104
B 737	Narrow Body	189	3132
A320	Narrow Body	515	5267
B 757	Narrow Body	279	1050
A300	Wide body	243	567
B 767	Wide body	370	1129
B 787	Wide body	440	809
A340	Wide body	269	380
B 777	Wide body	180	1585
B 747	Jumbo	313	1549
A380	Jumbo	555	238

Table 3.1: Example Aircraft details

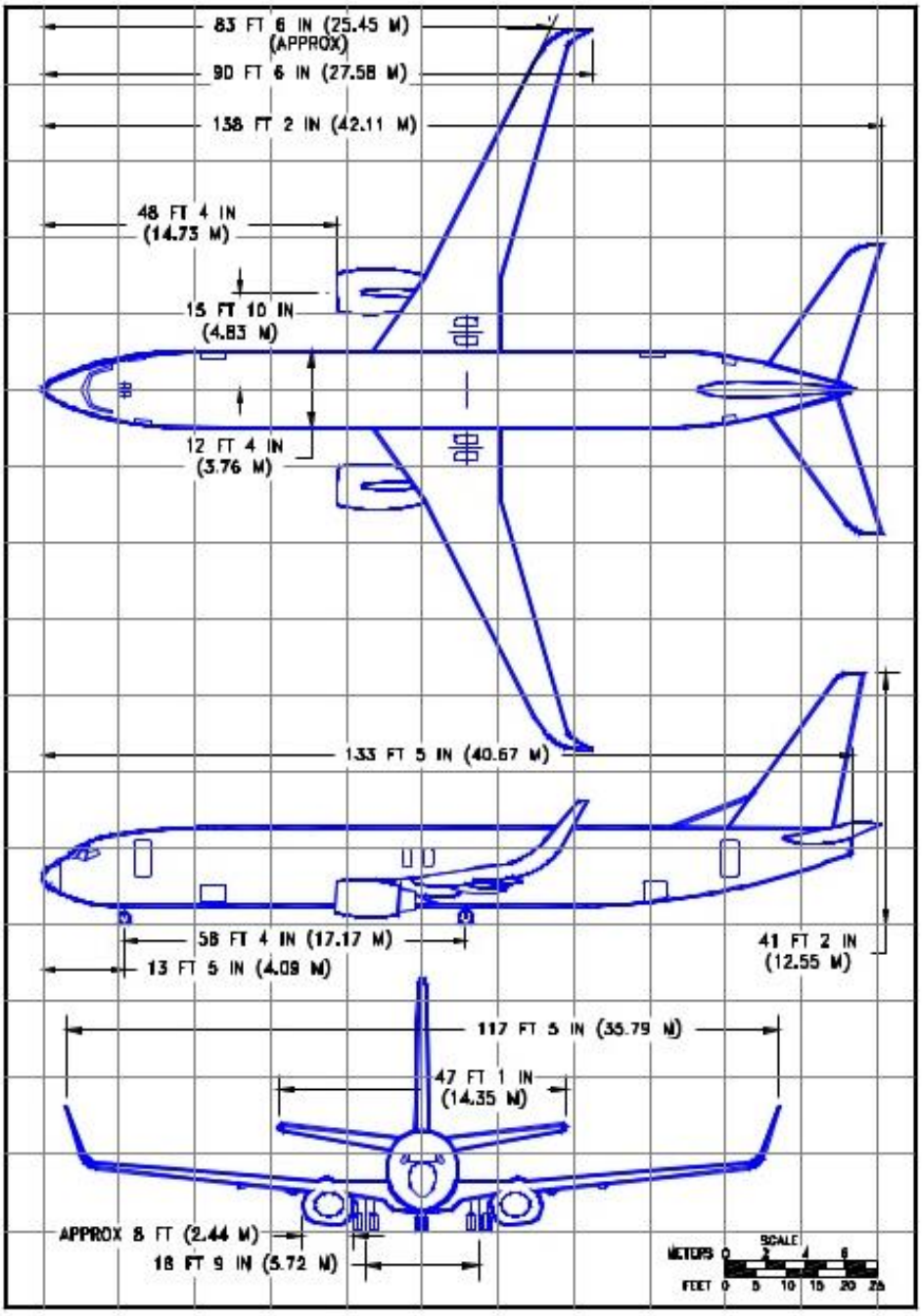
Boeing 737-900ER aircraft is selected as a “typical” model for which to conduct the initial feasibility assessment. It is a twin-engine airplane designed to operate over short to medium range from sea level runways of less than 6000 ft in length [18]. The services required for the 737 airplanes can be accomplished by the standard ground handling equipment. There are two sizes available in 737-900 model: The 737-900ER and 737-900ER with winglets. The winglets have an additional wingspan of 4.98 ft. They both are capable of carrying up to 189 passengers [18]. The landing gear system is conventional tricycle type system which has a dual-wheel NLG at the front

and a pair of four-wheel MLG. Figure 3.3 represents the general dimensions of B737-900 aircraft [18]. It is observed from Table 3.1, total number of aircraft produced for B737s is higher compared to other sizes. This makes the selection of typical aircraft among all aircraft is more reasonable and it is in service at several places in the world. Table 3.2 shows the example set of aircraft with length, wingspan and wheel base normalized using the B 737-900ER dimensions as the units.

Aircraft	Total Length (B737 Units)	Wing span (B737 Units)	Wheel base(B737 Units)
ERJ 135	0.626	0.56	0.725
CRJ 200	0.636	0.594	0.665
CRJ 700	0.772	0.65	0.875
B 737-900ER	1	1	1
A320-200	0.893	1.001	0.737
B 757-300	1.293	1.064	1.302
A300-C4	1.271	1.253	1.08
B 767-400ER	1.458	1.451	1.524
B 787-10	1.622	1.68	1.682
A340-600	1.79	1.773	1.915
B 777-300ER	1.754	1.811	1.819
B 747-8	1.787	1.912	1.728
A380-800	1.728	2.229	1.857

Table 3.2: Aircraft dimension in B737-900 units

From the airport terminal survey, it is observed that the airplanes are very often parked parallel to each other. This provides an ideal situation for pushback an aircraft in between two airplanes. More over the taxi ways are constructed in parallel to the terminal buildings which makes the aircraft need to make a 90° turn for taxi. After observing these details, an analytical model is developed to pushback and pull-in the aircraft in a path makes 90° turn in between two aircraft. This will be used to assess the feasibility of the predefined path to push back the aircraft as well as to define the KOZ's.



2.2.14 GENERAL DIMENSIONS

MODEL 737-900, -900ER WITH WINGLETS

D6-58325-6

JULY 2007 41

Figure 3.3: Boeing 737-900 ER Dimensions [18]

3.2 TRAJECTORY MODELLING

A key objective of this thesis is development of a model to estimate the airplane motion and swept area for given initial conditions on a fixed NLG path towing system. This model will then be applied to determine feasibility for pull-in and push back within the constraints of a typical terminal and gate area. Trajectory modelling is based on the geometric relations among aircraft wheel base, nose gear angle and path that the aircraft is to traverse. The kinematic analysis of the position of main gear, wing and tail tips will give an overall space required to pushback/pull-in an aircraft in a certain path. This trajectory is modelled by using the geometric relation of the nose gear, main gear and wings with respect to the nose gear angle and path segments. To do the modelling and simulation, the required input details are defined. The model of an aircraft is represented as a simple diagram and required points are defined. Figure 3.4 shows the simplified model representation of an aircraft and small aircraft picture is included for better understanding of simplified representation. These dimensions are represented from the nose gear and the X and Y values of each points are given as an input to the simulation.

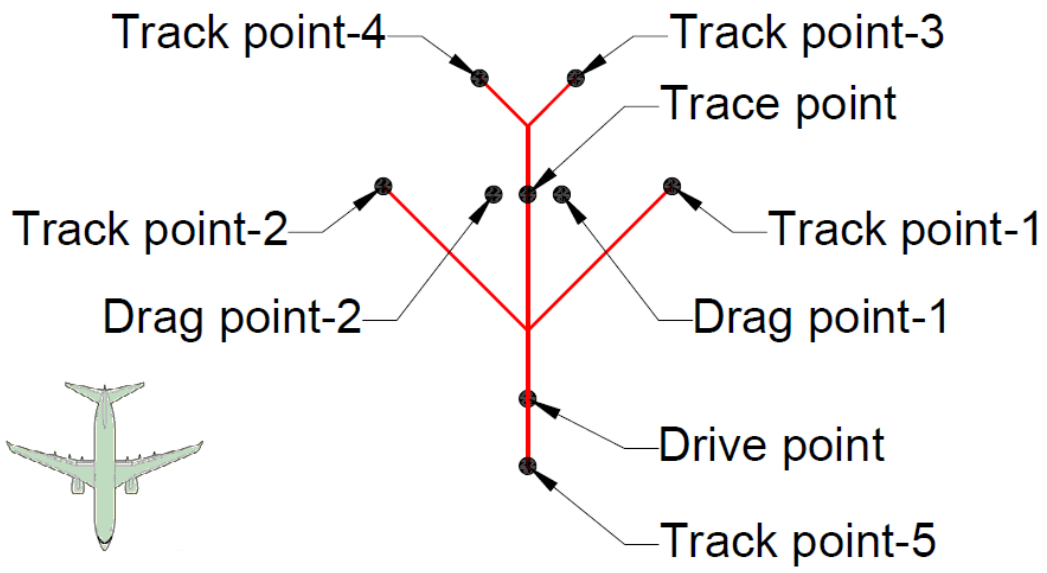


Figure 3.4: Point Definition for Airplane Model Geometry

Drive point – Center of the Nose Landing Gear

Trace Point – Center point of Main Landing Gear along the aircraft axis

Drag points 1 & 2 – Main Landing Gear positions

Track points 1 & 2 – Wing tips

Track points 3 & 4 – Tail tips

Track point 5 – Nose tip

To simplify the calculations the followings assumptions are made as per the methods of [7] [19]

1. All locations are considered as points.
2. Distances between points on the aircraft are constant (rigid body).
3. Nose gear rotates to 360 degrees without any restrictions as it moves along the path.
4. No inertial, wind, propulsion or friction effects.

Although the tires on the aircraft is in touch with the ground as in surface, the virtual center point of the tire at the intersection between the line parallel to axis of tire and the ground is considered as a drag point. The effects of considering inertial and wind may overturn the main gear which is drag point. The aircraft my deviate from it geometric position due to friction between the tire on the main landing gear and ground surface. When it comes to nose gear, the nose gear tire is clamped to the rotating plate in dolly. The positions of the track and drag points are calculated by considering the center point of NLG is in align with the center point of the rotating plate in dolly. The NLG in the aircraft is not capable of rotating 360 degrees on its own, so the rotating plate in the dolly will rotate a complete revolution. This allows the NLG to rotate with respect to aircraft's orientation as the dolly travels along the predefined path. There will be friction when the main gear tire rolls on the ground which will affect the pushback operation to some extent. This friction is likely to vary with the weather (rain, ice), tire size, ground formation, acceleration, material properties and pattern

of tires, etc. Since this simulation is based upon the kinematic approach the friction and other factors that affect the orientation and position of the aircraft is neglected. After considering all these assumptions, this kinematic approach is modelled in in Excel™ with a Visual Basic macro for inputs of NLG path, airplane geometry and initial condition of nose gear angle. The code and example worksheets are included in Appendix A. The next section describes the specific steps within the code along with the user interface.

3.3 GEOMETRIC REPRESENTATION

The angular position of drag and track points remains same with respect to global positioning when the aircraft travels along the straight path. This phenomenon occurs because the aircraft is considered as a rigid body [8]. This section discusses about the change in orientation of the aircraft when the aircraft is travelling along a curved path. Aircraft needs to travel from simple straight and curved paths to complex paths like combination of both during taxi process. The transition from straight to curve or vice versa should be smooth. In general, there are two cases in turning of an aircraft [3]. One is the turning radius is greater than the wheel base and second is the turning radius is lesser than the wheel base. The latter case is uncommon in near gate operations, so this analysis will consider only the first case. This means the paths generated have a constant radius greater than the wheel base.

Nose gear angle increases as the aircraft travels along the curved path and comes back to its original position at the end of the curve. This happens because the nose gear is mounted on a rotating plate which rotates along the curve and keeps the nose gear aligned with the aircraft. To determine the trajectory, the position of trace point relative to the drive point must be calculated [4]. Figure 3.5 shows the geometric representation of an aircraft travelling along the curved path with one step angle. The angles represented in the figure is used to calculate the push back of an aircraft along a curved path. The aircraft starts rotating in clockwise direction with respect to the trace point for

each step angle as it moves along the path. It will turn into to 90 degrees as it reaches the other end of the path.

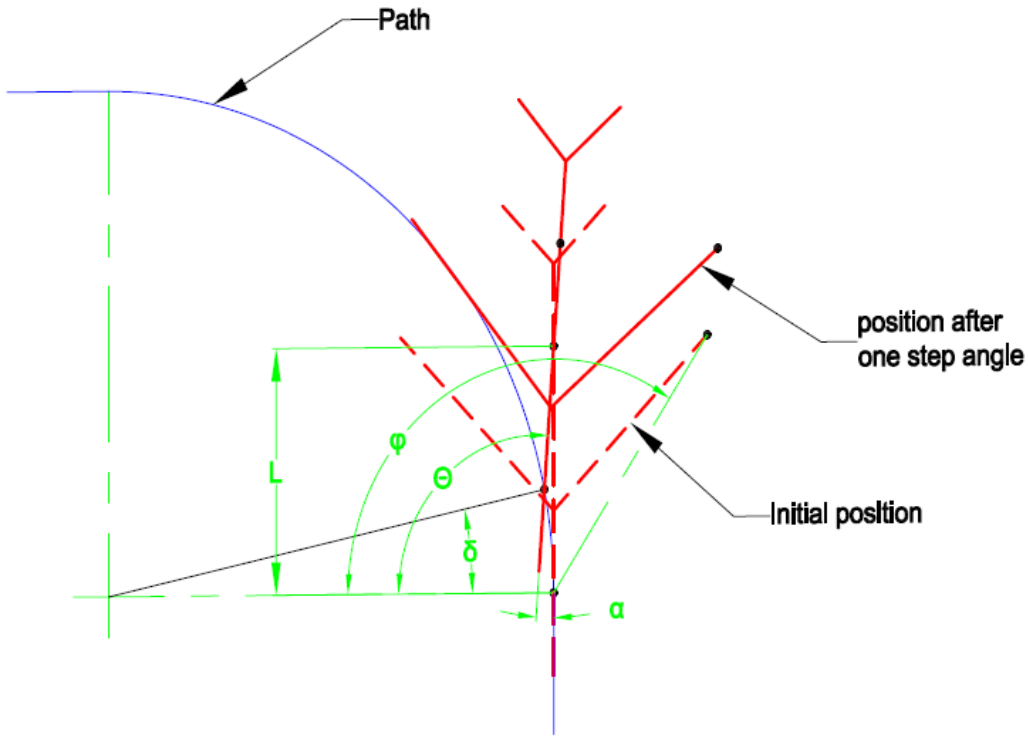


Figure 3.5: Geometric Representation

The code for this simulation works on the basis of drive point, path and the trace point. Initially the drive point is located at the beginning or start position of the path. Now, the drive point moves to one step angle along the path for travel by keeping the trace point as an origin. The step angle depends on path radius, initial and turn angles. After the drive point moves a step, this creates a new trace point as the distance between drive point and trace point is constant (Rigid body). All required angles are calculated, and the positions of each point is noted down. Now the new drive point becomes an initial point of the aircraft and again it moves for one step angle by keeping trace point as an origin. This process continues until the drive point reaches the end position of the curved path. Proper constraints are provided in the code to avoid the error occurred for ambiguity.

3.4 MODEL EQUATIONS

Equations are developed suitable for both pushback and pull-in operations. The aircraft is pushed back, up to certain point in the path and then it is pulled along the path. This push and pull while travelling in a single path occur based on several factors like initial nose angle, path formation and aircraft size. Primarily the equations are developed to calculate the positions of each point with respect to the drive point. For example, to find the instantaneous position of trace point with respect to drive point the following equations are used under several conditions. The angle (θ) between the drive and trace point is calculated first in order to find out the instantaneous position of the trace point.

$$\theta = \arctan\left(\frac{y2-y1}{x2-x1}\right) \tag{3.1}$$

$$Xt = Xd - L * \cos(\theta) \tag{3.2}$$

$$Yt = Yd - L * \sin(\theta) \tag{3.3}$$

Xt – Instantaneous X coordinate of Trace point

Yt – Instantaneous Y coordinate of Trace point

Xd – Instantaneous X coordinate of Drive point

Yd – Instantaneous Y coordinate of Drive point

Where x1, y1, x2, y2 are the X and Y coordinate values of drive and trace points respectively. These values of coordinate points can be calculated by using the input values of initial position and wheel base length (L). When the aircraft is positioned in vertical ($\alpha = \pm\pi/2$) the value of the denominator becomes zero in Equation 3.1 which leaves the θ value as invalid. So, suitable conditions are provided in the algorithm to avoid such errors. To calculate the position of drag and track points, the initial locations of these points are calculated with respect to drive point. Unlike trace points, the drag and track points are located at a certain angle from the aircraft axis. The angle

(φ) of those positions are calculated by using the equation 3.1 shown above. Then the initial position can be calculated as [6]

$$\text{X coordinate of Track/drag position} = -\text{Length}^* \sin(\varphi-\theta) \tag{3.4}$$

$$\text{Y coordinate of Track/drag position} = \text{Length}^* \cos(\varphi-\theta) \tag{3.5}$$

Here the length is the distance between the drive point and Track/Drag point for which the location needs to be found. When the nose angle is zero, this represents the angle between the predefined path and the aircraft axis is zero. So, the aircraft is aligned with the path and the angle between the path/aircraft axis and drag/track points is same. But, when the nose angle is not zero, this means the aircraft axis is positioned at an angle from the path. Which increases the angle of drag/track points from the path. These angle differences are considered to find out the initial position of the drag and track points with respect to the drive point. Once the initial position is calculated the results are noted down in the simulation and the drive point is moved to the next position in the path to calculate the new positions of the drag/track points. When the drive point moves from its initial position, the drag and track points moved along with the drive point. This instantaneous positions of drag and track points can be calculated when the aircraft travels along the predefined path. This can be done using following equations,

$$Drag(X') = Drag(X) * \text{Sin}(\theta) + Drag(Y) * \text{Cos}(\theta) + Drive(X) \tag{3.6}$$

$$Drag(Y') = -Drag(X) * \text{Cos}(\theta) + Drag(Y) * \text{Sin}(\theta) + Drive(Y) \tag{3.7}$$

Here X' and Y' are the instantaneous positions and this is applicable to calculate track and drag points at its respective position. The formulation is developed to find the coordinate values of track/drag points and these values are noted down on the respective cells on the excel sheet to make the plot.

3.5 SIMULATION

The simulation is compiled in Excel sheet by using the Visual Basic for Application (VBA). Since the Excel is based on cell definition, the algorithm is written as using the values in the cells. The data is called from a cell and used for calculation then the results are printed in required cells. Figure 3.6 shows an example of simulation page with input details required for one track points. The green shaded cells are the input values entered manually.

Number of time steps	150					Nose Gear angle (deg)	10		Number of track points	1	
						Radians	0.174532925				
Point number	Drive point	Drive point Y	Trace point X	Trace point Y	Drag point1 X	Drag point1 Y	Drag point2 X	Drag point2 Y	Track point 1 X	Track point 1 Y	Degree of Tangent
	0	0	0	56.332	9.375	56.332	-9.375	56.332	58.7106	77.0833	

Figure 3.6 : Input Details for Simulation (User Interface)

Number of time steps represented in Figure 3.6 is coordinate values of predefined path that is divided into points. A path can be divided into several number of points, the higher the point the curve looks smoother. In this research, the number of steps for a straight-curve-straight path will be split into 50 for each segment. An example path representation is shown in Figure 3.7. In this picture the path has a straight-curve-straight formation. Each segment is divided into 50 individual parts and the coordinate values of each points are calculated and given as an input to drive points (NLG). For example, if the straight line has a 100 units length, this 100 units is divided into 50 numbers of small 2 units.

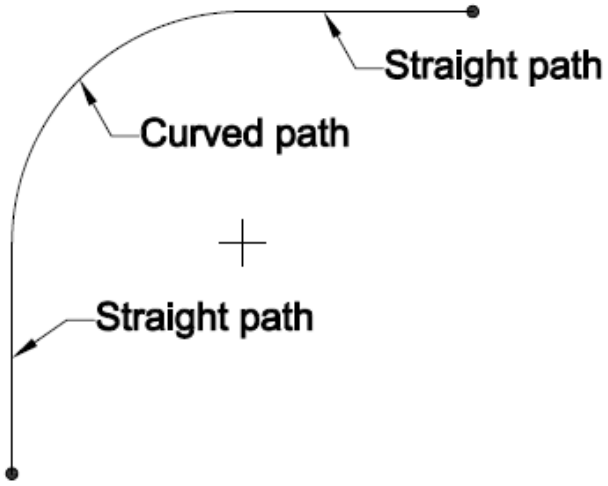


Figure 3.7: Example path

This path is generated in a separate sheet and the coordinate values are pasted in to the simulation sheet. This path generation is done based on the aircraft dimensions. In this thesis, the B-737 is selected as a typical aircraft for simulation and experiment. The path is created based on total length of the aircraft, which means the length of the straight lines to be the total length of the aircraft and the radius of the curve is again the total length of the aircraft. So, the path is defined as 1L-1R-1L where L and R are the total length of the aircraft.

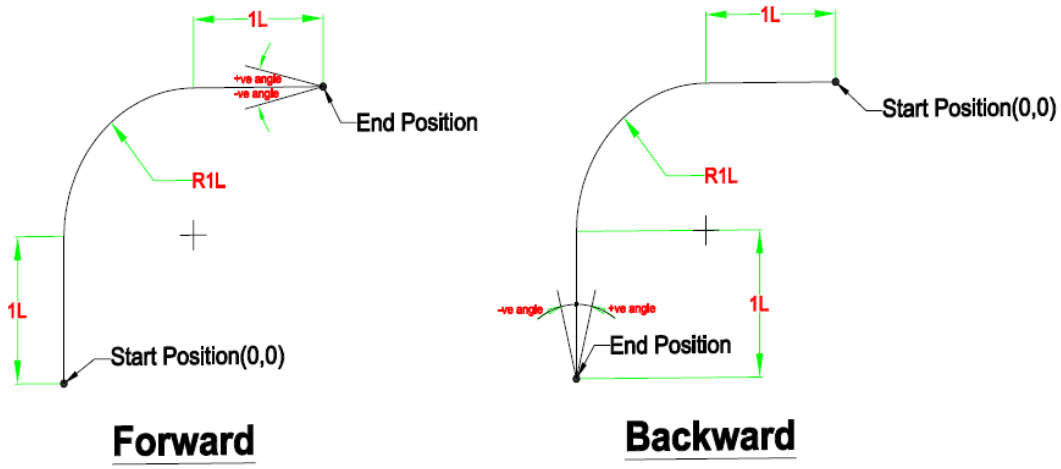


Figure 3.8: NLG Path for Simulation (L-Total Length of Aircraft)

Figure 3.8 shows the forward (pushback) and backward (pull-in) path that is used for simulation with dimensional values along the starting (initial location) and end position (target location). Aircraft is pushed pack from starting position to the end position along the path. The positive and negative angle shown in Figure 3.8 is the aircraft position at end after its pull-in/pushback. This may be applicable for any other aircraft. This simulation is adaptable for any other aircraft to find out the trajectory formation but, necessary changes need to be done for a specific aircraft. The length and initial positions are given in the path creator sheet to create a straight line as first element. Now, the curved path is formed by providing initial points, radius, turn angle and origin. The end point of the first element is serves as a starting point of the second element. In this way different segments are connected together to create a complete path.

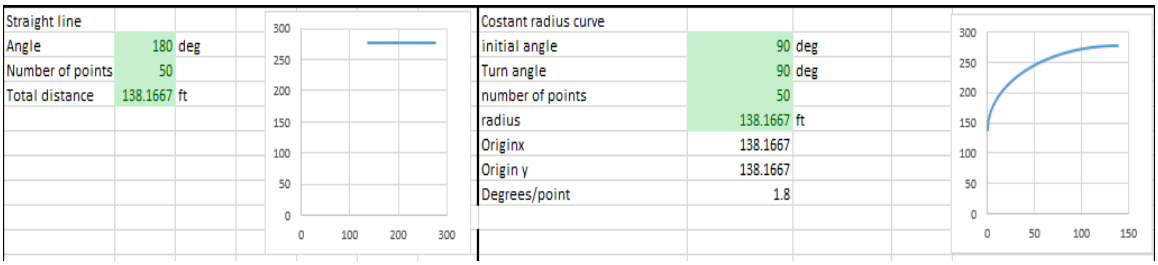


Figure 3.9: Path Creator

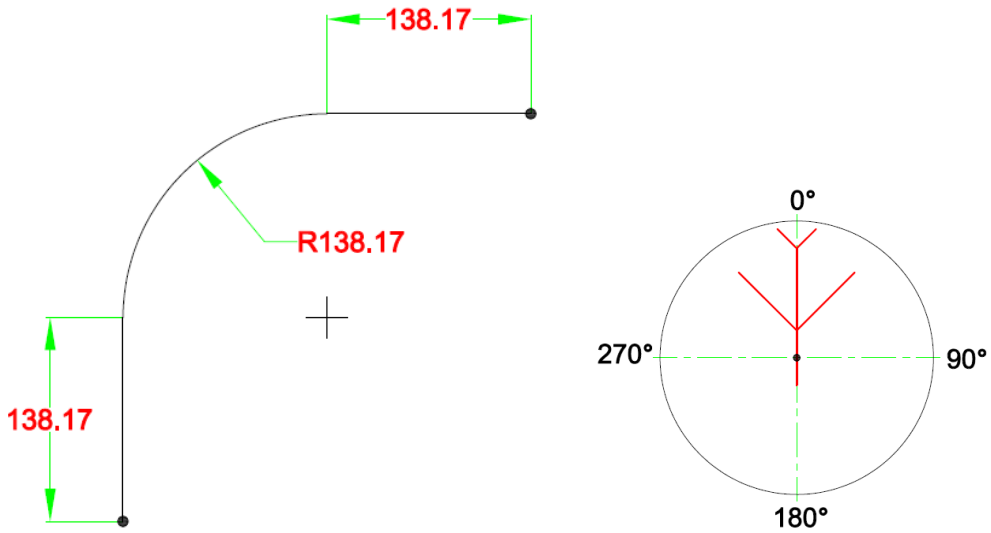


Figure 3.10: Path for B-737 (Units in ft) and Angular representation of Aircraft Start position

Figure 3.9 shows the preview about the inputs needed for path creation. Formulation is written in a way that the graphical forms are developed with respect to the input provided. Once the desired path is generated, the entire coordinate points are copied and pasted in the simulation sheet where it serves as a drive point. There are two options created in the simulation sheet one is format sheet and the another one is run simulation. The format sheet option will clear all unnecessary data that are presented in the sheet and the run simulation option is the one calculates the trajectory of the aircraft. The following figures 3.11-a, 3.11-b and 3.11-c shows the results calculated from the simulation model for different nose angles of B737-900ER with 1L-1R-1L path. Values of X and Y axis in the simulation results represents the distance in all figures.

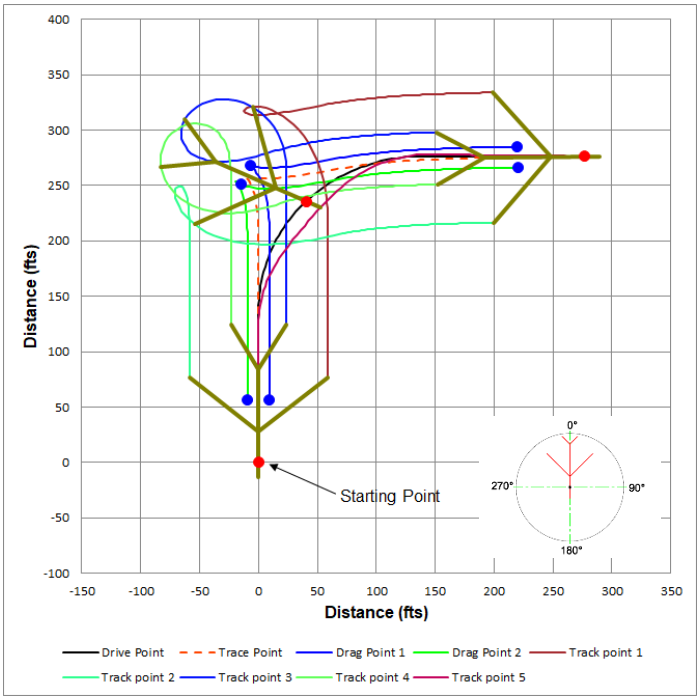


Figure 3.11-a: Starting Nose Angle 0 Degrees

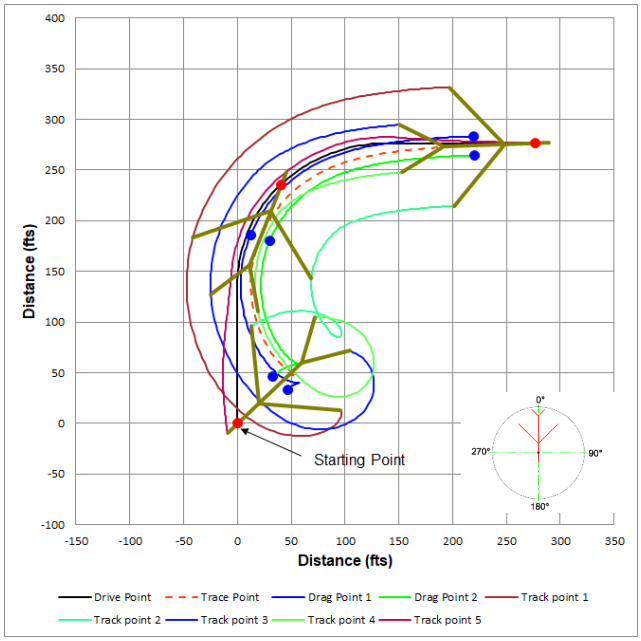


Figure 3.11-b: Starting Nose Angle 45 Degrees

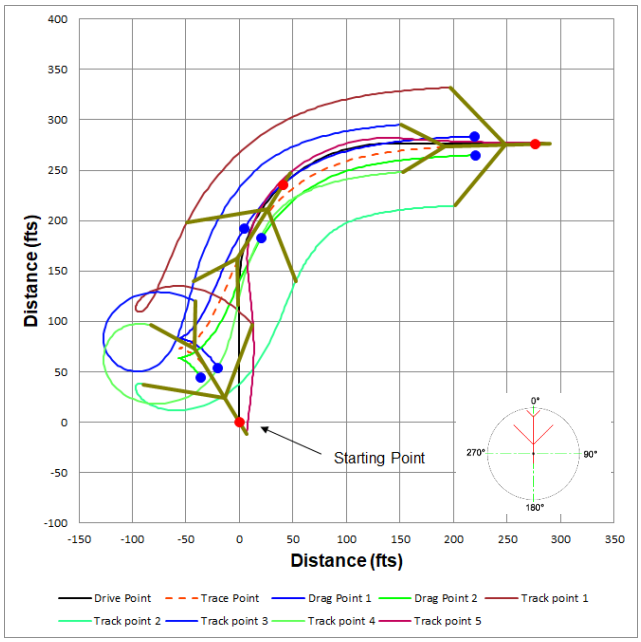


Figure 3.11-c: Starting Nose Angle 330 Degrees

The line in black color shown in above pictures represents the predefined path and red color dot travels along the path is the Nose Landing Gear. The position of the aircraft is shown at three stages,

that is at initial, middle and final stage. Multiple colors are chosen to show the trajectory of different point like drag point, track point-1, trach point-2, etc. Dashed line in the pictures shows trajectory of the trace point. Since it is an imaginary point which is used to calculate the position of the drag points it is shown in dashed line. By comparing these three figures, it is observed that the aircraft is being pushed back up to a certain point based upon its initial orientation and then it is pulled towards the end position. The changeover (pivot) from push back to pull in occurs at different levels for different NLG angles.

Creating aircraft representations in the trajectory simulation is the difficult part. Center point where the two wings meet at the aircraft axis are defined and then a rigid connection between wing tips and intersection point is created. This connection should adjust with the nose gear angle so that the shape of the aircraft will not change with respect to nose gear and while travelling along the path. Similar approach is used for tail tips and this presence of the aircraft on the trajectory helps to understand the movement of the aircraft while pushing back or pulling in. The first part of the research objectives is achieved at this stage.

3.6 EXPERIMENT AND VALIDATION

Although this thesis mainly focuses on simulation and analysis of push back trajectories, an experiment is conducted to provide confidence in the results obtained from simulation. This experiment is conducted by making a scaled physical model of Boeing 737 and the predefined path is drawn on a foam board. The model of the aircraft is made to hold the marker in order to draw the wings and tail tips, when the nose gear moves along the predefine path. The trajectories obtained from the experiments are accessed with the analytical model and the conclusions are made upon the ability of the simulation program to provide useful trajectories and swept area estimates.

3.7 EXPERIMENT

3.7.1 AIRPLANE MODEL

This scaled model is made up of 80/20™ aluminum frame with adjustable brackets and it is 1/100 scale relative to Boeing 737-900 ER. This experimental aircraft consists of Nose Landing Gear, Main gear and wings. Since it is adjustable, we can modify the distance between nose and main gear, as well as the lateral dimensions of the main gear according to the actual dimensions. Nose gear is made up of a bolt with cap nut which has a round surface at its bottom and it is shown in Figure 3.12. This round section helps the airplane to slide smoothly on the foam board when the airplane is pulled in or pushed back along the predefined path. The Main gear consists of two wheels equally assembled on both sides of the airplane axis. These wheels are having bearings at inner for smooth rolling action and these rolling wheels are providing the friction and slippage for real effects. A 3D printed bracket is modelled and printed to assemble the wheels into the frame to ensure the rigid connection between tracking points. In order to define the tail and wing tips, one more model is made in solid works and 3D printed to hold the marker for track points. This 3D printed component has a slot that can be fitted to the frame with the help of bolts and nuts. There is a hollow cylinder to hold the marked which helps to draw the trajectory of track points.

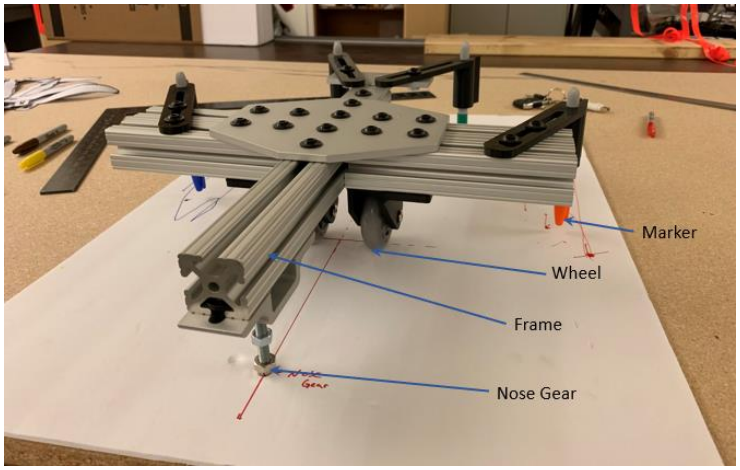


Figure 3.12: Scaled Model of B737

The marker is mounted on the cylinder and its position adjusted by moving the bracket on its slot on the frame. This assembly is tightened once the marker adjusted to its desired position. There are totally four markers assembled in the frame, two represents the wing tips and another two represents the tail tips. Markers with different colors are used to identify the tracking point numbers in order to compare them with the simulation.

3.7.2 FOAM BOARD

Four foam boards are used to draw the predefined path and track points. These foam boards are attached together by a tape on a big wooden flat board. Since it is a light weight material and can be cut easily, a cut out of B737 is made and attached to the physical model. Figure 3.13 shows the arrangement of four foam boards and a cut out of Boeing 737 is placed on the physical model.

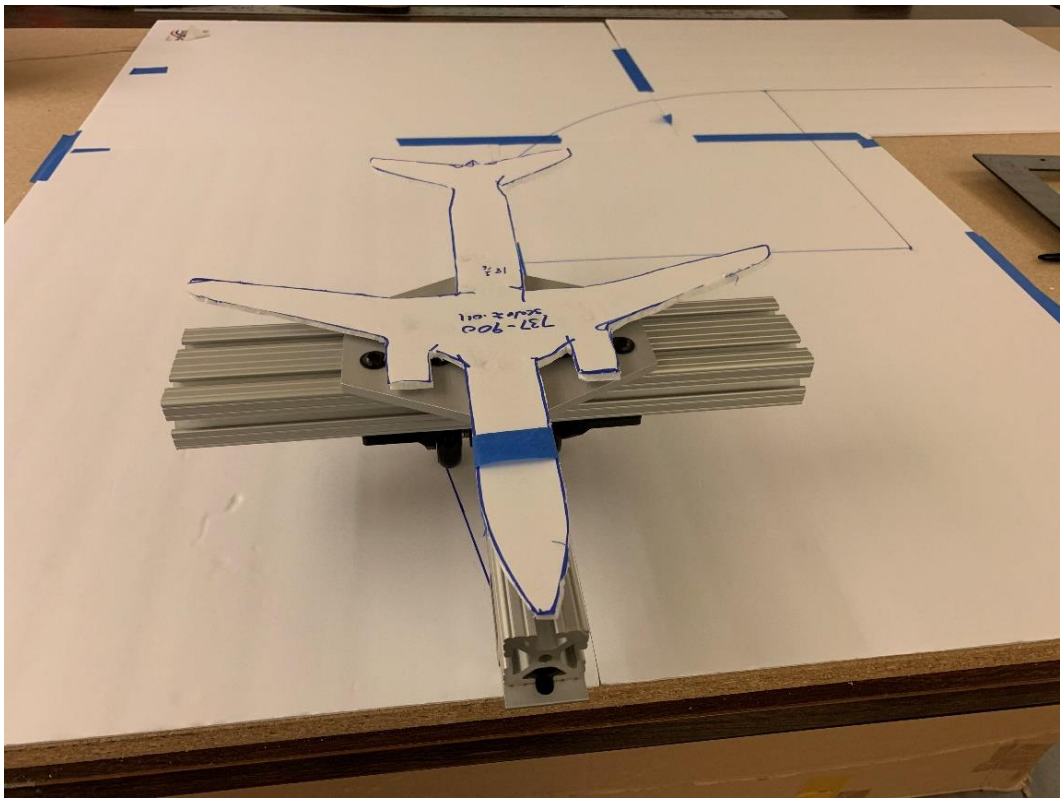


Figure 3.13: Arrangement of Foam Board with Boeing 737 cut out

3.7.3 EXPERIMENTAL PROCEDURE

This experiment is done by moving the nose gear along the desired path while the main gear and extreme points are tracked. After setting up the foam board, a scaled path is drawn on the foam board which serves as a base for constructing the layout. Since it a process for validation, the path (1L-1R-1L) which is considered in simulation is drawn by considering scale factor along with initial and end positions. Now, the positions of the markers are measured and noted at the starting point to conduct the experiment, then the airplane model is placed on this position. The airplane is moved along the predefined path after checking the initial angle. The traces of the track points are drawn by the markers as the nose gear moves along the defined path. These paths formed by the track points are measured in the interval of 6 inches in the predefined path and these values are compared with the simulation results. Since the tires are rolling, the drag points are not formed and track point-5 is not created in the experiments as well. The trace point is an imaginary point it is not possible to draw its path during this, Figure 3.14 show the results obtained in experiment.

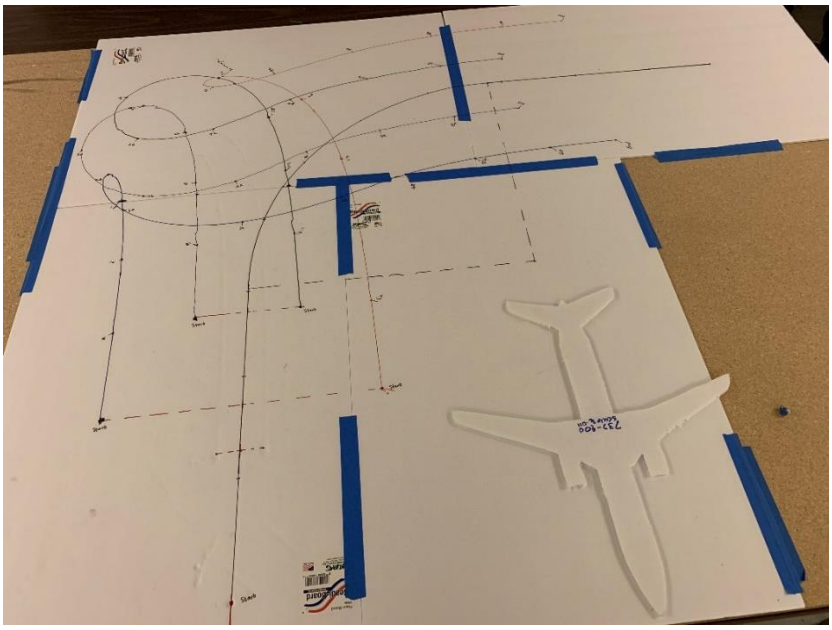


Figure 3.14: Experimental Track Results

3.7.4 COMPARING THE RESULTS

To check the credibility of the simulation, the results from the simulation and experiment are compared and analyzed. The values of each points are measured with the reference as (0,0) which is the starting point of the nose gear. Values at 6-inch interval in the predefined path of the nose gear is marked and corresponding locations are noted in the track points. These corresponding locations are measured from the reference point. Simulation results are extracted at the same points measured during experiments. The measured and extracted values are tabulated and Table is shown in appendix A. Graphical representation is the best way to compare these results, so the results of both analytical and experimental are drawn in graph. The data points used to generate the curves in simulation is same as used in the experiment. So, this simulation curve for comparison may not represent the complete path. The experiment is conducted for push back in 1L-1R-1L path with 0° nose angle. These tabulated values are drawn as graph and shown for track points 1,2,3 and 4 in figures 3.15 a, b, c and d respectively.

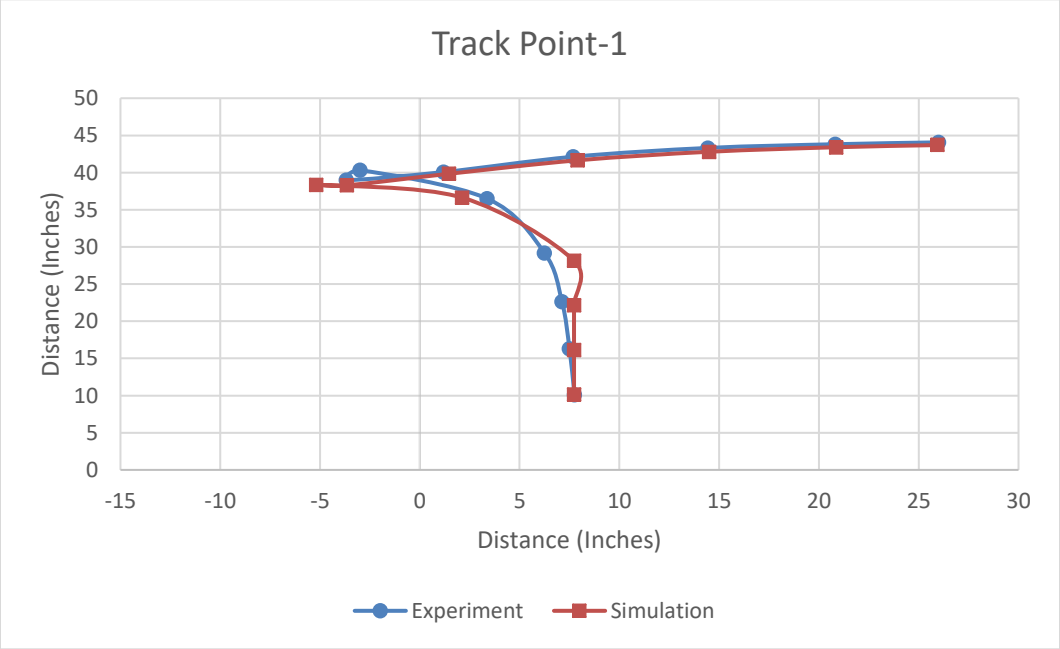


Figure 3.15-a: Experiment vs Simulation for Track Point-1 (Left Wing Tip)

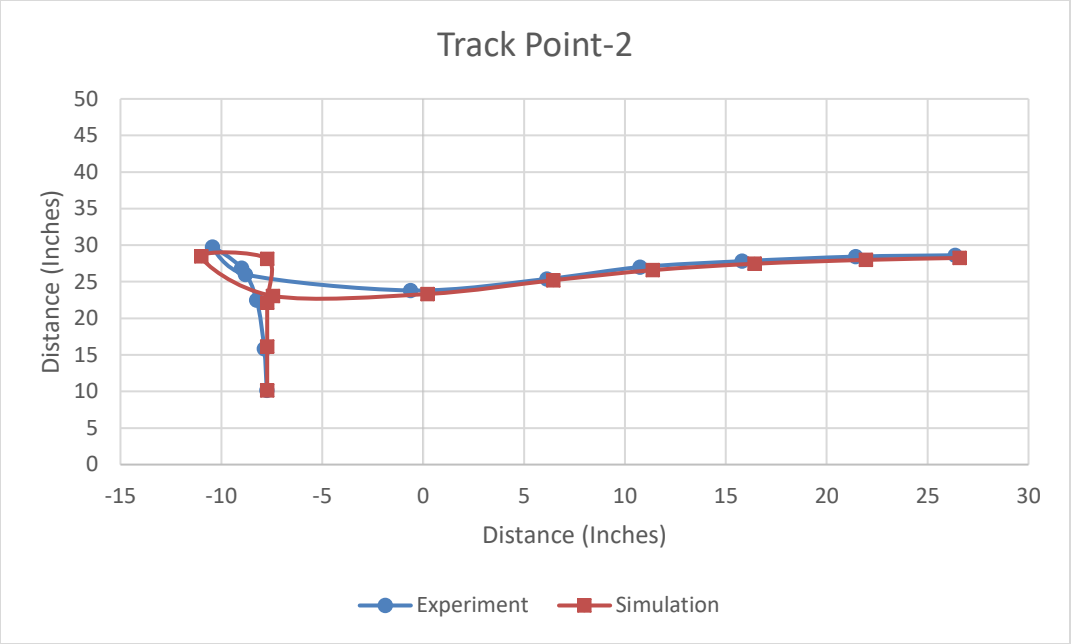


Figure 3.15-b: Experiment vs Simulation for Track Point-2 (Right Wing Tip)

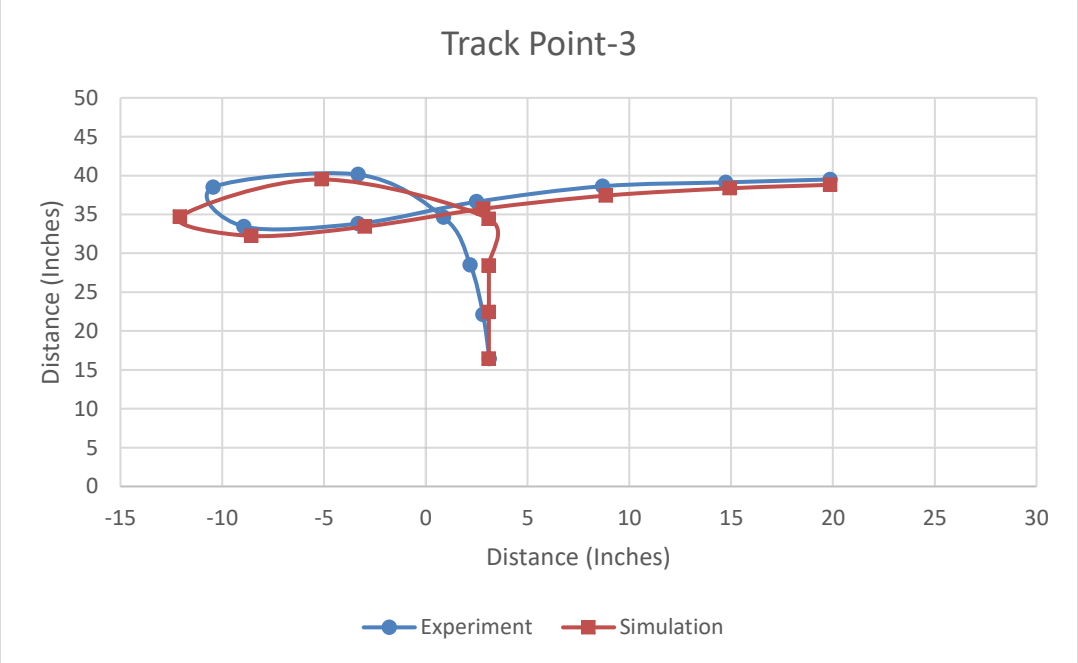


Figure 3.15-c: Experiment vs Simulation for Track Point-3 (Left Tail Tip)

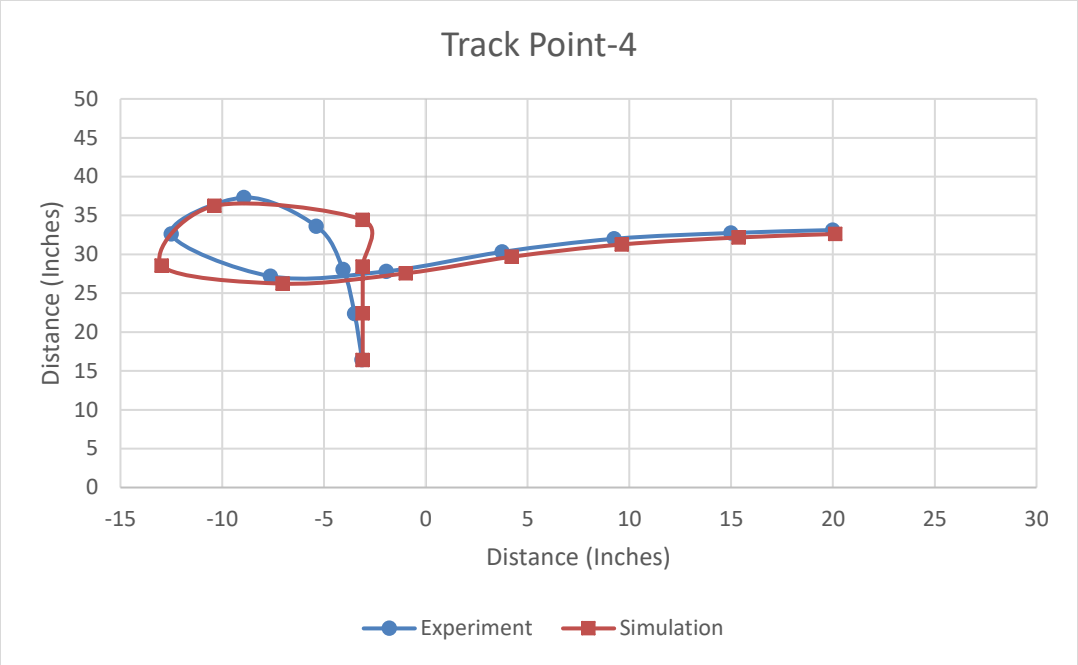


Figure 3.15-d: Experiment vs Simulation for Track Point-4 (Right Tail Tip)

It is clearly understandable from the above figures that the simulation program is closely matches with the experiment results up to some extent. However, the small variations are occurred due to the measurement uncertainties, physical effects of the main gear wheels on the foam board and the movement of airplane along the predefined path. These measurement uncertainties include the position of the markers, wheels, initial alignment of airplane with the marked location. For example, the distance between the NLG and Track Point-1 along horizontal axis in simulation is 7.73 inches, whereas in experiment is 7.75±0.25 inches. Rulers and squares are used to measure the distance between each point with reference to the starting position. Moreover, the rolling effects made by the wheel on the foam board is not considered in the simulation, this may cause the change in results obtained from experiment. The wheels in the main gear makes an impression on the foam board due to the weight of the airplane model and it slips over the connecting tape on the foam board. Since the airplane is moved manually the travel is not smooth throughout the path. The airplane is stopped for every 6-inch advancement to mark points for future measurement. After

analyzing all these factors, additional experiment could reduce these variations and may provide the results which is closer to the simulation. However, the trajectories are very similar in simulation and experiment and thus the simulation is esteemed as adequate for push back assessment.

3.7.5 QUANTIFICATION OF MESUREMENT DEVIATION

A ubiquitous feature of experimental data is measurement error, which establishes the difficulty in interpreting the accuracy of predicted values. In many cases the primary purpose of the experiment is to determine the quantitative model to compare it with the analytical (Predicted) values and find out how well the simulation obeyed. The need for this regression analysis of uncertainty with the model parameters is to find out the confidence limit of the simulation. One of the methods widely used to find out the differences between values predicted by a model and the values observed during experiment is Root Mean Square Error (RMSE) [41]. This helps to find out the deviation of measured values from the centered values of model. The following equation is used to calculate the RSME of track points measured during experiment.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(P_{,i}-O_i)^2}{N}} \tag{3.8}$$

Where, i is the sequence of values, N is the number of values, P is the simulation (Predicted) value and O is the experimental (Observed) value. The RSME value is calculated by using all the simulation values correspond to the experiment values for each track points in X and Y coordinates and the values are represented in Table 3.3.

Track Point	Coordinate	RSME (in)
1	X	0.87
	Y	0.74
2	X	0.71
	Y	1.04
3	X	1.01
	Y	1.29
4	X	0.93
	Y	1.32

Table 3.3 : RSME values of Track Points

The values of RSME is the deviation from the mean error calculated for simulation and experiment values. This ranges from 0.71 (Track point-2,X) to 1.32 (Track point-4,Y) for a data ranging from -12.97 to 44.06 (Refer Table A1 and A2 from appendices). As in general the lower the RSME value the more accurate the model. It can be concluded that the RSME value is small for the data considered in this thesis, which implies the accuracy of the simulation model is reasonable. To check the validity of the model in another perspective, the offset angle at the end of the experiment is found and compared it with the simulation. The value of offset angle in simulation is -2.24° and in experiment is -1.41°. The difference in angle is due to the variations in experimental procedure and simulation. More over these values are scaled as 1:100 and the errors may be reduced further when the experiment conducted in its own dimensions. Additional experiments are recommended with more control on path and tracking of extreme points if the simulation is to be used for more than just an initial feasibility assessment.

3.8 VERIFICATION WITH AIRPLANE CHARACTERISTICS MANUAL

Simulation results from the mathematical model is compared with the MLG tire tacks provided in the airplane characteristics for airport planning by Boeing [18] for B737-900 to provide another reference. Selected points from the nose landing gear path and main gear tracks measured from the airplane characteristics manual [18] and curve fitting is performed to create smooth interconnecting lines. The details from the manual is imported in AutoCAD ® and Solidworks ® to measure the

dimension and parameters of NLG and MLG paths. Details obtained for NLG path from manual is given as an input in the simulation. The image from the OEM manual is imported in to AutoCAD and the dimensions are measured. These measured dimensions are compared with the dimensional details provided in manual and scale factor is calculated. Values are measured at different intervals in the airplane characteristics manual to make a smooth curve during graphical representation. The same number of intervals are considered when extracting the values from simulation. Similar, to experimental validation these results are represented in the graphical form. To do this, the measured values and simulated values are plotted in Excel™ for a common horizontal axis. As an example, set of values are shown in Table 3.3. There are two NLG paths represented in the manual, one is for 90° turn and another is more than 90° turn. Figure 3.16 shows the airplane manual represents the 90° turning.

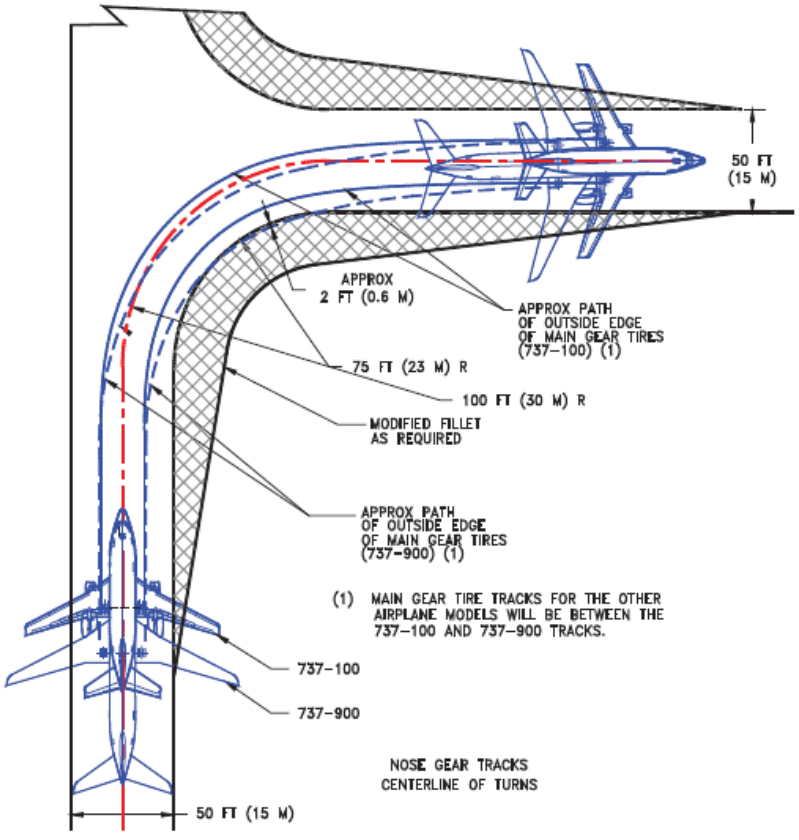


Figure 3.16: Runway and taxiway turn paths 90°

Drag Point 1 (left), fts		
X	Ysim	Yoem
-11	-56.332	-50.1043
0	100.339	100.386
50	161.769	162.69
100	180.455	180.658
150	187.956	188.054
200	191.142	191.417

Table 3.4: Coordinate values of simulation and Aircraft manual

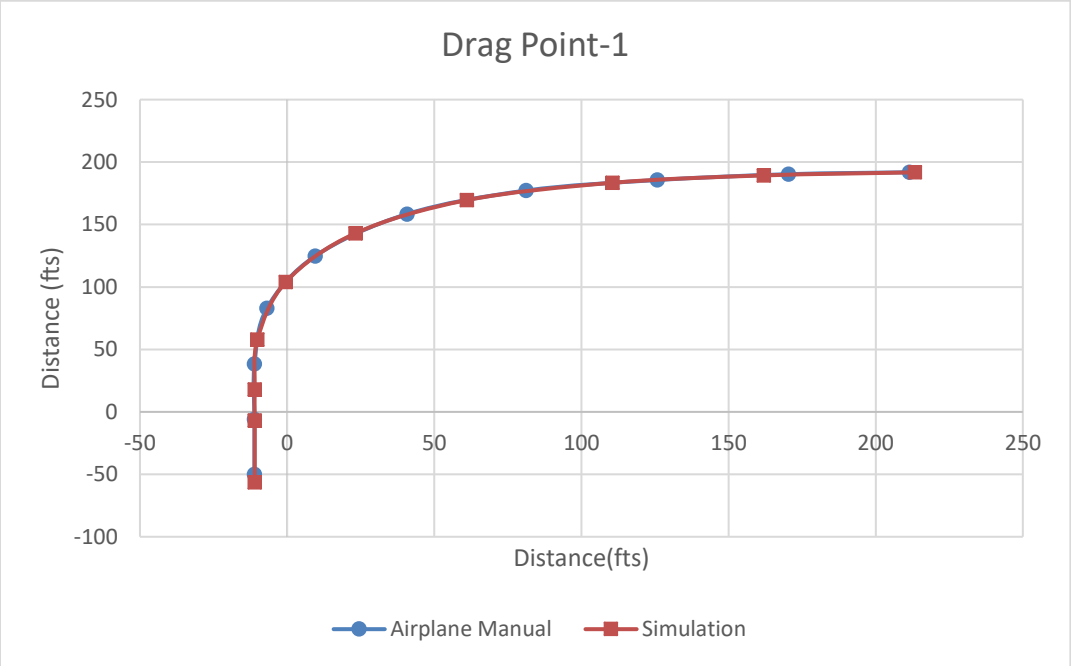


Figure 3.17: Airplane manual vs Simulation for Drag Point-1 (for 90° turn)

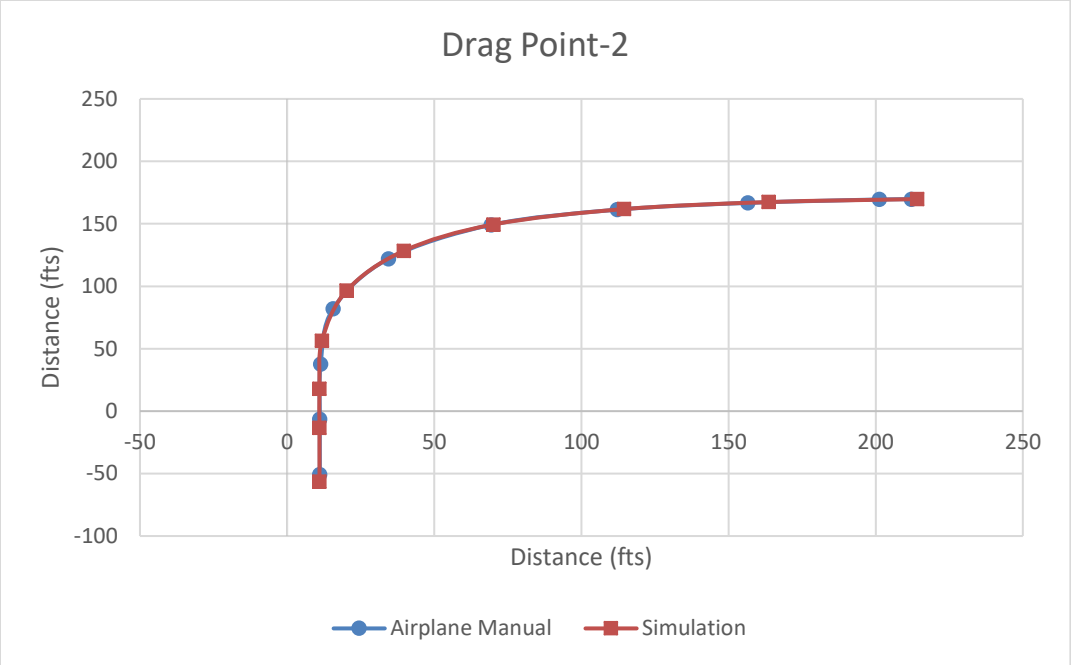


Figure 3.18: Airplane manual vs Simulation for Drag Point-2 (for 90° turn)

From Figures 3.17 and 3.18 it is observed that the simulation results are aligned with the MLG track in the airplane manual. This analytical model developed to find out the trajectories of MLG is adequate to analyze the feasibility of aircraft. Small variations in the results between the simulation and airplane manual occurs due to the curve fitting and approximation during measurement.

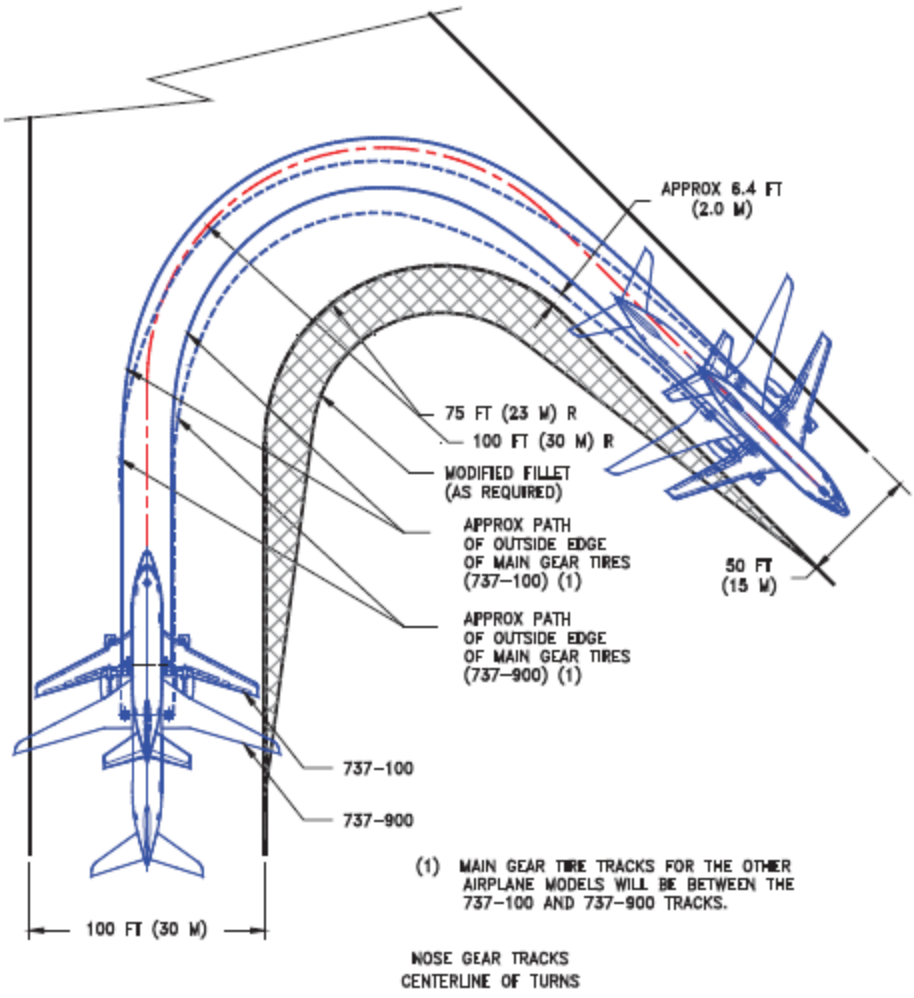


Figure 3.19: Runway and taxiway turn paths more than 90°

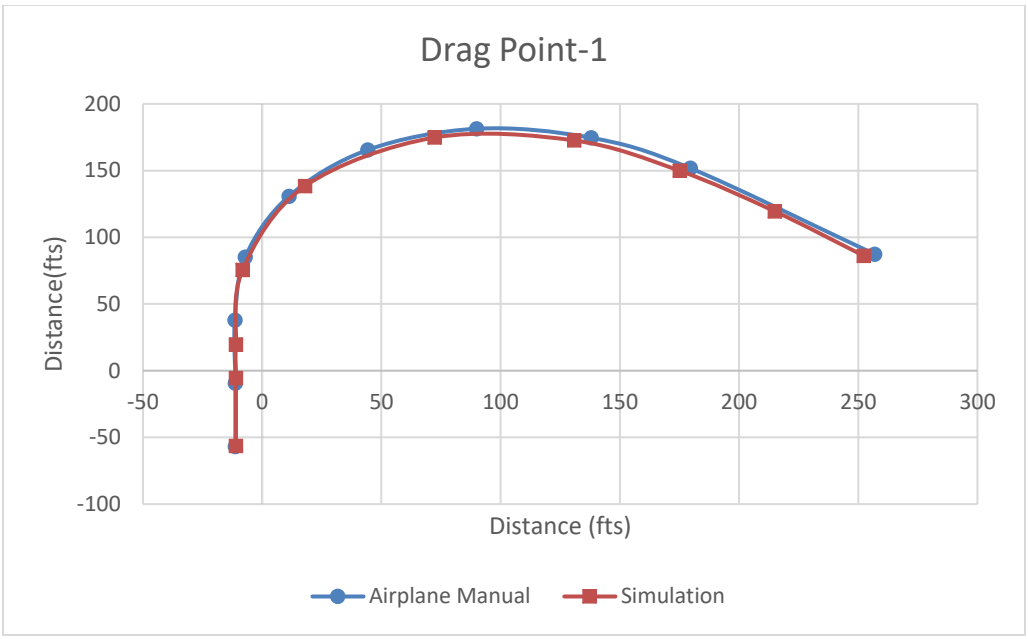


Figure 3.20: Airplane manual vs Simulation for Drag Point-1 (for more than 90° turn)

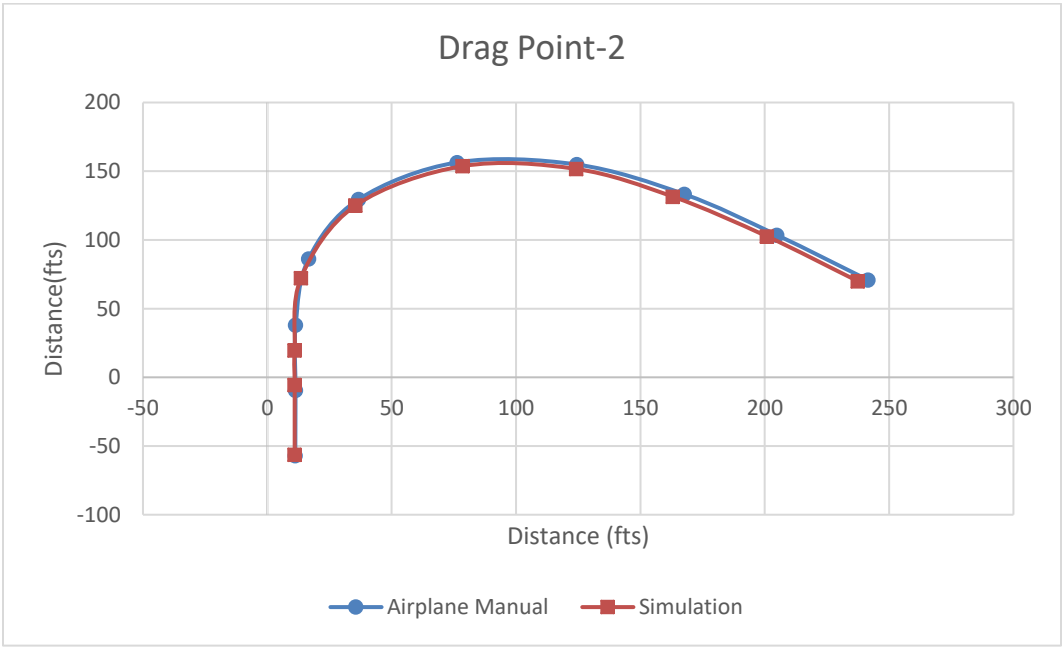


Figure 3.21: Airplane manual vs Simulation for Drag Point-2 (for more than 90° turn)

The comparison results for more than 90° turn is represented in figures 3.20 and 3.21. The variation in the results between simulation and airplane manual is higher than the variation observed in 90° turn. This is due the approximation and measurement deviation in curve fitting.

CHAPTER IV

FEASIBILITY ASSESMENT AND PATH IMPROVEMENT

The feasibility of fixed track pull-in and push-back operations in a typical airport environment is evaluated using the simulation developed in the previous chapter. In this context, feasibility will be shown if there is found one or more fixed track paths that can enable movement of the example aircraft into and out of a parking location in a realistically constrained area.

4.1 APPLICATION OF TRAJECTORY AND SWEPT AREA

A key output of the simulation is the area swept out by the aircraft during its movement. This predicted swept area shape and size will help in designing the track for a given airport layout. Figure 4.1 shows a terminal area based on typical single-aisle airliner operations using dimensions from the Boeing 737 aircraft. This is a generic terminal layout to represent the KOZ's, terminal building and taxiways. Currently, ground personnel will be stationed near the wing tips and nose of the airplane and guide the pilot or tug driver to park and to push back. These personnel will watch out for the obstacles and signal the driver or pilot to move the aircraft safely. In the case of a fixed track system, the swept area must be predictable for all permitted aircraft operations to avoid collisions. Using the simulation software, various track trajectories can be evaluated and the swept area may be found. This area is not necessarily formed by the farthest point from the nose gear but rather the wing tips form the outer boundary when the aircraft travels in a straight path. The tail tips sweep over the widest area during turns. Figure 4.2 shows the example for a given trajectory and resulting swept area.

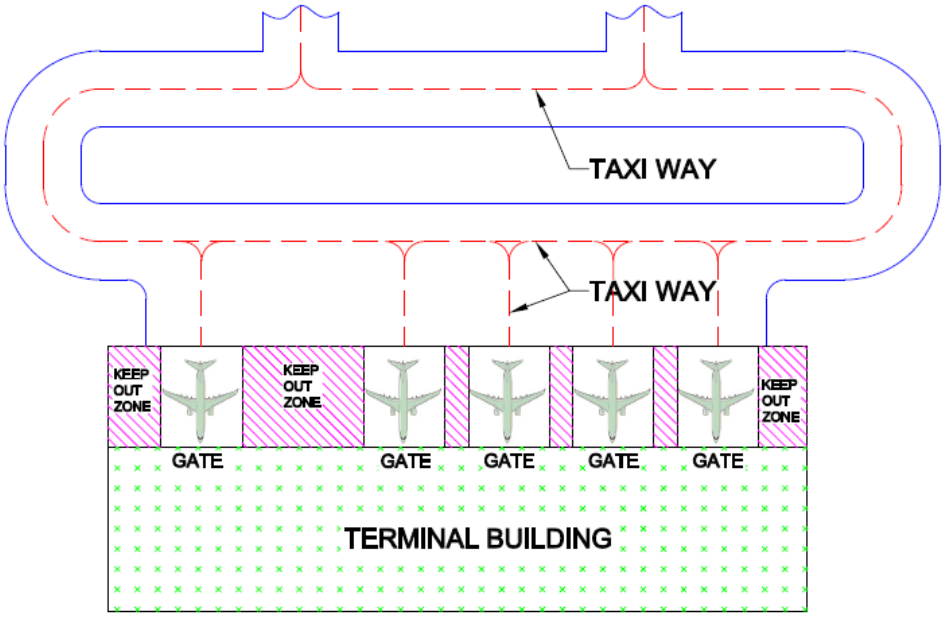


Figure 4.1: Generic Terminal Layout based on Boeing 737 aircraft

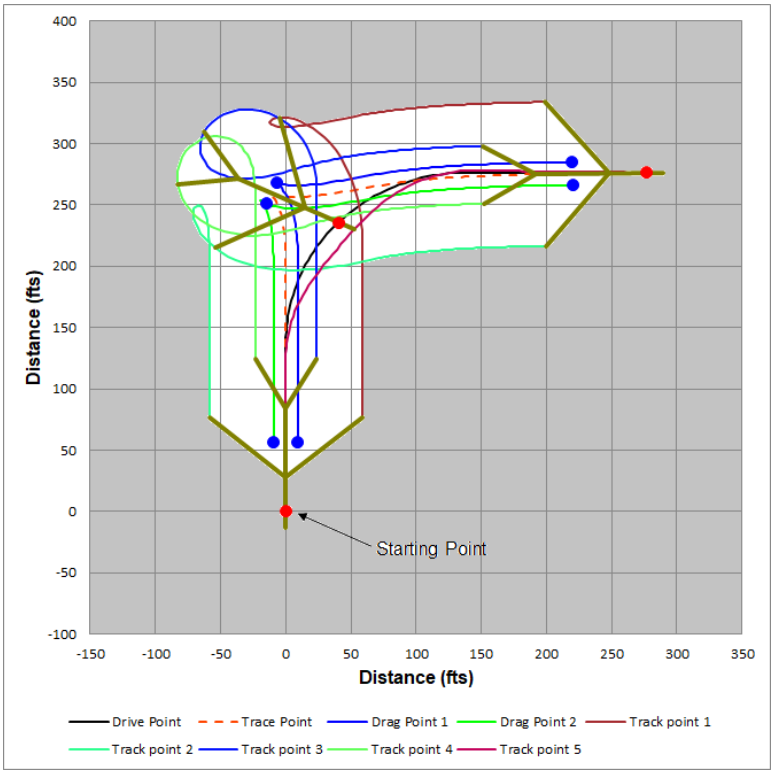


Figure 4.2: Example Swept Area

4.2 ANALYSIS OF PULL-IN AND PUSHBACK

During the taxi after landing to the terminal gate parking area, a typical path would be an approach to the gate by a straight path parallel to the terminal building with a 90° turn into the parking location for a fixed track path with dimensions 1L-1R-1L, and an arrival as shown in Figure 4.3. In this path the numerical value represents the scale factor of total length of the aircraft and L,R represents the total length of the aircraft. Since the aircraft are placed parallel, it is decided to turn the aircraft after moving a distance equal to the tail end of adjacent aircraft. The aircraft will arrive at parked position with a 2.24° offset from its path. For this example, the aircraft is assumed to be exactly aligned with the path at the start of pull in. From the parked position with the offset angle of 2.24° a push-back simulation is shown in Figure 4.4. Note that the aircraft pivots inside the curve during this operation. This kind of behavior requires that initial orientation ranges be assessed along with fixed track path variations.

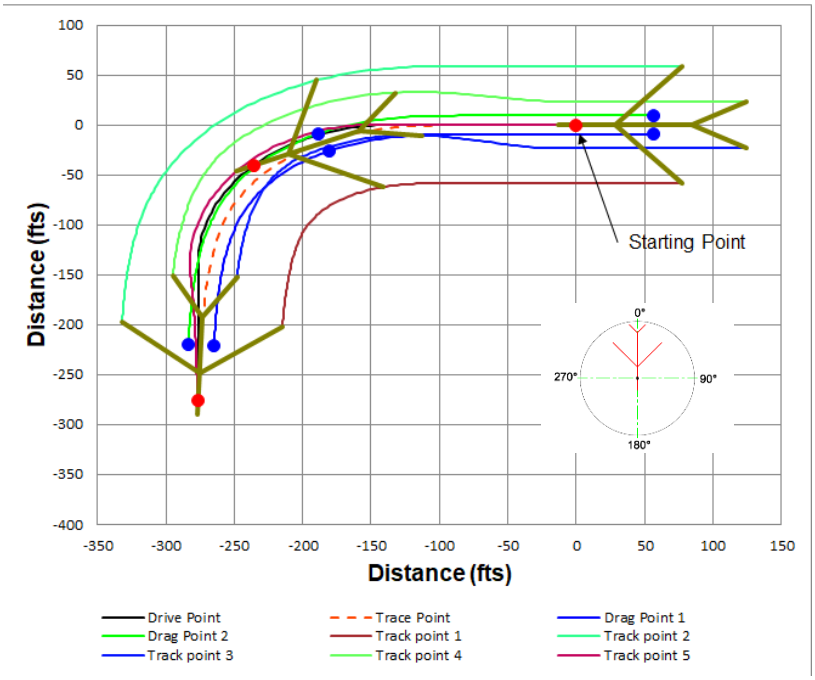


Figure 4.3: Pull-in path from 0° initial offset

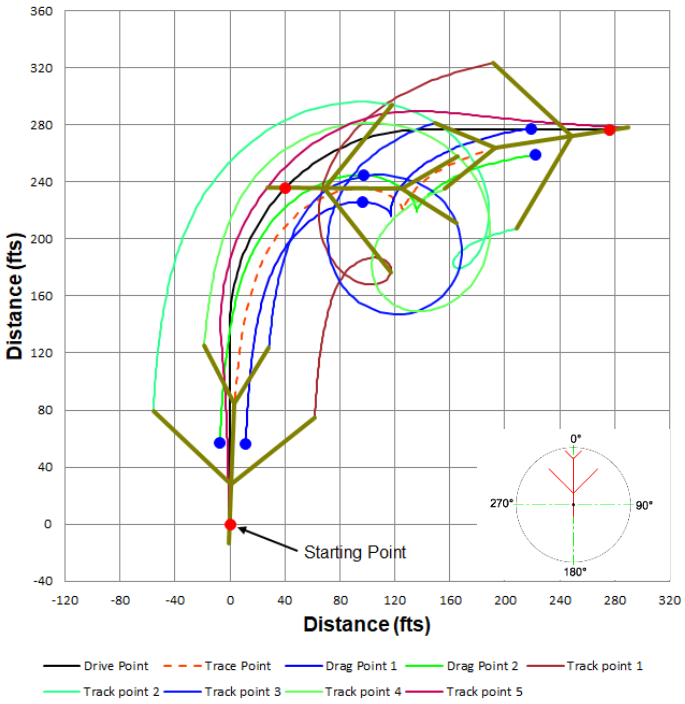


Figure 4.4: Pushback path from 2.24° initial offset

It is observed from Figure 4.4 that the aircraft arrives at the end of the push-back operation with an angle of -8.32° relative to the path. Since the dolly is aligned with the path, the ability of the aircraft to safety dismount from the dolly must be evaluated. Analyses should be made on the offset angle with different aircraft for the planned dolly dimensions to find out the allowable offset angle range. This will be discussed later in this chapter. In the simulation, the aircraft can be placed at small offset angles at the initial position. However, in real operations, it will be difficult to align the aircraft exactly with the path. Any fixed track system will need to incorporate aircraft orientation sensing and pilot cueing to minimize offset angles and to provide safe mounting onto the movement apparatus (i.e., the dolly or equivalent). By considering this fact, the aircraft trajectories and orientations are evaluated with -3° to +3° as initial offsets at the starting position. The following figures show the trajectory formation.

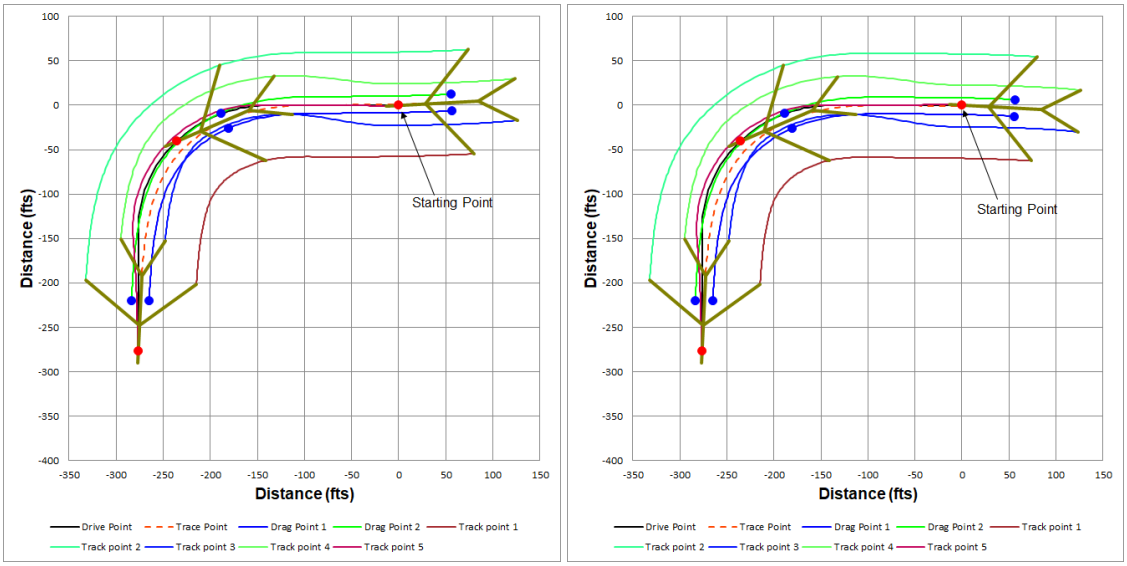


Figure 4.5: Pull-in path (Backward) from initial -3° (left) and +3° right

When the aircraft is pulled in within ±3° initial offset angle the aircraft forms at approximately 2.24° offset at the end point. The airport terminals are designed in a way that a gate has capable of serving variety of aircraft sizes and types. Since the B737-900 is used as the typical aircraft, a larger and a smaller aircraft are evaluated using the example path. The ERJ-135 regional jet is selected as the smallest model among the aircraft surveyed. The B777-300ER is selected as the larger aircraft as it is conceivable that a terminal serving single aisle jets may also serve dual aisle. The baseline track is sized based on B737-900 ER which is 137.18 ft long and 117.417 ft wings span [18]. The ERJ-135 is 86.417 ft long with 69.75 ft wing span [20]. The B777-300ER has a total length of 242.33 ft and wing span of 212.583 ft [21]. Figures 4.6 and 4.7 represent the trajectory of B777 and ERJ-135 in a B737 track respectively.

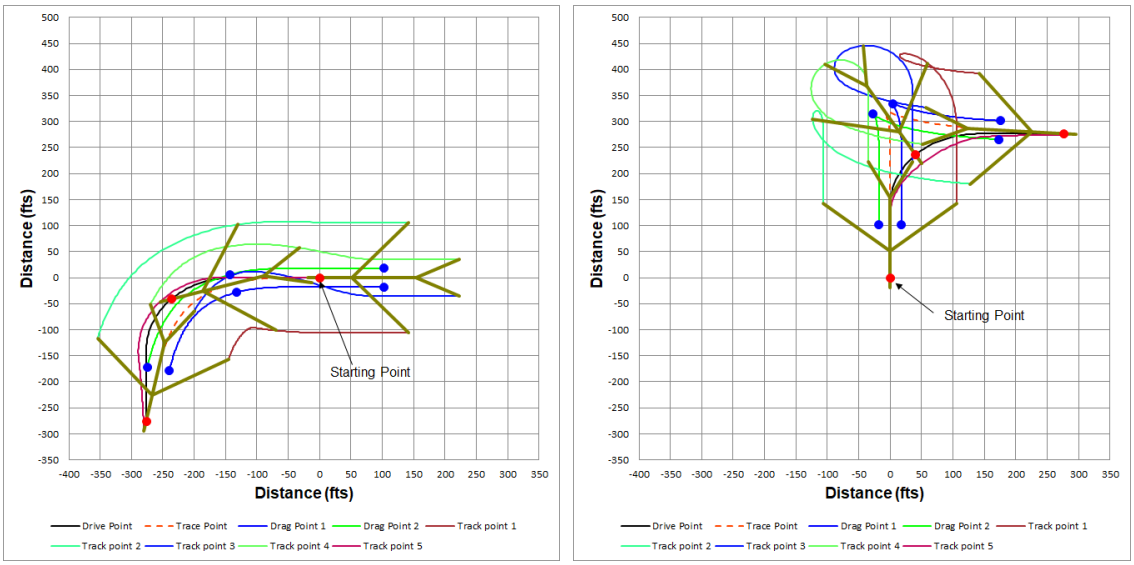


Figure 4.6: Boeing 777 Pull in (left) and Push back (right) along the B737 based track

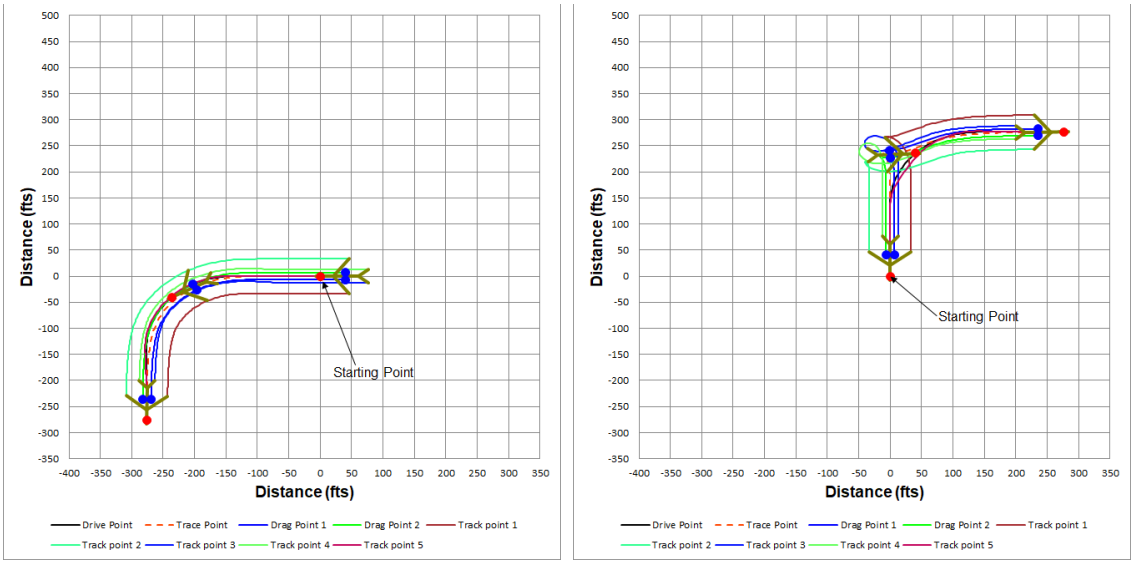


Figure 4.7: ERJ-135 Pull in (left) and Push back (right) in B737 track

The larger and smaller aircrafts are behaving similar in terms of pivoting during pull-in and pushback operation. But, the offset angles generated at the end is higher in larger aircraft than the smaller one. The path is not feasible for the larger aircraft because the aircraft needs more space to be at desired position at end. Moreover, it can intrude into the safe zone of adjacent aircraft. The

smaller airplane is well behaved with the path. The offset angles at end position during pull-in and pushback seems to be well within the limits. This path is big enough to perform these operations without disturbing the KOZ's.

4.3 OFFSET ANGLE ANALYSIS WITH DOLLY

As mentioned earlier, the offset angle found during push back should be within the accessible limit of the dolly width. If the offset angle is too large, the aircraft may not be able to travel down the ramp while detaching from the dolly. If the dolly is designed to accommodate several types and sizes of aircraft, it is important to check the feasibility of unloading at the extremes of size and relative angles. Tire dimensions and distance between tires are collected from the OEM catalogue of all considered aircraft. The dolly dimensions used are based on the A380 nose landing gear size with a small margin on either side. The A380 based dolly interior width is then 68.0 inches and the distance to the ramp corner is set at 35.25 inches to allow approximately +/- 4 degrees for this size nose gear. Given these dimensions, the remaining example aircraft have the allowable dolly mounting and dismounting relative angles as shown in Table 4.1.

Aircraft Nose Gear wheel Dimensions (inches)							Dolly Dimensions for anlge measurement		
Aircraft	Tire size	Dist. b/w wheel center	Diameter	Width	NG thickness	NG thick/2	dx	dy	Allowed offset angle (one side) (Degrees)
ERJ 135	19.5x6.75-8	13.38	19.5	6.75	20.13	10.065	68	35.25	20.33
CRJ 200	18x4.4-12	11.5	18	4.5	16	8			21.84
CRJ 700	20.5x6.75-10	12.2	20.5	6.75	18.95	9.475			20.76
B 737-900ER	27x7.7-15	16	27	7.7	23.7	11.85			18.99
B 747-8	50x20-R22	36	50	20	56	28			6.09
B 757-300	31x13-12	24	31	13	37	18.5			13.84
B 767-400ER	37x14-15	25	37	14	39	19.5			13.05
B 777-300ER	43x17.5-R17	30.8	43	17.5	48.3	24.15			9.28
B 787-10	40x16-R16	29	40	16	45	22.5			10.62
A300-C4	40x14-20	24.6	40	14	38.6	19.3			13.21
A320-200	30X8.8-15	19.68	30	8.8	28.48	14.24			17.17
A340-600	45x180R17	32.04	45	18	50.04	25.02			8.56
A380-800	50x20-R22	41.34	50	20	61.34	30.67			3.86

Table 4.1: Aircraft wheel dimensions[18] [20] [21] [30…39] and allowable offset angles for dolly

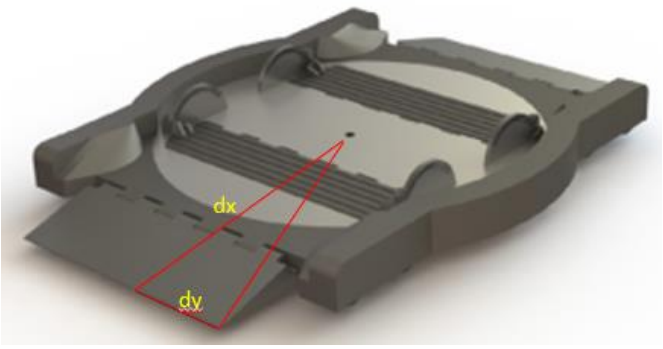


Figure 4.8: Dolly dimensions [2]

From Table 4.1, most of aircraft are allowed to engage and disengage to dolly with more than 8 degrees as offset angle. However, two aircraft (B747 and A380) have lower offset angles such as 6.0 and 3.8 degrees. To achieve these tight angles, a highly capable navigation and guidance system will likely be required to position the aircraft at desired location All these angles are calculated by considering the center axis of the Nose Landing Gear is in align with the center axis of the dolly. These angles may vary if there is an offset in the alignment.

4.4 FEASIBLE PATH IDENTIFICATION

At this point the key constraints on feasibility have been developed (KOZ's, swept areas, initial and final orientation limits). Various trajectories and initial orientation angles are now evaluated to determine if there are any feasible paths. First the aircraft is pulled in to the gate from the taxiway and then it is pushed back from the gate to taxiway. The aircraft is completely attached with the dolly during this operation. According to reference [22], there is a minimum required separation distance between two aircraft for safe parking. This separation distance is tabulated based on the aircraft wing span, total width of Main gear wheel, wheel base, etc. The pull-in is shown in Figure 4.9 with 1L-1R-1L path and the end position is in between two aircraft. From this figure, it can be interpreted that this path is feasible to pull in the B737 aircraft which is placed in line with the path at starting position. The offset angle at the end of pull-in is 2.24°.

Push back operation is performed from this position and orientation of aircraft, in the same path. The feasibility of this path is analyzed by make sure that the swept area remains outside the KOZs of adjacent aircraft. Figure 4.10 shows that the swept areas remain clear adjacent KOZ's. The offset angle at the end of the push back is found as -8.37° which is within the allowable offset angle of dolly for the B737. The 1L-1R-1L path is feasible for both pull-in and pushback operations for B-737-900 ER aircraft with initial pull in alignment at 0 degrees to the dolly at mounting. In real operations, there will be slippage and traction on the tires and other environmental effects. Therefore, an examination of a range of initial angles will help assess the sensitivity to small changes.

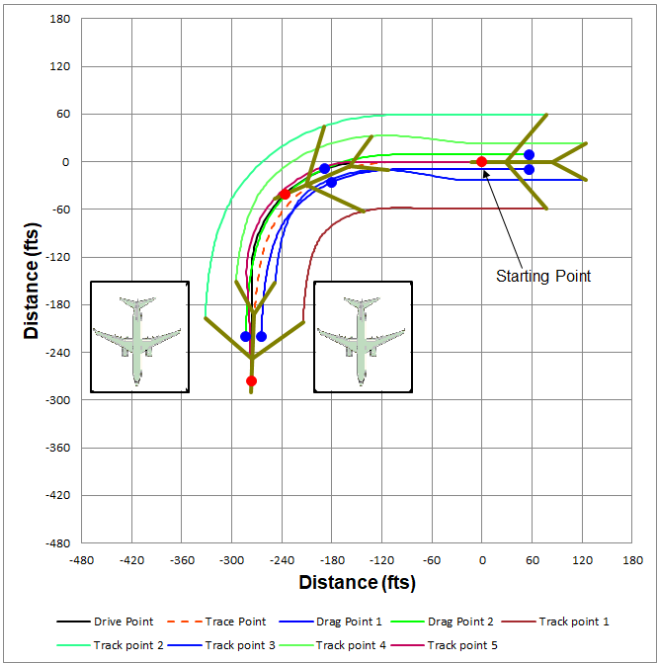


Figure 4.9: Pull-in between two Aircraft with 90° at initial position

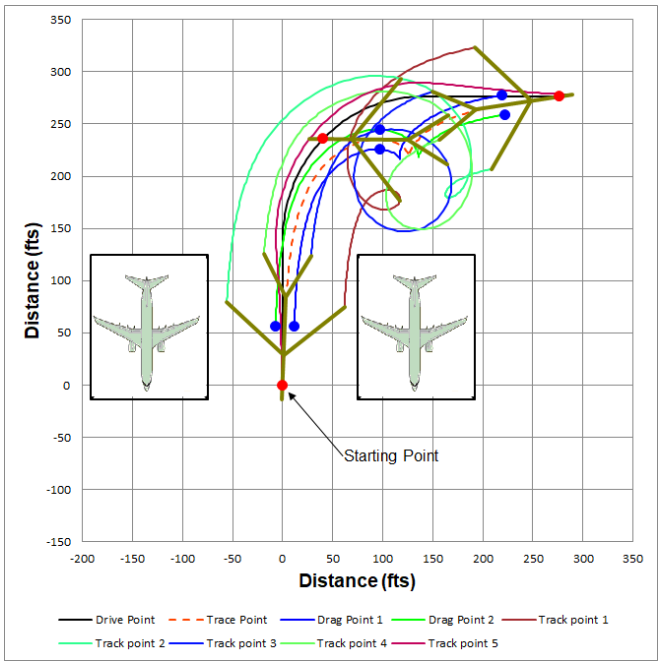


Figure 4.10: Push-back between two Aircraft with 2.24° initial offset

The offset angle at the arrival does not change (negligible difference) when the aircraft is pulled in with ±3° variation from 90° at the starting position of the path.

Incoming aircraft may often arrive at the taxiway in the opposite direction from the 1L-1R-1L 90 degree turn. In this case the arrival angles vary about exactly aligned (0 or 360 degrees). Figures 4.11-a and b represents the push-back of the aircraft with 3° and 357° initial angle from the path i.e. ±3° variation. The results of the projected trajectories are looks to be convincing. Although, the trajectories of wing and tail tips are very close to KOZ, it is not extending adjacent aircraft. This give an insight about the limitations of the 1L-1R-1L path. The pivot of turn over from push back to pull-in occurs at inside the curve when the initial angle is 3° and it occurs at outside when the initial angle is 357°. Recall that the dolly rotating plate allow unrestricted rotation of the dolly relative to the aircraft. Therefore, complete pivot turns are not a constraint on the system feasibility. Scuffing of main landing gear tires may be an operational issue but that is beyond the scope of this assessment.

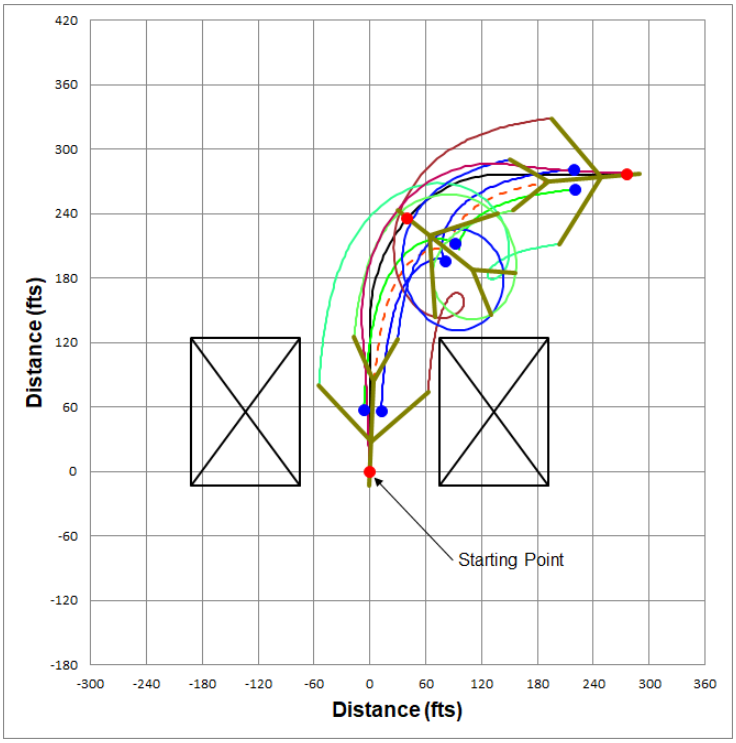


Figure 4.11-a: Push back between two Aircraft with 3° initial offset

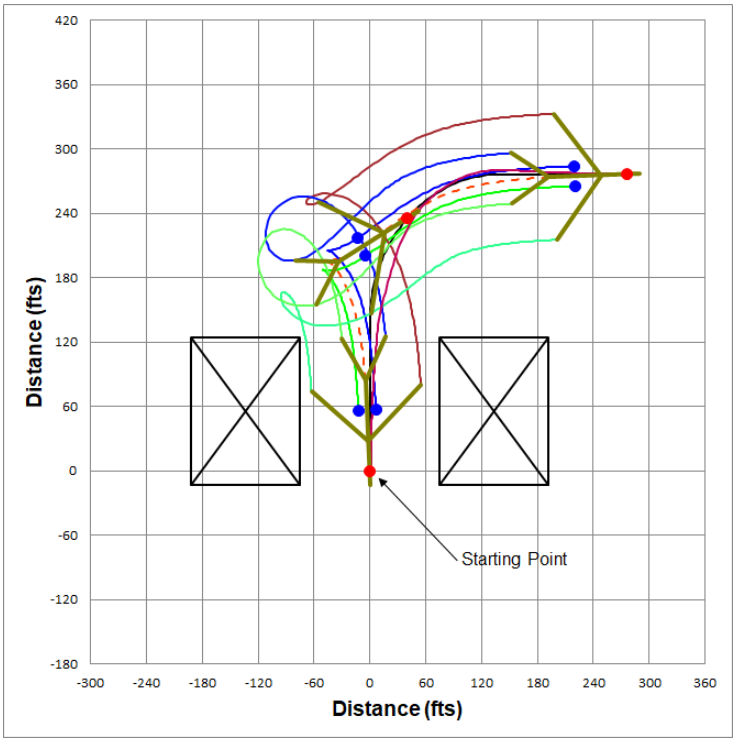


Figure 4.11-b: Push back between two Aircraft with 357° initial offset

Sometimes the initial offset angle may vary beyond ±3°, so it is advisable to check the feasibility of 1L-1R-1L path for push back operation during this situation. To verify this the pushback is simulated with ±4° and the results are shown in figure 4.12-a and 4.12-b.

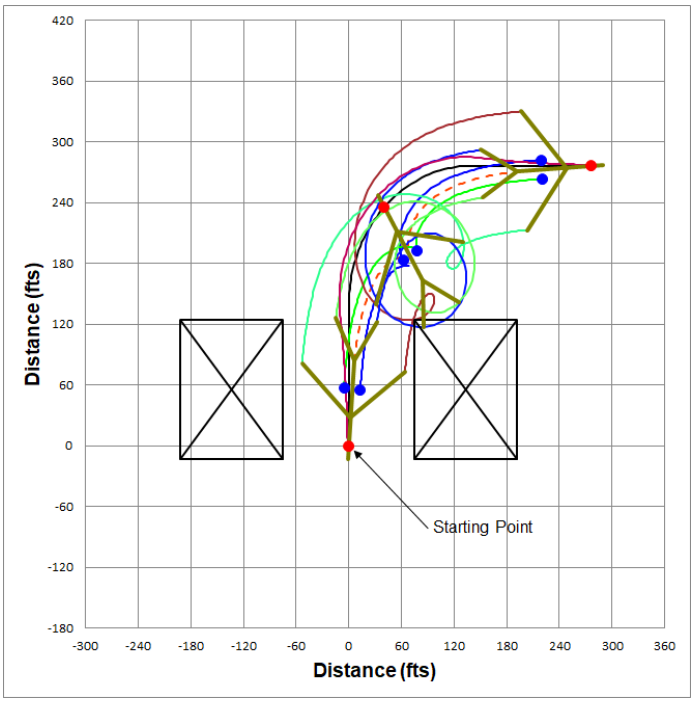


Figure 4.12-a: Push-back between two Aircraft with 4° initial offset

It is clearly shown that the wing and tail tips are intruding into the safe zone of the adjacent aircraft placed on the right when it is pushed back with 4° initial angle. On the other hand, the wing tips are intruding in the adjacent aircraft placed on the left side when it is pushed back with 356°. So, the limitations of this 1L-1R-1L path is that it is not feasible to push back when the aircraft is oriented at an offset angle of higher than 3° on either side of the path.

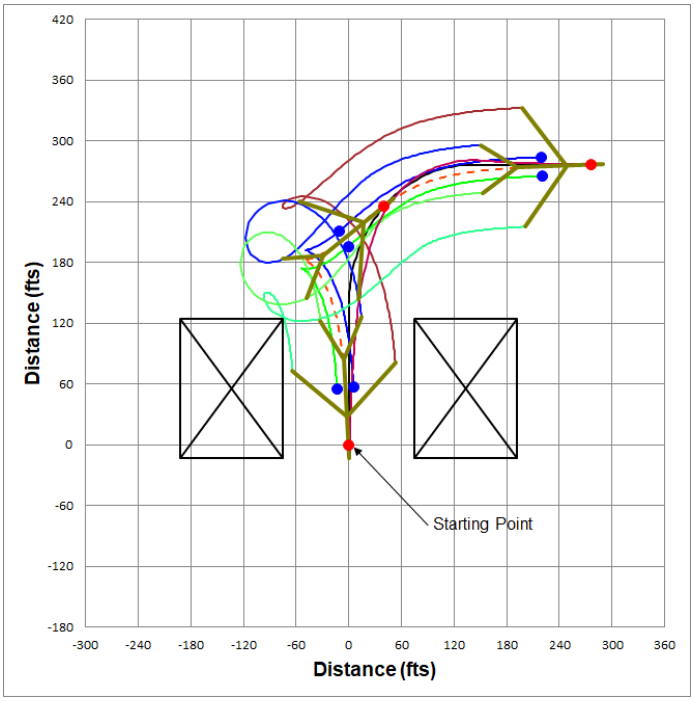


Figure 4.12-b: Push-back between two Aircraft with 356° initial offset

To be complete, pull-ins are analyzed for 270 degrees with ±3° variation. Figure 4.13 shows the pull-in along the 1L-1R-1L path and it is clear that the trajectories are not intruding into the safe zones. This offset angle gives better alignment with the path when the aircraft approaches the gate. The offset angle at end of pull-in operation is 0.81° and it forms 0.21° at the end when it is pushed back from 0.81°. These are the lowest offset angle of this path for both pull-in and push back operation. When it comes to ±3° variation i.e., when the aircraft is pulled in at 267° as initial offset angle the trajectories are intruding the safe zone. It is observed from figure 4.14 that this path is not feasible because the trajectories are entered the KOZ. Note that the KOZ's are conservatively constructed with respect of the empennage outlines.

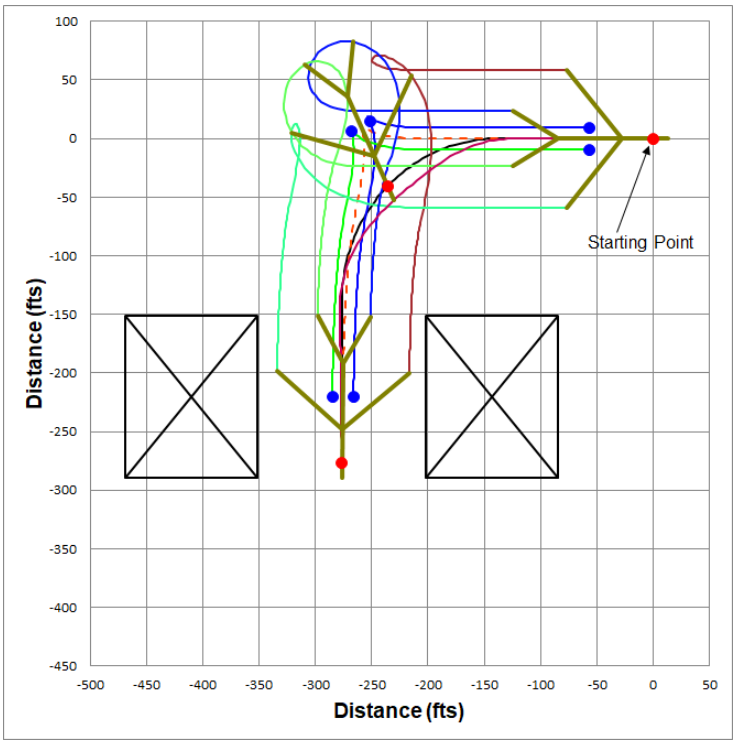


Figure 4.13: Pull in between two Aircraft with 270° at initial position

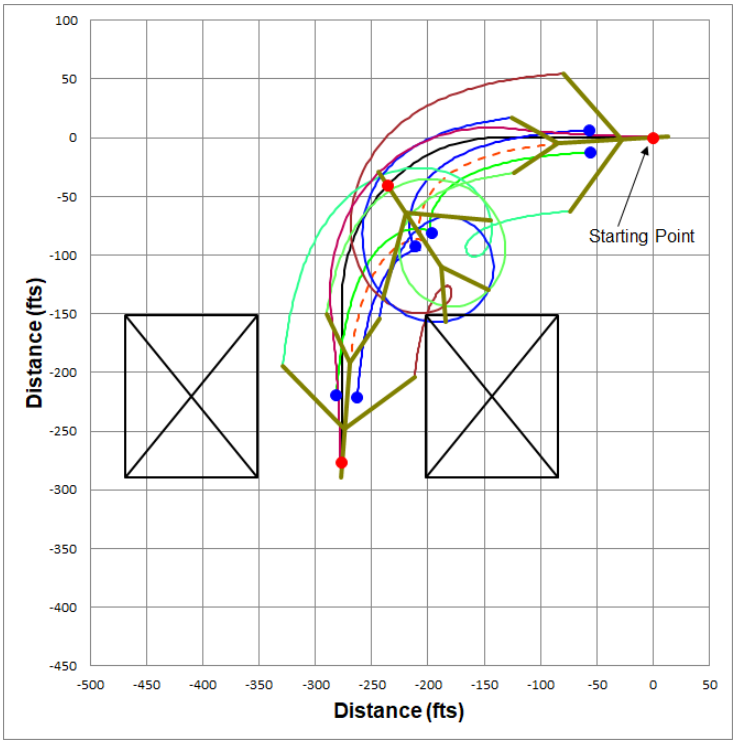


Figure 4.14: Pull in between two Aircraft with 267° at initial position

Since this initial path (1L-1R-1L) is not feasible for certain initial angles, it is better to analyze the offset angles at the end for pull -in and pushback for all initial angles within ±3°. It is evaluated and shown in Table 4.2. The offset angle at the gate for 268° initial angle is higher, but it is within the allowable offset angle. Trajectory of this push back falls inside the KOZ and it is shown in figure 4.15.

Initial Path (1L-1R-1L)			
Pull-in		Push back	
Initial(°)	end (°)	initial (°)	end (°)
87	2.24		-8.43
88	2.24		-8.43
89	2.24		-8.43
90	2.24		-8.42
91	2.24		-8.42
92	2.24		-8.41
93	2.24		-8.41
267	4.47		-3.28
268	17.41		-2.44
269	-0.73		-1.19
270	0.81		0.21
271	1.29		2.08
272	1.52		4.83
273	1.66		9.46

Table 4.2: Offset angle for initial path

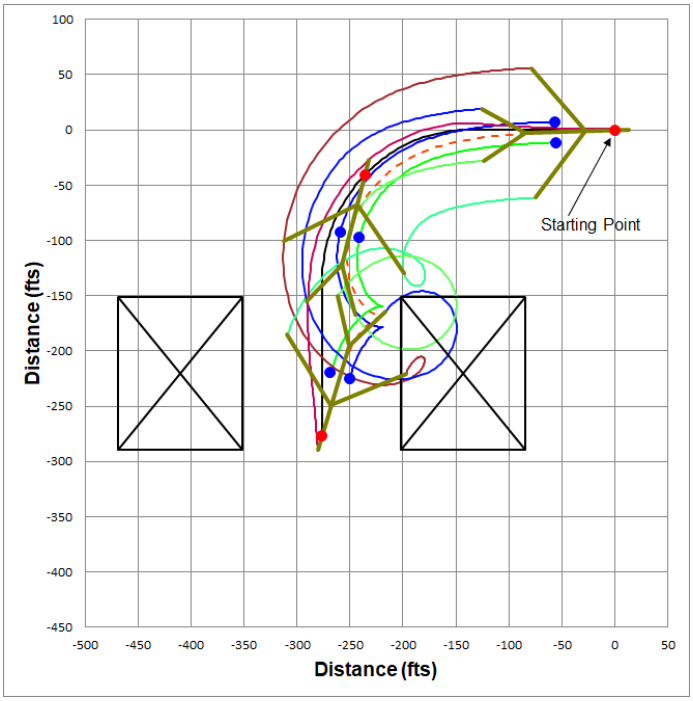


Figure 4.15: Pull in between two Aircraft with 268° at initial position

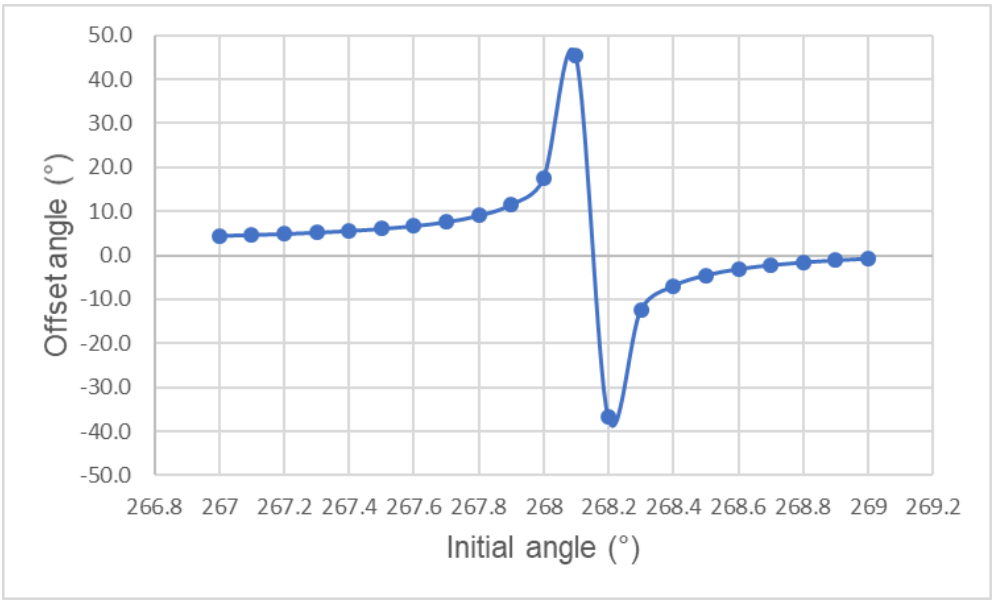


Figure 4.16: Divergence of offset angle for initial path

From Figure 4.16, the offset angle at the end of pull-in is increased gradually from 4.47° to 45.32° for initial angles of 267° to 268.1° respectively. But, for 268.2° it suddenly drops to -36.82° and gradually increases to -0.72° for the initial angle of 269°. From the above analysis, it can be concluded that the 1L-1R-1L path is feasible in all considered scenarios for pull-in and push back except for initial offset angles of greater than 268°. This again highlights the need for good alignment sensing and cueing for the pilots.

Different sizes of aircraft may be handled at a single gate. The predefined path developed for one size may not be feasible for other size aircraft. It is necessary to evaluate the feasibility of predefined path for different aircraft. As an example, the arrival and departure flights are found for gate A13 at Dallas Fort Worth (DFW) airport [15] for a randomly selected day. Airplane sizes are found by using these arrival/departure details [16]. This gate has a capacity to handle B737, B787 and A321 aircraft. B737 and A321 are similar in various aspects, so it is reasonable to analyze the path for B737/A321 with B787. Since B737 is already used as a typical aircraft the feasibility of the path is evaluated with B787 [17]. To perform the evaluation, a pushback with 90° turn is considered with adjacent aircraft. Figure 4.17 shows this pushback operation with 0° initial angle along in a 1L-1R-1L path in B787 gate.

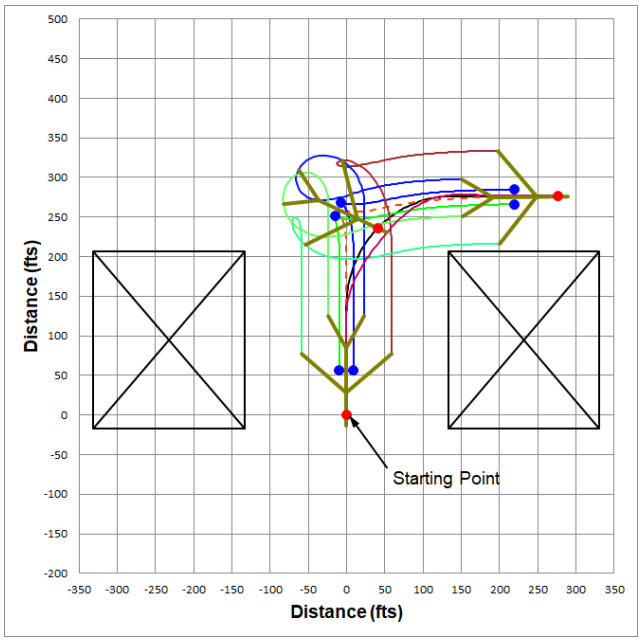


Figure 4.17: Pushback B737 in 1L-1R-1L

It is observed that this path is feasible for this operation at this condition. Now, B787 is pushed back in the same path to evaluate its intrusion for KOZ's and the KOZ's are defined based on B787.

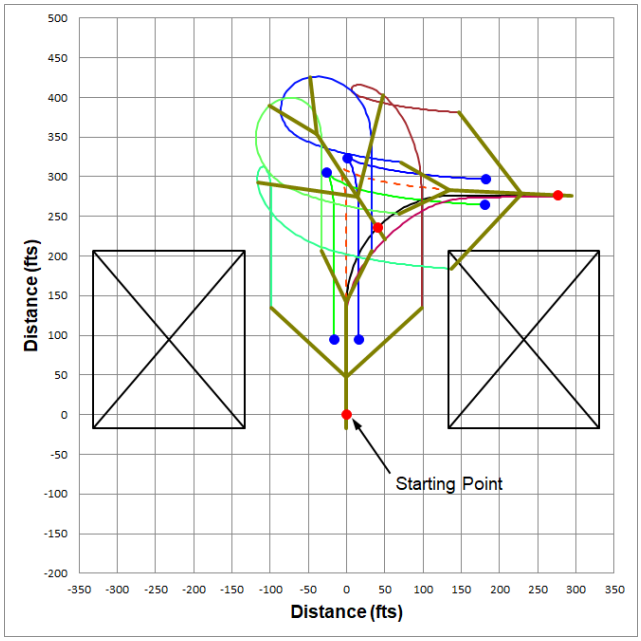


Figure 4.18: Pushback B787 in 1L-1R-1L

It is shown that the 1L-1R-1L path (B737 based) is not feasible for B787 aircraft due to the incursion in the adjacent KOZ. This path would not be feasible for pull-in given the wingtip incursion into the KOZ from the starting position. This predefined path is needing to be modified to perform pushback operation for both B787 and 737 aircraft. To do this, a different path with 1.622L-1.622R-1.622L (a 737 path scaled up based on total length of B787) is considered. The result is shown in Figure 4.19. The aircraft stays out of the KOZs. This informs that it is better to create the predefined path based on the larger aircraft for a gate handling multiple aircraft. Pushback of B737 is performed in 1.622L-1.622R-1.622L path for checking the intrusion in KOZ and this result is shown in figure 4.20. It is clear that the pushback can be performed with B737 and B787 aircraft in a single predefined path (1.622L-1.622R-1.622L) for the specified gate. Feasibility of a predefined path is now shown for paths sized for the largest of aircraft planned for a given gate location.

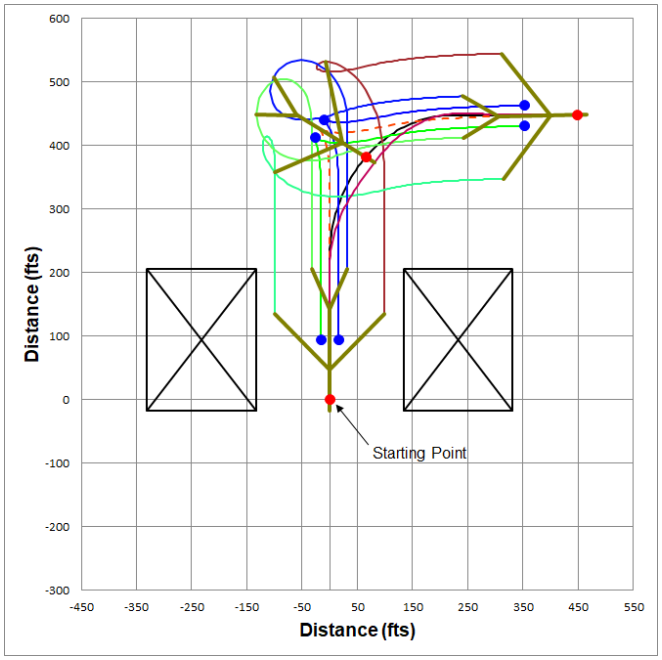


Figure 4.19: Pushback B787 in 1.622L-1.622R-1.622L

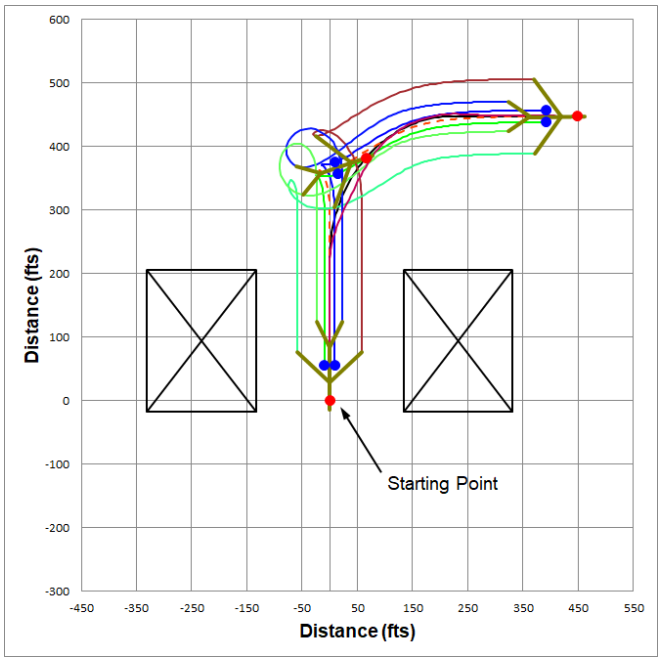


Figure 4.20: Pushback B737 in 1.622L-1.622R-1.622L

4.4.1 PATH IMPROVEMENT

Finding out an improved path is a process of determining a nearest distance to perform an activity without any change in the objectives of the existing method. i.e., the aircraft stay out of KOZ and minimize the ramp depth required. In this thesis, the path improvement refers to the shortest track distance required to park the aircraft with desired orientation during pull-in and to deliver it with an acceptable orientation after push back. Since B737 is simulated earlier with 1L-1R-1L path, the same aircraft is selected and both pull-in and push back operation is performed between adjacent aircraft with a different path. Different radii and length are explored manually to find out the behavior of aircraft for finalizing the improved path. Total length of the improved path is less than the initial path and the feasibility of this path is analyzed in all aspects similar to the initial path. After several trials the following (Figure 4.22) improvement is achieved. This path is similar to 1L-1R-1L in terms of geometric representation, but the overall length of the path is reduced. Different combinations of length and radius are tried out to achieve this path and, in each combination, either one of the aspects are being satisfied during the trial. Paths are decided for trial is based on the turn

radius provided on the aircraft manual [18]. Pull-in and pushback are simulated in these trail paths for various initial angles and example of trail paths shown in figure 4.21. Therefore, this improved path is found out to be feasible for both pull-in and push back operations.

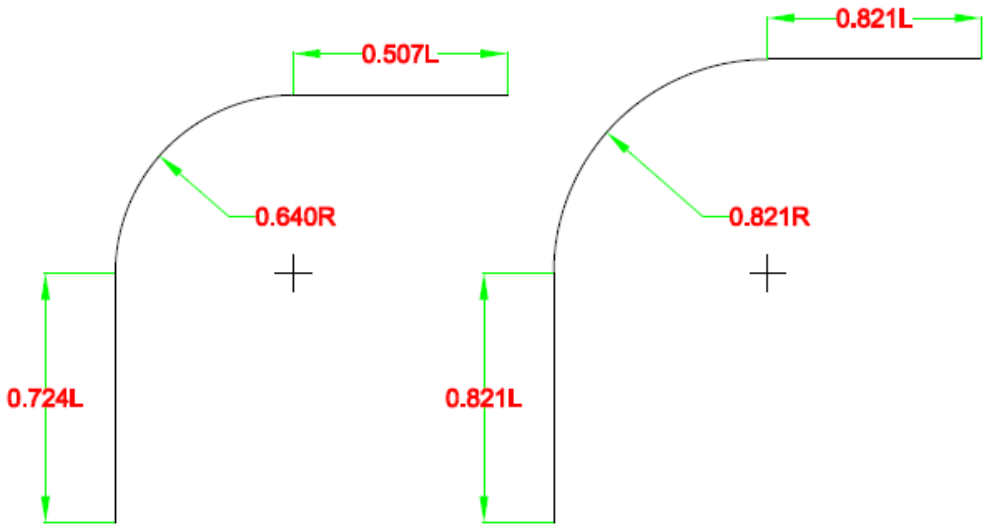


Figure 4.21 : Trail Paths

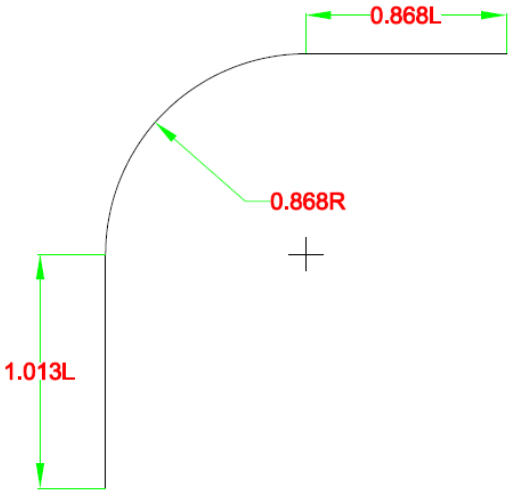


Figure 4.22: Improved path

This improved path has almost the same length as in initial path for vertical and is lesser in radius as well as horizontal segments. Similar to the 1L-1R-1L path, all aspects are analyzed for this improved path. Table 4.4 shows the offset angles for different initial angles for improved path.

Improved Path			
Pull-in		Push back	
Initial (°)	end (°)	initial (°)	end (°)
87	2.47		-16.7
88	2.47		-16.69
89	2.47		-16.67
90	2.47		-16.65
91	2.47		-16.64
92	2.47		-16.62
93	2.47		-16.6
267	164.28		-3.48
268	-3.31		-2.32
269	-0.52		-1.16
270	0.44		0.11
271	0.93		1.54
272	1.22		3.17
273	1.42		5.08

Table 4.3: Offset angle for improved path

From the above Table it is observed that the offset angle (2.47°) at the end of the pull in is not changed for the initial offset angle from 87° to 93°. The trajectories are analyzed for its interference of KOZ's and it is not intruding into the KOZs. Also, the offset angle at the end is well within the allowable range when push back is performed with 2.47° initial offset. The offset angle at the end of pull-in is drastically changed for a small variation between 267° and 268°, so it is necessary to check the trajectory formation. Aircraft does not form any pivot or changes its orientation during pull-in the aircraft at 267° as initial angle. However, the aircraft clears the KOZ's, it travels backwards throughout the path which gives undesirable positioning and orientation near the gate. Figure 4.23 shows this trajectory and it is observed that the initial offset angle plays an important role in the trajectory formation of the aircraft during pull-in and push back operation.

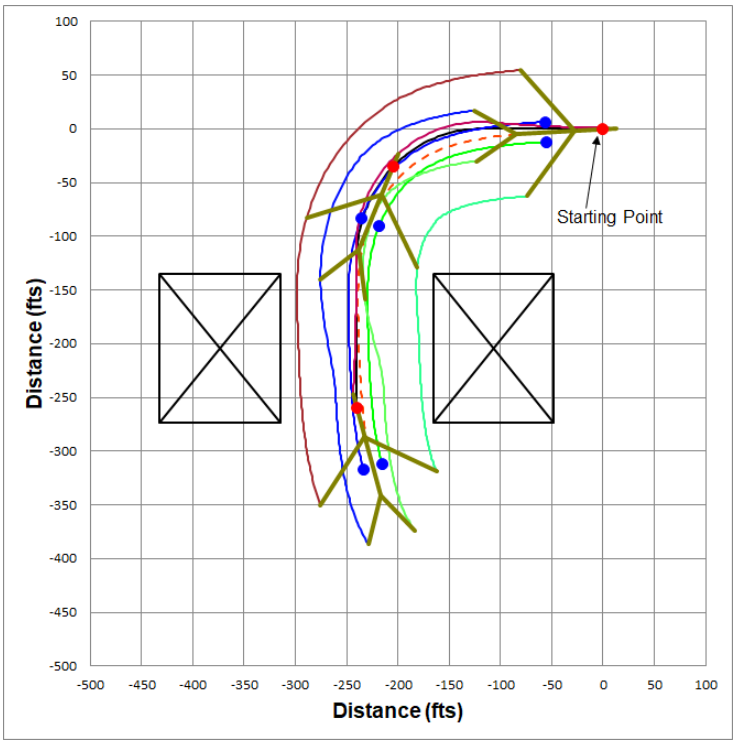


Figure 4.23: Pull-in improved path with 267° initial offset

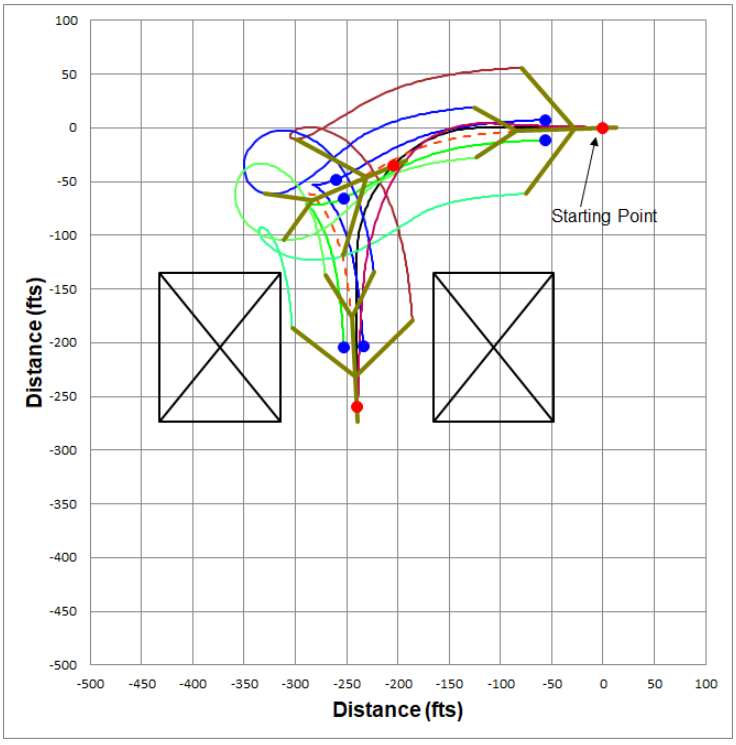


Figure 4.24: Pull-in improved path with 268° initial offset

In this situation the pilot needs to go back and realign the nose gear with a desired offset angle. The pull-in operation with 268° initial offset is shown in Figure 4.24 and it is observed that the aircraft trajectories are well behaved, as well as within the allowable range in all aspects. The same results occurred for other offset angles from 269° to 273° and the arrival angles are varied according to the initial offset angle. So, this improved path is feasible for all pull-in and push back operations except 267°.

Total length of the initial path is 493.4 ft where the total length of the improved path is 448.5 ft. So, the time required to push back/pull in the aircraft should also be less compared to the initial path. This comparison of the initial and improved path is shown in Figure 4.25. The behavior of the aircraft in both initial and improved path is similar in terms of turning from pull-in to pushback, only the trajectory dimensions are varying according to the path. To compare this, pushback is performed with zero degrees as an offset angle. The length between the gate and lateral distance of the taxiway ramp after the main landing gear wheel is reduced in the improved path which gives the reduced space for pushback operations. Some other benefits of improved path are lower installation cost, energy saving and reduced taxi time.

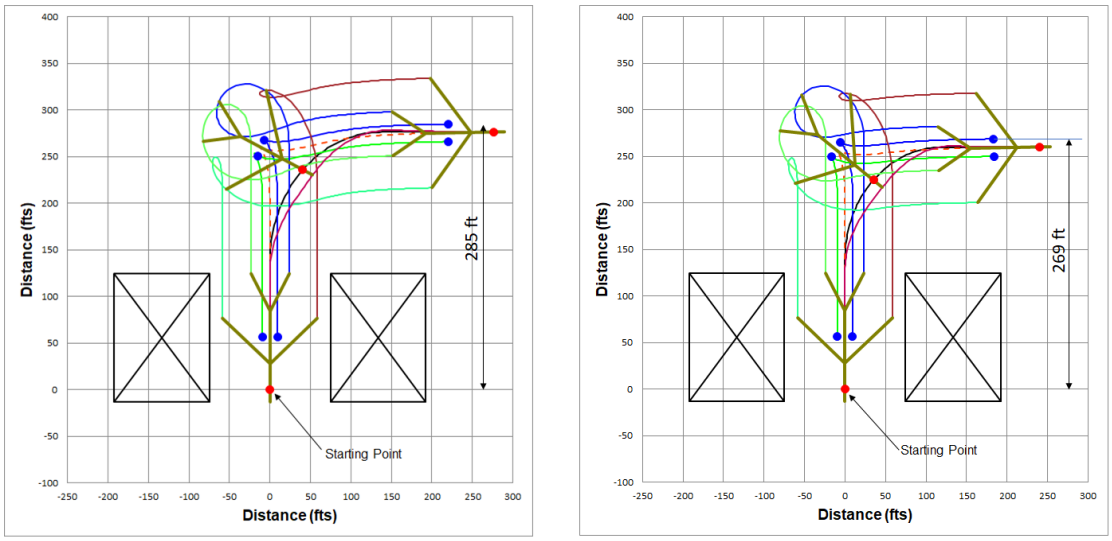


Figure 4.25: Path Comparison

4.4.2 ALTERNATIVE PATH

In initial and improved paths, the offset angle at the gate is around 2.5° during pull-in for 90 ° turn. This might need to be reduced for proper alignment of gate operations like positioning the bridge that connects the aircraft and terminal building, unloading baggage and refueling. To achieve this, a modified path was evaluated, and this path helps to align the main gear with the nose gear better than the previous paths when the aircraft approached the gate. The dimension of the alternative path is shown in Figure 4.26 and the total length of this path is 489.8 ft (based on B737 sizing for the path).

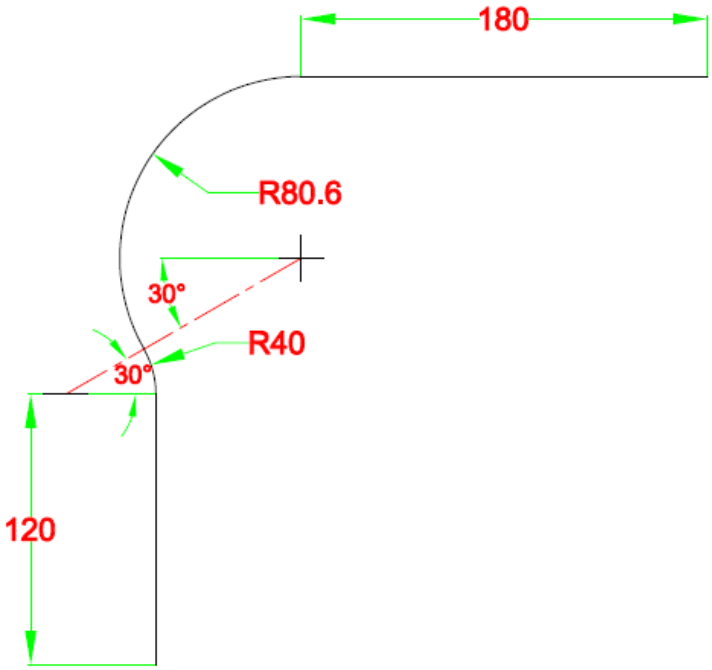


Figure 4.26: Alternative Path (Units : ft)

The offset angle while approaching to gate that is during pull-in operation is 0.45°. This is almost parallel to the path which could not be achievable by the 1L-1R-1L and improved path. So, this jog-in path has a significant improvement in offset angle for ease of near gate operations. This path keeps the aircraft trajectories out of intrusion into the KOZ's during pull-in. The offset angles are

shown in Table 4.4 and it can be inferred that the offset angle at the end is not changing for initial angles of 87° to 93° during pull-in operation and it is almost same for pushback from 0.45°. The difference in the offset angle at the end of pushback is that these trajectories are simulated for an accuracy of decimal places more than hundredths.

Alternative Path			
Pull-in		Push back	
Initial (°)	end (°)	initial (°)	end (°)
87	0.45		-18.12
88	0.45		-18.09
89	0.45		-18.06
90	0.45		-18.04
91	0.45		-18.01
92	0.45		-17.99
93	0.45		-17.96
267	2.55		-3.39
268	6.03		-2.48
269	-6.48		-1.3
270	-1.79		-0.13
271	-0.91		1.14
272	-0.53		2.5
273	-0.32		3.99

Table 4.4: Offset angle for Alternative path

The pull-in trajectory with 90° initial angle is shown in figure 4.27 and the push back from 0.45° is shown in figure 4.28. So, this alternative path is feasible for pull-in and pushback operations which gives lesser difference between aircraft axis and path near the gate. This path has a curved portion called jog-in which helps to change the orientation of the aircraft very close to the path at the end of the pull-in. During pushback, the aircraft pivots inside the radius which utilizes the area inside curved segment and the aircraft makes around -18° as an offset angle at the end. It can be concluded that these offset angles are within the allowable range for the dolly. Also, the total length of the alternative path is almost equal to the initial (1L-1R-1L) path. Variation in offset angle is simulated in the alternative path to check the feasibility.

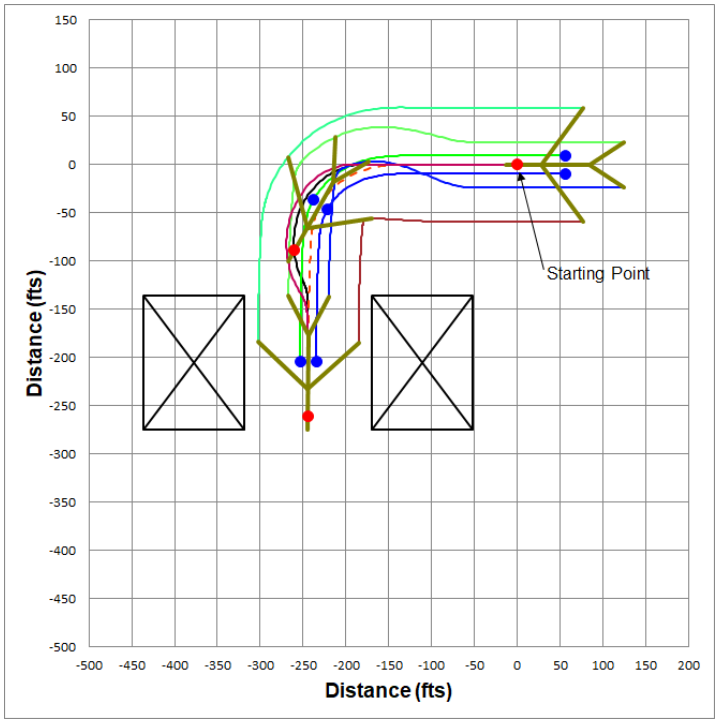


Figure 4.27: Pull-in Alternative Path with 0° initial offset

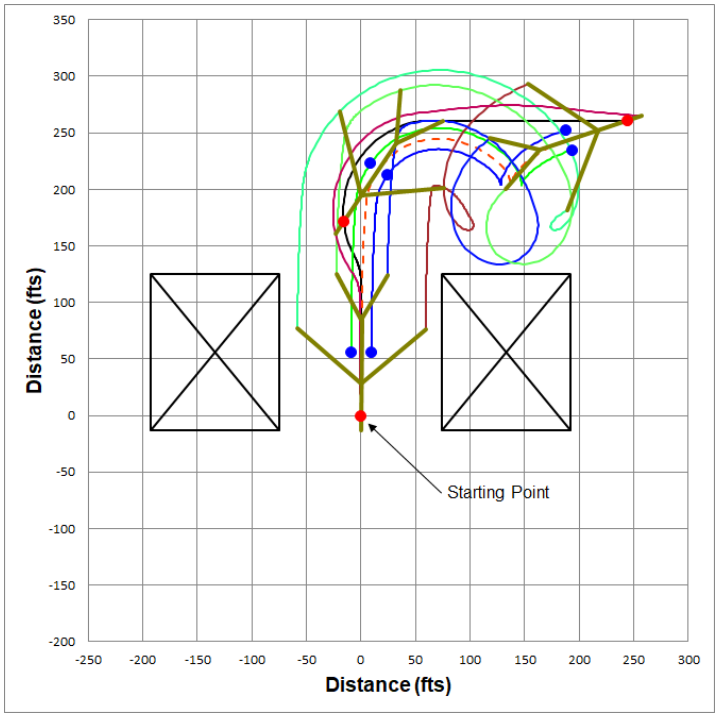


Figure 4.28: Pushback Alternative Path with 0.45° initial offset

The offset angle at the end is increased higher than 2.5°, when the aircraft is pulled in with 268° and 269° as initial angle in the jog-in path. Figure 4.29 shows the formation of trajectory for 268° initial angle and as expected, the wing and tail tips are intruding inside the KOZ of the adjacent aircraft. The angle formed at the end of the pull in is 6.03° and result of pushback from this angle is intruding in to the KOZ.

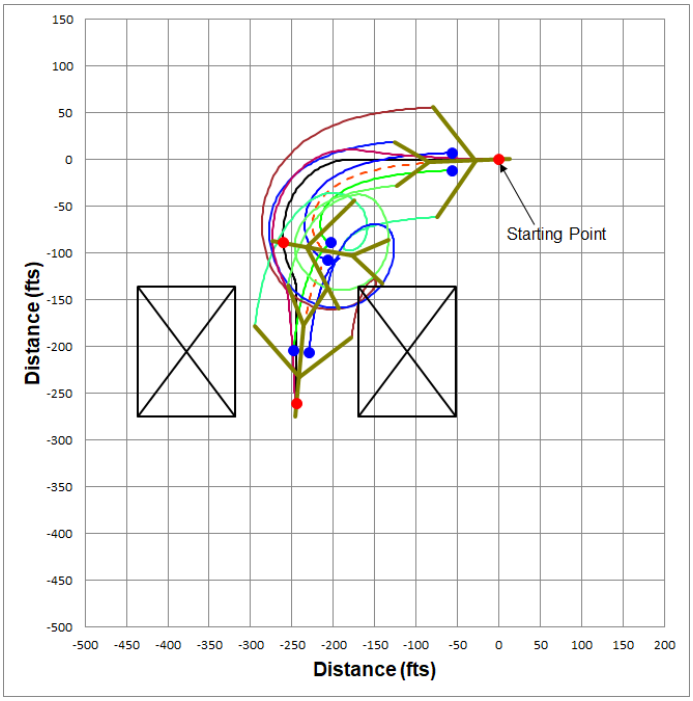


Figure 4.29: Pull in Alternative Path with 268° initial offset

This confirms that the alternative path is not feasible for pull in and push back for some specific initial angles. Similar way the trajectories are evaluated for 269° initial angle and the results are shown in figure 4.30. In this, the wing tips are intruding inside the keep out zone of the aircraft placed on the left side. Form this analysis, it can be concluded that this path provides desirable arrival offset angles for some initial angles during pull-in operation.

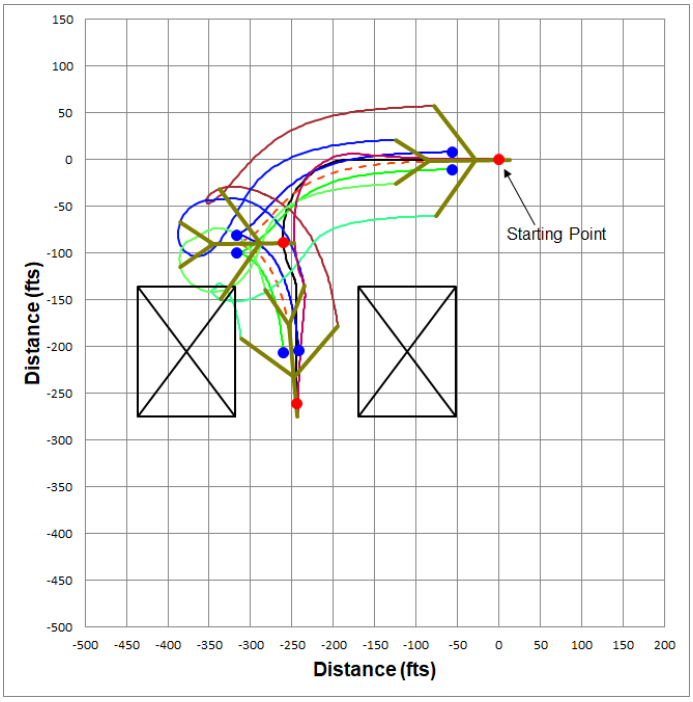


Figure 4.30: Pull in Alternative Path with 269° initial offset

4.5 PATH ANALYSIS

Behavior of aircraft trajectories are varying depends on the path it travels and the initial offset angle. This variation can be analyzed by changing the length and radius of the path. This B737-900 ER aircraft is evaluated with three different turning radii to analyze its behavior. These radii vary from small, medium and large and the dimensions are decided based upon the total length of the aircraft. The largest radii will be 138.17ft (1R) and the medium one will be 69.09 ft (0.5R) and the small is 34.54 ft (0.25R). To compare the variation in the trajectory, horizontal and vertical length of the path is kept constant (1L) and only parameter that changes is radius. The path dimensions are shown in figure 4.31. Offset angle at the end changes according to the radius of the path when the aircraft is pulled in with 90° initial angle. As the radius decreases the offset angle at the end of the pull-in is increases. Also, the aircraft enters inside the KOZ of the adjacent aircraft as the radius decreases and this is shown in Figures 4.32-a,b and c.

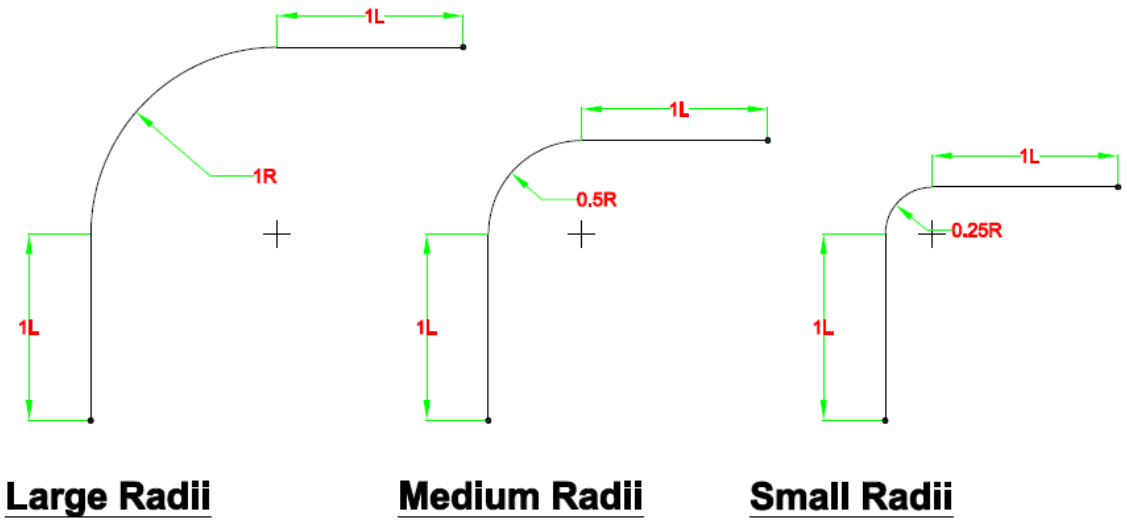


Figure 4.31: Path analysis-different radius

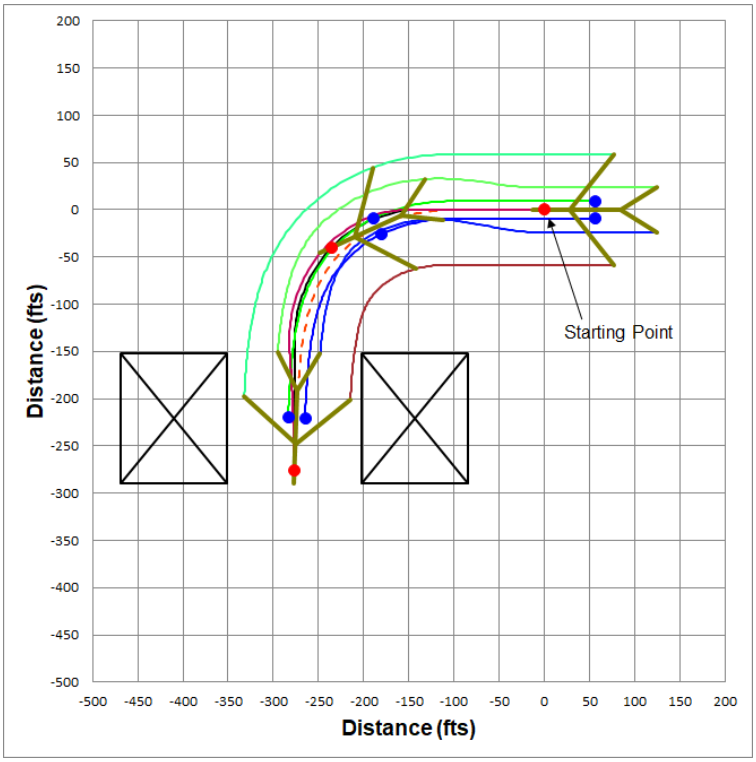


Figure 4.32-a: Trajectory Comparison- Path analysis (Large Radii)

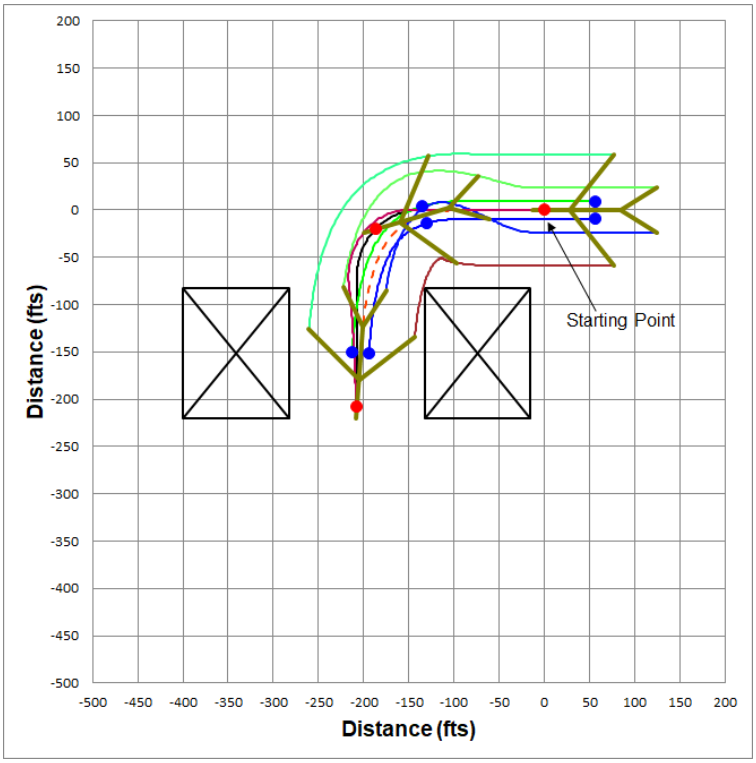


Figure 4.32-b: Trajectory Comparison- Path analysis (Medium Radii)

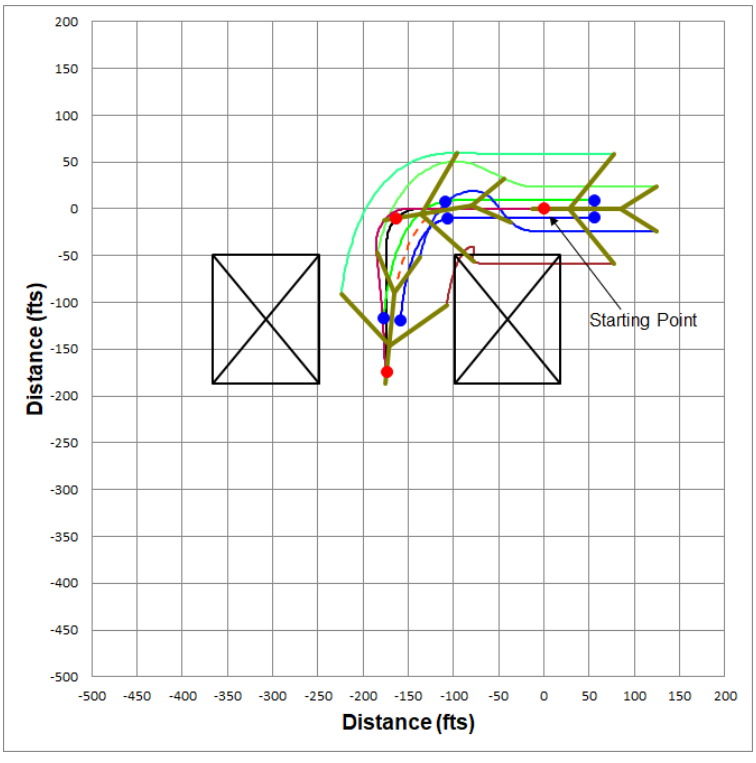


Figure 4.32-c: Trajectory Comparison- Path analysis (Small Radii)

The values of offset angles during pull-in at large, medium and small radii is 2.24°, 4° and 6.05° respectively. As the radius becomes smaller the lateral difference between the nose landing gear and main gear increases. This phenomenon is same for pushback operation and pivot point for push back to pull-in behaves same in all three radii. Also, the length of the path may affect this offset angle at the arrival. To evaluate this the vertical length of the small radii is increased to 1.086L (150 ft) the check the change in behavior of the aircraft. In this the values of the offset angle at the end is reduced to 4.95° from 6.05°. The offset angle decreases as the length of the path increases. Figure 4.33 shows the path of the increased length for small radii (left) and the trajectory formation in that path (right).

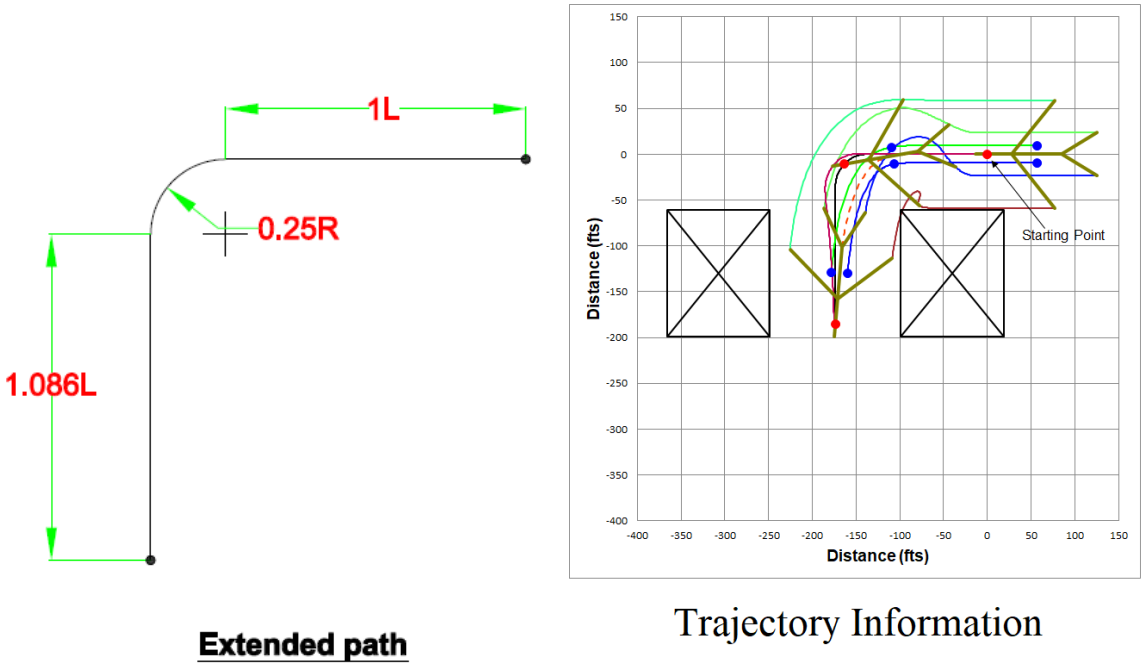


Figure 4.33: Path Analysis-Different length

Paths can be created for a specific aircraft by using the numerical values for dimensions in several units such as feet, meter, etc. But, when creating the path in generic way for different aircraft it is better to define the path in terms of total length of the aircraft. For example, the initial path is referred as 1L-1R-1L and this will help us to find an overall sense about how the trajectories will

behave. If we consider B777 aircraft for pull in/push back with 1L-1R-1L with its total length, the trajectory behaves similar to the initial path in B737. This is because the trajectories are calculated based on the geometric representation of the aircraft. Since, most of the aircraft behave similarly, this nomenclature is easier to work with and reduces the confusion. For example, the improved path is represented in terms of aircraft length as 1.013L-0.868L-0.868L.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSIONS

In this project, a macro is developed to model airplane trajectories for given NLG paths. The algorithm is based on kinematic relationships and has the capability to generate the trajectories on various points of the aircraft as the NLG is moved along a predefined path. KOZ's are represented in the algorithm to verify the feasibility of different paths. A scaled physical model of B737-900 ER was built and an experiment conducted to validate the simulation. The results from simulation and experiment are compared and show that the model provides similar results. The model is also compared to paths provided by the OEM for airport planning. These comparisons also showed very good agreement. The analysis of initial 90 degree turn (1L-1R-1L) path showed feasible outcomes when initial orientation angles at pull were less than 3 degrees off of straight alignment. During this, it is found that the aircraft pivots inside the radius when the initial offset angle is positive and outside the radius when the initial angle is negative. Aircraft larger and smaller than the baseline example were evaluated using the baseline path. Although the aircrafts show similar kind of trajectories, the offset angle at the end of pull-in/pushback of larger aircraft is higher than smaller aircraft.

With this result, an improved path is developed, and its feasibility is investigated to reduce overall track length and ramp space required. As a part of analysis, the maximum allowable offset angle for the dolly is calculated and it is verified with the offset angles produced at the end of pull-in/push

back in different paths. To reduce the offset angle near the gate an additional path is created with a jog-in. This path allowed the aircraft to be very closely aligned at the end of pull-in (near gate). However, it requires tighter initial pull-in alignment and is less accommodating of aircraft larger than baseline. Overall, the results provide evidence that a fixed track pull-in and push-back architecture can provide desired aircraft trajectories in a typical terminal layout.

5.2 FUTURE WORK

Although this thesis focuses on the feasibility of aircraft push back trajectories on various paths, a couple of significant design aspects are highlighted here that deserve additional research. The model in this study include only the geometry and kinematic relationships. Real-world effects like frictional forces and Inertial forces need to be included in future analytical and experiment work to ensure the results remain acceptable prior to system implementation. There is also a need to provide pilots with precise and accurate steering cues during mounting of the dolly to keep the NLG near center and aircraft within alignment tolerances. It is envisioned that the ground-based system will need to detect and track inbound aircraft and then provide steering cues via existing links or visual signaling. It would be highly desirable to avoid requiring costly aircraft modifications for special equipment to use the fixed track system. Future work may involve system specification development and investigation of sensor technologies that could provide required information in all operational environments. The software algorithms for guidance and for pilot cueing are also areas where significant work could be done.

REFERENCES

[1] https://en.wikipedia.org/wiki/Pushback

[2] http://www.at-system.eu/system.html

[3] Franziska Dieke-Meier, Thomas Kalms, Hartmut Fricke and Michael Schultz Modeling Aircraft Pushback Trajectories for Safe Operations, ATACCS 2013, pp.76-84.

[4] Erza Hauer, Determination of Wheel Trajectories, 1970, Transportation Journal of ASCE, pp.463-470.

[5] Wilfred Revell and John T Tuohey, Discussion of "Determination of Wheel Trajectories by Ezra Hauer", 1971, Transportation Journal of ASCE, pp.706-710.

[6] Giorgio Figliolini and Chiara Lanni, Kinematic Analysis of the Planar Motion of Vehicles When Travelling Along Tractrix Curves, 2015, IDETC/CIE 2015.

[7] Jiening Wang, Jiang Qu, Yuandi Zhao and Mei Dong, Research and Simulation of An Aircraft Behavior During U-turn Maneuver, 2013, Applied Mechanics and Materials. pp.1742-1749.

[8] Juraj Kardoˇs, AutoTaxi System Design for Aircraft, 2015, excel@fit, brno University of Technology.

[9] Jaroslav Bursik, Jakub Kraus and Marek Stumper, Automation of Taxiing, 2017, Magazine of Aviation Development.

[10] Timothy D. Ropp, jin young Kim, Keaton Aktay and Kory Aktay, ANTS-Automated NextGen Taxi System, Design Competition in FAA, 2010.

[11] https://earth.google.com/web/@31.3274112,-92.5451856,24.59227193a,886.73441395d,35y,0h,45t,0r/data

[12] https://www.flightradar24.com/data/airports/aex/departures

[13] https://earth.google.com/web/@32.8555195,-97.0390841,169.12173786a,870.5188126d,35y,0h,45t,0r/data

[14] https://www.airfleets.net/exploit/exploitation.htm

[15] https://www.flightview.com/airport/DFW-Dallas-TX-(Dallas-Ft_Worth)/arrivals

[16] https://flightaware.com/live/flight/CSN1716

[17] Boeing 787 Airplane Characteristics for Airport Planning, December 2015.

[18] Boeing 737 Airplane Characteristics for Airport Planning, September 2013.

[19] Zöbel, D. Mathematical Modeling of the Kinematics of Vehicles, 2001, Mathematical Modeling of Technical Processes. Socrates/Erasmus Summer School, pp. 178-200.

[20] EMB135 Airport Planning Manual, Embraer S. A., APM-145/1238, Rev L, January 2008.

[21] 777-200LR / -300ER / -Freighter Airplane Characteristics for Airport Planning, Boeing Commercial Airplanes, D6-58329-2, May 2015.

[22] Aerodrome Design Manual, Part 2 Taxiways, Apron and Holding Bays, Fourth Edition-2005, International Civil Aviation Organization, DOC 9157 AN/901, 2005.

[23] Thorsten Luettel, Michael Himmelsbach and Hans-Joachim Wuensche, Autonomous Ground Vehicles-Concepts and a Path to the Future, Proceedings of the IEEE, 2012, pp.1831-1839.

[24] Evgeni D Ganev, Electric Drives for Electric Green Taxiing System, IEEE Electrification Magazine, December 2017.

[25] https://www.mototok.com/free-consulting-tugs?gclid=EAIaIQobChMI3_zqoqGg4QIVApJbCh3EYQJ5EAAYASAAEgIvFvD_BwE

[26] https://www.slideshare.net/jaspreetrooprai138/airport-engineering

[27] https://www.hindustantimes.com/delhi-news/close-shave-at-delhi-airport-as-wings-of-ai-ethiopian-airlines-planes-collide/story-TDozo3AtStxcphnhYx8JdI.html

[28] http://code7700.com/pdfs/runway_pave_load.pdf

[29] https://www.youtube.com/watch?v=RCFeyKBybog

[30] CRJ 200 Airport Planning Manual, Bombardier Volume 1, Rev 8, January 2016.

[31] CRJ 700 Airport Planning Manual, Bombardier Volume 1, Rev 15, December 2015.

[32] Boeing 747-8 Airplane Characteristics for Airport Planning, Rev B, December 2012.

[33] Boeing 757-200/300 Airplane Characteristics for Airport Planning, Rev F, August 2002.

[34] Boeing 767 Airplane Characteristics for Airport Planning, Rev H, December 2005.

[35] Boeing 787 Airplane Characteristics for Airport Planning, Rev L, December 2015.

[36] A 300 Airplane Characteristics for Airport Planning, Rev 22, December 2009.

[37] A 320 Aircraft Characteristics Airport and Maintenance Planning, Rev 31, May 2016.

[38] A 340-500/-600 Aircraft Characteristics Airport and Maintenance Planning, Rev 13, October 2016.

[39] A 380 Aircraft Characteristics Airport and Maintenance Planning, Rev 15, December 2016.

[40] https://www.airliners.net/forum/viewtopic.php?t=1355819

[41] Cort J Willmott, Some Comments on the Evaluation of Model Performance, Center for Climate Research, Department of Geography, University of Delaware, Newark, del. 19711.

APPENDICES

APPENDIX-A

Increament	Path		Track Point 1		Track Point 2		Track Point 3		Track Point 4	
	X	Y	X	Y	X	Y	X	Y	X	Y
Start	0.00	0.00	7.75	10.13	-7.75	10.13	3.13	16.44	-3.13	16.44
6	0.00	6.00	7.50	16.31	-7.88	15.81	2.81	22.13	-3.50	22.38
12	0.00	12.00	7.13	22.63	-8.25	22.50	2.19	28.50	-4.06	28.06
18	0.00	18.00	6.25	29.19	-9.00	26.88	0.88	34.63	-5.38	33.63
24	0.93	23.93	3.38	36.50	-10.44	29.75	-3.31	40.13	-8.94	37.31
30	3.73	29.23	-3.00	40.31	-8.81	26.00	-10.44	38.50	-12.50	32.63
36	8.10	33.34	-3.69	39.00	-0.63	23.81	-8.94	33.44	-7.63	27.19
42	13.56	35.80	1.19	40.06	6.13	25.38	-3.31	33.81	-1.94	27.81
48	19.53	36.40	7.69	42.13	10.75	27.00	2.50	36.63	3.75	30.31
54	25.53	36.40	14.44	43.31	15.81	27.81	8.69	38.63	9.25	32.00
60	31.53	36.40	20.81	43.81	21.44	28.44	14.75	39.13	15.00	32.75
End	36.40	36.40	26.00	44.06	26.38	28.63	19.88	39.50	20.00	33.13

Table A-1: Values Measured from Experiment (Corresponding to Figures: 3.19-a,b,c and d)

Increament	Path		Track Point 1		Track Point 2		Track Point 3		Track Point 4	
	X	Y	X	Y	X	Y	X	Y	X	Y
Start	0.00	0.00	7.73	10.15	-7.73	10.15	3.10	16.42	-3.10	16.42
6	0.00	6.00	7.73	16.15	-7.73	16.15	3.10	22.42	-3.10	22.42
12	0.00	12.00	7.73	22.15	-7.73	22.15	3.10	28.42	-3.10	28.42
18	0.00	18.00	7.73	28.15	-7.73	28.15	3.10	34.42	-3.10	34.42
24	0.93	23.93	2.13	36.63	-11.00	28.46	-5.12	39.51	-10.38	36.23
30	3.73	29.23	-5.19	38.36	-7.44	23.07	-12.07	34.69	-12.97	28.56
36	8.10	33.34	-3.66	38.28	0.21	23.32	-8.57	32.23	-7.02	26.23
42	13.56	35.80	1.46	39.84	6.45	25.21	-2.98	33.43	-0.98	27.56
48	19.53	36.40	7.91	41.65	11.38	26.59	2.83	35.73	4.23	29.69
54	25.53	36.40	14.50	42.80	16.43	27.46	8.85	37.42	9.63	31.27
60	31.53	36.40	20.87	43.41	21.94	27.99	14.93	38.35	15.36	32.17
End	36.40	36.40	25.94	43.70	26.59	28.25	19.86	38.81	20.12	32.62

Table A-2: Values Calculated from Simulation (Corresponding to Figures: 3.19-a,b,c and d)

The column represented as increment in the above Tables represent the 6-inch increment from the start point to the end point along the predefined path. X and Y are the coordinate values with the corresponding points. This coordinate values for the path is measured from the starting point.

Drag Point 1 (left)					Drag Point 2 (right)			
Simuation		Airplane manual			Simuation		Airplane manual	
X	Y	X	Y		X	Y	X	Y
-11.00	-56.33	-11.12	-50.10		11.00	-56.33	11.12	-50.79
-11.00	-6.83	-11.12	-5.86		11.00	-13.43	11.12	-6.54
-11.00	17.92	-11.07	38.39		11.00	17.92	11.40	37.70
-10.09	58.00	-6.88	82.86		11.85	56.43	15.59	81.85
-0.41	104.10	9.59	124.83		20.25	96.53	34.45	121.99
23.28	143.04	40.75	158.39		39.65	128.34	69.38	149.17
61.09	169.54	81.22	177.21		70.07	149.46	112.24	161.31
110.61	183.43	125.79	185.64		114.48	161.78	156.55	166.72
162.01	189.34	170.32	190.29		163.63	167.40	201.22	169.46
213.36	191.78	211.46	191.85		214.03	169.79	212.15	169.67

Table A-3: Values for 90° turn (Corresponding to Figures: 3.21 and 3.22)

Drag Point 1 (left)					Drag Point 2 (right)			
Simuation		Airplane manual			Simuation		Airplane manual	
X	Y	X	Y		X	Y	X	Y
-11.00	-56.33	-11.24	-56.89		11.00	-56.33	11.26	-56.89
-11.00	-5.63	-11.24	-9.48		11.00	-5.63	11.34	-9.48
-11.00	19.72	-11.45	37.92		11.00	19.72	11.42	37.92
-8.24	75.42	-7.10	85.14		13.53	72.26	16.57	86.12
17.99	138.46	11.23	130.77		35.48	125.10	36.66	129.44
72.41	174.83	44.31	165.54		78.57	153.70	76.34	156.40
130.93	172.59	89.98	181.38		124.15	151.66	124.45	154.88
175.11	149.90	137.99	174.59		163.02	131.52	167.71	133.23
215.07	119.41	179.60	151.77		200.92	102.56	204.91	103.50
252.34	86.15	256.94	87.32		237.37	70.03	241.57	70.84

Table A-4: Values for more than 90° turn (Corresponding to Figures: 3.23 and 3.24)

APPENDIX-B

Excel ™ VBA code used to develop the model is presented in this appendix.

```
Option Explicit

Sub format_sheet1()

Dim data_points As Integer  'number of data point used to calculate each path

Dim show_points As Integer  'number of data points that appear on graph per path

Dim Track_points As Integer 'number of points on the aircraft that are being tracked, not
counting the drive and drag points

Dim drag_points As Integer  'number of drag points

Dim cht As ChartObject      'the graph

Dim i As Integer            'counter

'deletes the graph from the sheet

For Each cht In Sheet1.ChartObjects

   cht.Delete

Next

'Inputs values from the sheet

data_points = Sheet1.Cells(2, 3).Value

Track_points = Sheet1.Cells(2, 12).Value

'clears cells and removes any color applied

'Does not clear the input cells

Sheet1.Range(Cells(7, 5), Cells(2 * data_points, 500)).ClearContents

Sheet1.Range(Cells(6, 11 + 2 * Track_points), Cells(6, 500)).ClearContents

Sheet1.Range(Cells(5, 11 + 2 * Track_points), Cells(5, 500)).ClearContents

Sheet1.Range(Cells(6, 5), Cells(3 * data_points, 500)).Interior.Color = xlNone

'Applies titles to the sheet

Sheet1.Cells(5, 2).Value = "Point number"

Sheet1.Cells(5, 3).Value = "Drive point X"

Sheet1.Cells(5, 4).Value = "Drive point Y"
```

```
Sheet1.Cells(5, 5).Value = "Trace point X"

Sheet1.Cells(5, 6).Value = "Trace point Y"

Sheet1.Cells(5, 7).Value = "Drag point1 X"

Sheet1.Cells(5, 8).Value = "Drag point1 Y"

Sheet1.Cells(5, 9).Value = "Drag point2 X"

Sheet1.Cells(5, 10).Value = "Drag point2 Y"

Sheet1.Cells(2, 15).Interior.Color = RGB(207, 240, 210)

'Applies titles for tracking points to sheet

For i = 0 To Track_points - 1

Sheet1.Cells(5, 2 * i + 11).Value = "Track point " & i + 1 & " X"

Sheet1.Cells(5, 2 * i + 12).Value = "Track point " & i + 1 & " Y"

Sheet1.Cells(5, 2 * i + 13).Value = "Degree of Tangent"

Next i

'Applies titles for wing and tail centers

Sheet1.Cells(5, 2 * Track_points + 13).Value = "Wing center X"

Sheet1.Cells(5, 2 * Track_points + 14).Value = "Wing center Y"

Sheet1.Cells(5, 2 * Track_points + 15).Value = "Tail center X"

Sheet1.Cells(5, 2 * Track_points + 16).Value = "Tail center Y"

Sheet1.Cells(5, 2 * Track_points + 13).Font.Color = RGB(255, 0, 0)

'Applies data point numbers to sheet

For i = 0 To data_points

Sheet1.Cells(7 + i, 2).Value = i

Sheet1.Range(Cells(7 + data_points + 1, 2), Cells(7 + data_points + 100, 2)).ClearContents

Next i

'Makes Invisible for wing center and tail center

For i = 0 To data_points + 2

Sheet1.Cells(5 + i, 2 * Track_points + 13).Font.Color = RGB(255, 255, 255) 'this color makes
wing center x invisible
```

```
Sheet1.Cells(5 + i, 2 * Track_points + 14).Font.Color = RGB(255, 255, 255) 'this color makes wing center y invisible

Sheet1.Cells(5 + i, 2 * Track_points + 15).Font.Color = RGB(255, 255, 255) 'this color makes tail center x invisible

Sheet1.Cells(5 + i, 2 * Track_points + 16).Font.Color = RGB(255, 255, 255) 'this color makes tail center y invisible

Next i

'Colors relevent cells green

'Green cells mean this is a user input

Sheet1.Cells(2, 3).Interior.Color = RGB(207, 240, 210)

Sheet1.Cells(2, 6).Interior.Color = RGB(207, 240, 210)

Sheet1.Cells(2, 9).Interior.Color = RGB(207, 240, 210)

Sheet1.Cells(2, 12).Interior.Color = RGB(207, 240, 210)

Sheet1.Range(Cells(6, 3), Cells(6, 10 + 2 * Track_points)).Interior.Color = RGB(207, 240, 210)

Sheet1.Range(Cells(6, 3), Cells(6 + data_points, 4)).Interior.Color = RGB(207, 240, 210)

End Sub

Sub wheel_sim()

'defining variables

Dim data_points As Integer  'Number of data points used to calculate each path

Dim show_points As Integer  'Number of data points being shown on the graph per path

Dim Track_points As Integer 'Number of points on the aircraft that are being graphed not counting drive and drag points

Dim dtheta As Double  'The angle between where the aircraft is pointing, and its movement path

Dim theta As Double         'angle nose is pointing using unit circle angles

Dim phi As Double           'angle of a tracking point to the drive point using unit circle angles. Used to calculate Track_pos()

Dim p_theta As Double       'angle path is pointing using unit circle angles

Dim Length_DD1 As Double    'distance from drive to drag point1

Dim Length_DD2 As Double    'distance from drive to drag point2

Dim Length_DTCE As Double   'distance from drive to Trace point

Dim Length_DT() As Double   'distance from drive to track points
```

```
Dim Length_T As Double  'Distance from track point to drive point, used for calculating
Track_pos()

Dim Track_pos() As Double   'Position of track points relative to drive and trace points

Dim i As Integer            'counter for data points

Dim j As Integer            'counter for x and y ( 0 for X and 1 for Y)

Dim k As Integer            'counter for track points

Dim Drive() As Double       'Data points for the Drive path. Drive is the nose gear, or the point the
object is being pulled from

Dim Trace() As Double       'Trace is the center point of the main gear, or the point that defines
the point the aircraft rotates about for each data point.

Dim Drag1() As Double       'Data points for the drag point1.

Dim Drag2() As Double       'Data points for the drag point2.

Dim Track() As Double       'Data points for each of the track points

Dim Drag() As Double        'data ponits for each of drag points

Dim drag_pos() As Double    'Position of track points relative to drive and trace points

Dim Drag1_pos() As Double   'Data points for each of the Drag1 points

Dim Drag2_pos() As Double   'Data points for each of the Drag2 points

Dim min(0 To 1) As Integer  '0 is x and 1 is y. extreme positions which will be used to define
graph limits

Dim max(0 To 1) As Integer  '0 is x and 1 is y. extreme positions which will be used to define
graph limits

Dim pi As Double            '3.14 blah blah blah

Dim Alpha0 As Double        'angle between drive and trace

Dim phi_D1 As Double        'angle between drive and drag1

Dim phi_D2 As Double        'angle between drive and drag2

Dim Beta() As Double        'Angle between drive and track points

Dim NG_angle As Double      'Nose Gear angle fron aircraft's axis

Dim NGA_RAD As Double       'Nose gear anle in radinas

Dim Alpha1 As Double        'Initial Angle between drive and drag

Dim drag_points As Integer  'Number of points for main gear

Dim Length_IT As Double     'Length between drive and track points for initial nose angle
```

```
Dim Track_posIT() As Double 'Track position to calculate track points for nose angles
Dim phi_IT As Double        'angle between drive and track points at initail nose angles
Dim wing_cen() As Double    'data points for Wing center
Dim Length_wc As Double     'Length between drive point ans wing center
Dim theta_wc As Double      'angle between drive and wing center
Dim tail_cen() As Double    'data points for tail center
Dim Length_tc As Double     'Length between drive point ans tail center
Dim theta_tc As Double      'angle between drive and tail center
'//CALCULATING INITIAL VALUES//
'Inputs Values
data_points = Sheet1.Cells(2, 3).Value
show_points = Sheet1.Cells(2, 6).Value
NG_angle = Sheet1.Cells(2, 9).Value
Track_points = Sheet1.Cells(2, 12).Value
drag_points = 2
pi = 4 * Atn(1) 'defines pi as 3.14...
Sheet1.Cells(6, 2 * Track_points + 13).Value = ((Sheet1.Cells(6, 3).Value) + (Sheet1.Cells(6, 5).Value)) / 2 'Defines the X value for center point of wings
Sheet1.Cells(6, 2 * Track_points + 14).Value = ((Sheet1.Cells(6, 4).Value) + (Sheet1.Cells(6, 6).Value)) / 2 'Defines the Y value for center point of wings
Sheet1.Cells(6, 2 * Track_points + 15).Value = 3 * ((Sheet1.Cells(6, 3).Value) + (Sheet1.Cells(6, 5).Value)) / 2 'Defines the X value for center point of tail
Sheet1.Cells(6, 2 * Track_points + 16).Value = 3 * ((Sheet1.Cells(6, 4).Value) + (Sheet1.Cells(6, 6).Value)) / 2 'Defines the Y value for center point of tail
'Defines the array size for relevent variables
ReDim Drive(1, 0 To data_points) '0 for X and 1 for Y, timestep
ReDim Trace(1, 0 To data_points) '0 for X and 1 for Y, timestep
ReDim Drag1(1, 0 To data_points) '0 for X and 1 for Y, timestep
ReDim Drag2(1, 0 To data_points) '0 for X and 1 for Y, timestep
ReDim Drag1_pos(1, 0 To data_points) '0 for X and 1 for Y, timestep
```

```
ReDim Drag2_pos(1, 0 To data_points) '0 for X and 1 for Y, timestep

ReDim wing_cen(1, 0 To data_points) '0 for X and 1 for Y, timestep

ReDim tail_cen(1, 0 To data_points) '0 for X and 1 for Y, timestep

If Track_points > 0 Then

ReDim Track(1 To Track_points, 1, 0 To data_points) 'track point, 0 for X and 1 for Y, timestep

ReDim Track_pos(1 To Track_points, 1) 'Track points, 0 for distance along length of aircraft, and
1 for distance perpindicular to length

ReDim Drag(1 To drag_points, 1, 0 To data_points) 'drag point, 0 for x and  for y, timestep

ReDim drag_pos(1 To drag_points, 1) 'drag points,0 for distance along length of aircraft, and 1
for distance perpindicular to length

ReDim Track_posIT(1 To Track_points, 1) 'Track points, 0 for distance along length of aircraft,
and 1 for distance perpindicular to length

ReDim TrackIT(1 To Track_points, 1)   'Track points, 0 for distance along length of aircraft, and
1 for distance perpindicular to length

End If

'Generating input values

NGA_RAD = Sheet1.Cells(2, 9).Value * pi / 180 'Nose gear angle in radians

Sheet1.Cells(3, 9).Value = NGA_RAD

'Calculating Initial angles

Alpha1 = Atn((Sheet1.Cells(6, 7).Value) / (Sheet1.Cells(6, 8).Value))

'Read initial values of input

For i = 0 To 1

   Drive(i, 0) = Sheet1.Cells(6, 3 + i).Value

   Trace(i, 0) = Sheet1.Cells(6, 5 + i).Value

   Drag1(i, 0) = Sheet1.Cells(6, 7 + i).Value

   Drag2(i, 0) = Sheet1.Cells(6, 9 + i).Value

   For j = 1 To Track_points

      Track(j, i, 0) = Sheet1.Cells(6, 9 + i + 2 * j).Value

   Next j

   For k = 1 To drag_points
```

```
        Drag(k, i, 0) = Sheet1.Cells(6, 5 + i + 2 * k)
        Next k
    wing_cen(i, 0) = Sheet1.Cells(6, 2 * Track_points + 13 + i).Value
    tail_cen(i, 0) = Sheet1.Cells(6, 2 * Track_points + 15 + i).Value
  Next i
If NGA_RAD = 0 Then
Sheet1.Cells(7, 5).Value = Sheet1.Cells(6, 5).Value 'trace x
Sheet1.Cells(7, 6).Value = Sheet1.Cells(6, 6).Value 'trace Y
Sheet1.Cells(7, 7).Value = Sheet1.Cells(6, 7).Value 'drag1 x
Sheet1.Cells(7, 8).Value = Sheet1.Cells(6, 8).Value 'drag1 y
Sheet1.Cells(7, 9).Value = Sheet1.Cells(6, 9).Value 'drag2 x
Sheet1.Cells(7, 10).Value = Sheet1.Cells(6, 10).Value 'drag2 y
    For i = 1 To Track_points
    Sheet1.Cells(7, 2 * i + 9).Value = Sheet1.Cells(6, 2 * i + 9).Value 'Track points x
    Sheet1.Cells(7, 2 * i + 10).Value = Sheet1.Cells(6, 2 * i + 10).Value 'Track points y
    Next i
Sheet1.Cells(7, 2 * Track_points + 13).Value = Sheet1.Cells(6, 2 * Track_points + 13).Value
'wing center x
Sheet1.Cells(7, 2 * Track_points + 14).Value = Sheet1.Cells(6, 2 * Track_points + 14).Value
'wing center y
Sheet1.Cells(7, 2 * Track_points + 15).Value = Sheet1.Cells(6, 2 * Track_points + 15).Value 'tail
center x
Sheet1.Cells(7, 2 * Track_points + 16).Value = Sheet1.Cells(6, 2 * Track_points + 16).Value 'tail
center y
ElseIf NGA_RAD > 0 Then
Sheet1.Cells(7, 5).Value = Sqr(((Sheet1.Cells(6, 5).Value) ^ 2) + ((Sheet1.Cells(6, 6).Value) ^
2)) * Sin(NGA_RAD) 'Trace point X
Sheet1.Cells(7, 6).Value = Sqr(((Sheet1.Cells(6, 5).Value) ^ 2) + ((Sheet1.Cells(6, 6).Value) ^
2)) * Cos(NGA_RAD) 'trace point Y
Sheet1.Cells(7, 7).Value = ((Sheet1.Cells(6, 7).Value) - (Sheet1.Cells(6, 5).Value)) *
Cos(NGA_RAD) + (Sheet1.Cells(7, 5).Value) 'Drag point1 X
```

```
Sheet1.Cells(7, 8).Value = (((Sheet1.Cells(6, 7).Value) - (Sheet1.Cells(6, 5).Value)) * -1 *
Sin(NGA_RAD)) + (Sheet1.Cells(7, 6).Value) 'Drag point1 Y

Sheet1.Cells(7, 9).Value = (Sheet1.Cells(7, 5).Value) - (Abs(Sheet1.Cells(6, 9).Value) -
(Sheet1.Cells(6, 5).Value)) * Cos(NGA_RAD) 'Drag point2 X

Sheet1.Cells(7, 10).Value = (Sheet1.Cells(7, 6).Value) + (Abs((Sheet1.Cells(6, 9).Value) -
(Sheet1.Cells(6, 5).Value)) * Sin(NGA_RAD)) 'Drag point2 Y

Sheet1.Cells(7, 2 * Track_points + 13).Value = Sqr(((Sheet1.Cells(6, 2 * Track_points +
13).Value) ^ 2) + ((Sheet1.Cells(6, 2 * Track_points + 14).Value) ^ 2)) * Sin(NGA_RAD) 'Wing
center X

Sheet1.Cells(7, 2 * Track_points + 14).Value = Sqr(((Sheet1.Cells(6, 2 * Track_points +
13).Value) ^ 2) + ((Sheet1.Cells(6, 2 * Track_points + 14).Value) ^ 2)) * Cos(NGA_RAD) 'Wing
center Y

Sheet1.Cells(7, 2 * Track_points + 15).Value = Sqr(((Sheet1.Cells(6, 2 * Track_points +
15).Value) ^ 2) + ((Sheet1.Cells(6, 2 * Track_points + 16).Value) ^ 2)) * Sin(NGA_RAD) 'Tail
center X

Sheet1.Cells(7, 2 * Track_points + 16).Value = Sqr(((Sheet1.Cells(6, 2 * Track_points +
15).Value) ^ 2) + ((Sheet1.Cells(6, 2 * Track_points + 16).Value) ^ 2)) * Cos(NGA_RAD) 'Tail
center Y

'Finds Initial position of track points relative to Drive

For i = 1 To Track_points

   Length_IT = ((Track(i, 0, 0) - Drive(0, 0)) ^ 2 + ((Track(i, 1, 0) - Drive(1, 0)) ^ 2)) ^ (1 / 2)

      If Track(i, 0, 0) <> Drive(0, 0) Then   'prevents error from using the atn function at pi/2 and -
pi/2 radians

         phi_IT = Atn((Track(i, 1, 0) - Drive(1, 0)) / (Track(i, 0, 0) - Drive(0, 0)))

         If Track(i, 0, 0) < Drive(0, 0) And Track(i, 1, 0) >= Drive(1, 0) Then  'these statments are
to cause angle to be of range -pi to +pi, rather than -pi/2 to pi/2

            phi_IT = phi_IT + pi

         ElseIf Track(i, 0, 0) < Drive(0, 0) And Track(i, 1, 0) < Drive(1, 0) Then

            phi_IT = phi_IT - pi

         End If

      ElseIf Track(i, 1, 0) > Drive(1, 0) Then    'determines direction if path is vertical

         phi_IT = pi / 2

      Else: phi_IT = -1 * pi / 2

      End If
```

```
    TrackIT(i, 0) = Length_IT * Cos(phi_IT - NGA_RAD)
    TrackIT(i, 1) = Length_IT * Sin(phi_IT - NGA_RAD)
    Sheet1.Cells(7, 9 + 2 * i).Value = TrackIT(i, 0)
    Sheet1.Cells(7, 10 + 2 * i).Value = TrackIT(i, 1)
Next i
End If
If NGA_RAD < 0 Then
MsgBox "Nose Gear Angle Should be Positive (0 to 360 degrees)"
End If
'Read initial values after nose gear orientation
For i = 0 To 1
    Drive(i, 0) = Sheet1.Cells(7, 3 + i).Value
    Trace(i, 0) = Sheet1.Cells(7, 5 + i).Value
    Drag1(i, 0) = Sheet1.Cells(7, 7 + i).Value
    Drag2(i, 0) = Sheet1.Cells(7, 9 + i).Value
    For j = 1 To Track_points
        Track(j, i, 0) = Sheet1.Cells(7, 9 + i + 2 * j).Value
    Next j
    For k = 1 To drag_points
        Drag(k, i, 0) = Sheet1.Cells(7, 5 + i + 2 * k)
        Next k
    wing_cen(i, 0) = Sheet1.Cells(7, 2 * Track_points + 13 + i).Value
    tail_cen(i, 0) = Sheet1.Cells(7, 2 * Track_points + 15 + i).Value
  Next i
 'finds initial orientation
If Drive(0, 0) <> Trace(0, 0) Then 'prevents error from using the atn function at pi/2 and -pi/2
radians
```

```
If Drive(0, 0) > Trace(0, 0) Then    'these statments are to cause angle to be of range -pi to +pi, rather than -pi/2 to pi/2
      theta = Atn((Drive(1, 0) - Trace(1, 0)) / (Drive(0, 0) - Trace(0, 0)))
   ElseIf Drive(1, 0) > Trace(1, 0) Then
      theta = Atn((Drive(1, 0) - Trace(1, 0)) / (Drive(0, 0) - Trace(0, 0))) + pi
   Else: theta = Atn((Drive(1, 0) - Trace(1, 0)) / (Drive(0, 0) - Trace(0, 0))) - pi
   End If
ElseIf Drive(1, 0) > Trace(1, 0) Then 'determines direction if path is vertical
   theta = pi / 2
Else: theta = -pi / 2
End If
'//END OF READING INITIAL VALUES
'/CLACULATING ALGLES
'reads the values of drive points
For i = 1 To data_points
   For j = 0 To 1
   Drive(j, i) = Sheet1.Cells(7 + i, 3 + j).Value
   Next j
Next i
'Finds distance between drive and trace points
Length_DTCE = ((Drive(1, 0) - Trace(1, 0)) ^ 2 + (Drive(0, 0) - Trace(0, 0)) ^ 2) ^ (1 / 2)
'Finds position of drag points relative to Drive and Trace points
For i = 1 To drag_points
Length_DD1 = ((Drag(i, 0, 0) - Drive(0, 0)) ^ 2 + (Drag(i, 1, 0) - Drive(1, 0)) ^ 2) ^ (1 / 2)
If Drag(i, 0, 0) > Drive(0, 0) Then  'prevents error from using the atn function at pi/2 and -pi/2 radians
      phi_D1 = Atn((Drag(i, 1, 0) - Drive(1, 0)) / (Drag(i, 0, 0) - Drive(0, 0)))
ElseIf Drag(i, 0, 0) < Drive(0, 0) Then  'prevents error from using the atn function at pi/2 and -pi/2 radians
      phi_D1 = -1 * Atn((Drag(i, 1, 0) - Drive(1, 0)) / (Drag(i, 0, 0) - Drive(0, 0)))
```

```
 If Drag(i, 0, 0) < Drive(0, 0) And Drag(i, 1, 0) >= Drive(1, 0) Then  'these statments are to cause angle to be of range -pi to +pi, rather than -pi/2 to pi/2
            phi_D1 = phi_D1 + pi
         ElseIf Drag(i, 0, 0) < Drive(0, 0) And Drag(i, 1, 0) < Drive(1, 0) Then
            phi = phi_D1 - pi
         End If
      ElseIf Drag(i, 1, 0) > Drive(1, 0) Then    'determines direction if path is vertical
         phi_D1 = pi / 2
      Else: phi_D1 = -1 * pi / 2
      End If
If Drag(i, 0, 0) > Drive(0, 0) And Drag(i, 1, 0) > Drive(1, 0) Then
   drag_pos(i, 0) = -Length_DD1 * Sin(phi_D1 - theta)
   drag_pos(i, 1) = Length_DD1 * Cos(phi_D1 - theta)
   ElseIf Drag(i, 0, 0) > Drive(0, 0) And Drag(i, 1, 0) < Drive(1, 0) Then
   drag_pos(i, 0) = -Length_DD1 * Sin(phi_D1 - theta)
   drag_pos(i, 1) = Length_DD1 * Cos(phi_D1 - theta)
   ElseIf Drag(i, 0, 0) < Drive(0, 0) And Drag(i, 1, 0) > Drive(1, 0) Then
   drag_pos(i, 0) = -Length_DD1 * Sin(-phi_D1 - theta)
   drag_pos(i, 1) = Length_DD1 * Cos(phi_D1 + theta)
   ElseIf Drag(i, 0, 0) < Drive(0, 0) And Drag(i, 1, 0) < Drive(1, 0) Then
   drag_pos(i, 0) = -Length_DD1 * Sin(phi_D1 + theta)
   drag_pos(i, 1) = -Length_DD1 * Cos(-phi_D1 - theta)
   End If
Next i
'Finds position of track points relative to Drive and Drag points
For i = 1 To Track_points
   Length_T = ((Track(i, 0, 0) - Drive(0, 0)) ^ 2 + (Track(i, 1, 0) - Drive(1, 0)) ^ 2) ^ (1 / 2)
      If Track(i, 0, 0) <> Drive(0, 0) Then   'prevents error from using the atn function at pi/2 and -pi/2 radians
```

```
        phi = Atn((Track(i, 1, 0) - Drive(1, 0)) / (Track(i, 0, 0) - Drive(0, 0)))
If Track(i, 0, 0) < Drive(0, 0) And Track(i, 1, 0) >= Drive(1, 0) Then  'these statments are to cause angle to be of range -pi to +pi, rather than -pi/2 to pi/2
            phi = phi + pi
        ElseIf Track(i, 0, 0) < Drive(0, 0) And Track(i, 1, 0) < Drive(1, 0) Then
            phi = phi - pi
        End If
    ElseIf Track(i, 1, 0) > Drive(1, 0) Then    'determines direction if path is vertical
        phi = pi / 2
    Else: phi = -1 * pi / 2
    End If
  Track_pos(i, 0) = -Length_T * Sin(phi - theta)
  Track_pos(i, 1) = Length_T * Cos(phi - theta)
Next i
'\\ Calculation of new positions
'Finds orientation after rotating about the previous trace point to point at new drive point
For i = 1 To data_points
  If Drive(0, i) <> Trace(0, i - 1) Then
    If Drive(0, i) > Trace(0, i - 1) Then
      theta = Atn((Drive(1, i) - Trace(1, i - 1)) / (Drive(0, i) - Trace(0, i - 1)))
    ElseIf Drive(1, i) > Trace(1, i - 1) Then
      theta = Atn((Drive(1, i) - Trace(1, i - 1)) / (Drive(0, i) - Trace(0, i - 1))) + pi
    Else: theta = Atn((Drive(1, i) - Trace(1, i - 1)) / (Drive(0, i) - Trace(0, i - 1))) - pi
  End If
  ElseIf Drive(1, i) > Trace(1, i - 1) Then
    theta = pi / 2
  Else: theta = -pi / 2
  End If
```

```
'Finding new position of trace point
Trace(0, i) = Drive(0, i) - (Length_DTCE * Cos(theta))
Trace(1, i) = Drive(1, i) - (Length_DTCE * Sin(theta))
'Prints new position of trace point
Sheet1.Cells(7 + i, 5).Value = Trace(0, i)
Sheet1.Cells(7 + i, 6).Value = Trace(1, i)
'Finds orientation after rotating about the previous wing center point to point at new drive point
   'If Drive(0, i) <> wing_cen(0, i - 1) Then
      'If Drive(0, i) > wing_cen(0, i - 1) Then
         'theta_wc = Atn((Drive(1, i) - wing_cen(1, i - 1)) / (Drive(0, i) - wing_cen(0, i - 1)))
      'ElseIf Drive(1, i) > wing_cen(1, i - 1) Then
         'theta_wc = Atn((Drive(1, i) - wing_cen(1, i - 1)) / (Drive(0, i) - wing_cen(0, i - 1))) + pi
    'Else: theta_wc = Atn((Drive(1, i) - wing_cen(1, i - 1)) / (Drive(0, i) - wing_cen(0, i - 1))) - pi
 'End If
   'ElseIf Drive(1, i) > wing_cen(1, i - 1) Then
      'theta_wc = pi / 2
   'Else: theta_wc = -pi / 2
   'End If
'Finding new position of wing center point
Length_wc = ((Drive(1, 0) - wing_cen(1, 0)) ^ 2 + (Drive(0, 0) - wing_cen(0, 0)) ^ 2) ^ (1 / 2)
wing_cen(0, i) = Drive(0, i) - (Length_wc * Cos(theta))
wing_cen(1, i) = Drive(1, i) - (Length_wc * Sin(theta))
'Prints new position of wing center point
Sheet1.Cells(7 + i, 2 * Track_points + 13).Value = wing_cen(0, i)
Sheet1.Cells(7 + i, 2 * Track_points + 14).Value = wing_cen(1, i)
'Finds orientation after rotating about the previous tail center point to point at new drive point
   'If Drive(0, i) <> tail_cen(0, i - 1) Then
      'If Drive(0, i) > tail_cen(0, i - 1) Then
         'theta_tc = Atn((Drive(1, i) - tail_cen(1, i - 1)) / (Drive(0, i) - tail_cen(0, i - 1)))
```

```
        'ElseIf Drive(1, i) > tail_cen(1, i - 1) Then
            'theta_tc = Atn((Drive(1, i) - tail_cen(1, i - 1)) / (Drive(0, i) - tail_cen(0, i - 1))) + pi
        'Else: theta_tc = Atn((Drive(1, i) - tail_cen(1, i - 1)) / (Drive(0, i) - tail_cen(0, i - 1))) - pi
    'End If
    'ElseIf Drive(1, i) > tail_cen(1, i - 1) Then
        'theta_tc = pi / 2
    'Else: theta_tc = -pi / 2
    'End If
'Finding new position of tail center point
Length_tc = ((Drive(1, 0) - tail_cen(1, 0)) ^ 2 + (Drive(0, 0) - tail_cen(0, 0)) ^ 2) ^ (1 / 2)
tail_cen(0, i) = Drive(0, i) - (Length_tc * Cos(theta))
tail_cen(1, i) = Drive(1, i) - (Length_tc * Sin(theta))
'Prints new position of tail center point
Sheet1.Cells(7 + i, 2 * Track_points + 15).Value = tail_cen(0, i)
Sheet1.Cells(7 + i, 2 * Track_points + 16).Value = tail_cen(1, i)
'finds new drag point positions
    For k = 1 To drag_points
        Drag(k, 0, i) = drag_pos(k, 0) * Sin(theta) + drag_pos(k, 1) * Cos(theta) + Drive(0, i)
        Drag(k, 1, i) = -1 * drag_pos(k, 0) * Cos(theta) + drag_pos(k, 1) * Sin(theta) + Drive(1, i)
'Prints new position of drag points
        Sheet1.Cells(7 + i, 5 + 2 * k).Value = Drag(k, 0, i)
        Sheet1.Cells(7 + i, 6 + 2 * k).Value = Drag(k, 1, i)
    Next k
 ‘finds new track point positions
    For j = 1 To Track_points
        Track(j, 0, i) = Track_pos(j, 0) * Sin(theta) + Track_pos(j, 1) * Cos(theta) + Drive(0, i)
        Track(j, 1, i) = -1 * Track_pos(j, 0) * Cos(theta) + Track_pos(j, 1) * Sin(theta) + Drive(1, i)
'Prints new position of track points
        Sheet1.Cells(7 + i, 9 + 2 * j).Value = Track(j, 0, i)
```

```
        Sheet1.Cells(7 + i, 10 + 2 * j).Value = Track(j, 1, i)
    Next j
'Finds orientation of the drive path
    If Drive(0, i) <> Drive(0, i - 1) Then  'prevents error from using the atn function at pi/2 and -pi/2 radians
If Drive(0, i) > Drive(0, i - 1) Then   'these statments are to cause angle to be of range -pi to +pi, rather than -pi/2 to pi/2
            p_theta = Atn((Drive(1, i) - Drive(1, i - 1)) / (Drive(0, i) - Drive(0, i - 1)))
        ElseIf Drive(1, i) > Drive(1, i - 1) Then
            p_theta = Atn((Drive(1, i) - Drive(1, i - 1)) / (Drive(0, i) - Drive(0, i - 1))) + pi
        Else: p_theta = Atn((Drive(1, i) - Drive(1, i - 1)) / (Drive(0, i) - Drive(0, i - 1))) - pi
        End If
    ElseIf Drive(1, i) > Drive(1, i - 1) Then   'determines direction if path is vertical
        p_theta = pi / 2
    Else: p_theta = -1 * pi / 2
    End If
 'Finds difference between objects orientation and drive path in degrees
    dtheta = (p_theta - theta) * 180 / pi
    If dtheta > 180 Then dtheta = dtheta - 360
    If dtheta < -180 Then dtheta = dtheta + 360
'Prints the value of dtheta
        Sheet1.Cells(7 + i, 2 * Track_points + 11).Value = dtheta
 Next i
'Graph preparation
'finds extreme values for graph
For i = 0 To data_points
    For j = 0 To 1
     If Drive(j, i) < min(j) Then
            min(j) = Drive(j, i)
```

```
        ElseIf Drive(j, i) > max(j) Then
            max(j) = Drive(j, i)
        End If
        If Trace(j, i) < min(j) Then
            min(j) = Trace(j, i)
        ElseIf Trace(j, i) > max(j) Then
            max(j) = Trace(j, i)
        End If
    If Drag1(j, i) < min(j) Then
            min(j) = Drag1(j, i)
        ElseIf Drag1(j, i) > max(j) Then
            max(j) = Drag1(j, i)
        End If
      If Drag2(j, i) < min(j) Then
            min(j) = Drag2(j, i)
        ElseIf Drag2(j, i) > max(j) Then
            max(j) = Drag2(j, i)
        End If
    For k = 1 To Track_points
            If Track(k, j, i) < min(j) Then
                min(j) = Track(k, j, i)
            ElseIf Track(k, j, i) > max(j) Then
                max(j) = Track(k, j, i)
            End If
        Next k
    Next j
Next i
'causes x and y scale to be the same
If (max(0) - min(0)) > (max(1) - min(1)) Then
```

```
    max(1) = max(1) + (((max(0) - min(0)) - (max(1) - min(1))) / 2)
    min(1) = min(1) - ((max(0) - min(0)) - (max(1) - min(1)))
    Else
    max(0) = max(0) + (((max(1) - min(1)) - (max(0) - min(0))) / 2)
    min(0) = min(0) - ((max(1) - min(1)) - (max(0) - min(0)))
End If
'rounds values to more extreme value of 10
For j = 0 To 1
    min(j) = min(j) - 250 + Abs(min(j) Mod 10)
    max(j) = max(j) + 10 - Abs(max(j) Mod 10)
Next j
'runs code to graph the results
Call Graph_it(min(), max())
End Sub
Sub Graph_it(min() As Integer, max() As Integer)
Dim cht As ChartObject  'the area on the sheet that is reserved for the graph
Dim ct As Chart         'the graph itself which is inside the chartobject
Dim location(0 To 1) As Integer '0 is row, 1 is colum
Dim data_points As Integer  'Number of data points used to calculate each path
Dim show_points As Integer  'Number of data points being shown on the graph per path
Dim Track_points As Integer 'number of points on the aircraft that are being tracked, not
counting the drive and drag points
Dim chart_rng As Range      'the range of the axis on the graph
Dim series_col As SeriesCollection  'list of series contained in the graph
Dim Drive_point As Series   'series of data points for the drive path that is used to create the line
Dim drag1_point As Series    'series of data points for the drag1 path that is used to create the line
Dim drag2_point As Series    'series of data points for the drag2 path that is used to create the line
Dim track_value() As Series 'series of data points for the track paths that is used to create the line
```

```
Dim connecting_lines1() As Series  'series that creates lines connecting the points when markers are shown

Dim gridlines() As Series  'series that defines where gridlines will be

Dim gridline_space As Integer 'spacing between gridlines

Dim gridline_count As Integer   'number of gridlines

Dim i As Integer

Dim j As Integer

Dim k As Integer

Dim trace_value As Series  'series of data points for the track paths that is used to create the points on the line

Dim connecting_lines2() As Series  'series that creates lines connecting the points when markers are shown

Dim connecting_lines3() As Series  'series that creates lines connecting the points when markers are shown

Dim connecting_lines4() As Series  'series that creates lines connecting the points when markers are shown

Dim connecting_lines5() As Series  'series that creates lines connecting the points when markers are shown

Dim Nose_gear() As Series  'series that creates lines connecting the points when markers are shown

Dim wing_cen As Series 'Intersection point of wings at aircraft axis

Dim tail_cen As Series 'Intersection point of tails at aircraft axis

Dim Aircraft_step As Integer 'Number of aircraft picture

Dim drag1_tyre() As Series 'Showing drag1 tire location

Dim drag2_tyre() As Series 'Showing drag2 tire location

Dim adj_lineR As Series 'Creating block for adjacent aircraft at right

Dim adj_lineL As Series 'Creating block for adjacent aircraft at Left

Dim seperation_distance As Integer

'deletes any existing graph

For Each cht In Sheet1.ChartObjects

   cht.Delete

Next
```

```
'inputs values

data_points = Sheet1.Cells(2, 3).Value

Track_points = Sheet1.Cells(2, 12).Value

Aircraft_step = data_points / 2

seperation_distance = 35

'determines chart top left corner

location(0) = 7

location(1) = 13 + 2 * Track_points

'defines range used to position chart

Set chart_rng = ActiveSheet.Range(Cells(location(0), location(1)), Cells(location(0) + 7, location(1) + 7))

'calculates gridline spacing, and resizes min and max so origin is on gridlines

gridline_count = 10

gridline_space = (max(0) - min(0)) / gridline_count 'sets spacing so there are 10 gridlines

gridline_space = gridline_space + (10 - gridline_space Mod 10) 'expands space so they appear at multibles of 50

min(0) = min(0) - (gridline_space + (min(0) Mod gridline_space))    'changes the limits of graph to end on gridlines

max(0) = max(0) + (gridline_space - (max(0) Mod gridline_space))

min(1) = min(1) - (gridline_space + (min(1) Mod gridline_space))

max(1) = max(1) + (gridline_space - (max(1) Mod gridline_space))

'defines the array size for the track points

If Track_points > 0 Then

   ReDim track_value(1 To Track_points)

End If

ReDim gridlines(gridline_count + 1, 1)

ReDim connecting_lines1(Track_points, data_points) '(0 is drive to drag; others are from track to drag)

ReDim connecting_lines2(Track_points, data_points) '(0 is drive to drag; others are from track to drag)
```

```
ReDim connecting_lines3(Track_points, data_points) '(0 is drive to drag; others are from track to drag)

ReDim connecting_lines4(Track_points, data_points) '(0 is drive to drag; others are from track to drag)

ReDim connecting_lines5(Track_points, data_points) '(0 is drive to drag; others are from track to drag)

ReDim Nose_gear(data_points) '(0 is drive to drag; others are from track to drag)

ReDim drag1_tyre(data_points) '(0 is drive to drag; others are from track to drag)

ReDim drag2_tyre(data_points) '(0 is drive to drag; others are from track to drag)

'Create Values for adjacent aircraft block-Right

Sheet1.Cells(6, 27).Value = Sheet1.Cells(6, 11).Value + seperation_distance

Sheet1.Cells(7, 27).Value = Sheet1.Cells(6, 11).Value + seperation_distance

Sheet1.Cells(6, 28).Value = Sheet1.Cells(6, 16).Value

Sheet1.Cells(7, 28).Value = Sheet1.Cells(6, 20).Value

Sheet1.Cells(8, 27).Value = Sheet1.Cells(6, 27).Value + (2 * (Sheet1.Cells(6, 11).Value))

Sheet1.Cells(9, 27).Value = Sheet1.Cells(6, 27).Value + (2 * (Sheet1.Cells(6, 11).Value))

Sheet1.Cells(8, 28).Value = Sheet1.Cells(6, 16).Value

Sheet1.Cells(9, 28).Value = Sheet1.Cells(6, 20).Value

Sheet1.Cells(10, 27).Value = Sheet1.Cells(6, 11).Value + seperation_distance

Sheet1.Cells(11, 27).Value = Sheet1.Cells(6, 27).Value + (2 * (Sheet1.Cells(6, 11).Value))

Sheet1.Cells(10, 28).Value = Sheet1.Cells(6, 16).Value

Sheet1.Cells(11, 28).Value = Sheet1.Cells(6, 16).Value

Sheet1.Cells(12, 27).Value = Sheet1.Cells(6, 11).Value + seperation_distance

Sheet1.Cells(13, 27).Value = Sheet1.Cells(6, 27).Value + (2 * (Sheet1.Cells(6, 11).Value))

Sheet1.Cells(12, 28).Value = Sheet1.Cells(6, 20).Value

Sheet1.Cells(13, 28).Value = Sheet1.Cells(6, 20).Value

'Create Values for adjacent aircraft block-Left

Sheet1.Cells(6, 29).Value = Sheet1.Cells(6, 13).Value - seperation_distance

Sheet1.Cells(7, 29).Value = Sheet1.Cells(6, 13).Value - seperation_distance

Sheet1.Cells(6, 30).Value = Sheet1.Cells(6, 16).Value
```

```
Sheet1.Cells(7, 30).Value = Sheet1.Cells(6, 20).Value
Sheet1.Cells(8, 29).Value = Sheet1.Cells(6, 29).Value - (2 * (Sheet1.Cells(6, 11).Value))
Sheet1.Cells(9, 29).Value = Sheet1.Cells(6, 29).Value - (2 * (Sheet1.Cells(6, 11).Value))
Sheet1.Cells(8, 30).Value = Sheet1.Cells(6, 16).Value
Sheet1.Cells(9, 30).Value = Sheet1.Cells(6, 20).Value
Sheet1.Cells(10, 29).Value = Sheet1.Cells(6, 13).Value - seperation_distance
Sheet1.Cells(11, 29).Value = Sheet1.Cells(6, 29).Value - (2 * (Sheet1.Cells(6, 11).Value))
Sheet1.Cells(10, 30).Value = Sheet1.Cells(6, 16).Value
Sheet1.Cells(11, 30).Value = Sheet1.Cells(6, 16).Value
Sheet1.Cells(12, 29).Value = Sheet1.Cells(6, 13).Value - seperation_distance
Sheet1.Cells(13, 29).Value = Sheet1.Cells(6, 29).Value - (2 * (Sheet1.Cells(6, 11).Value))
Sheet1.Cells(12, 30).Value = Sheet1.Cells(6, 20).Value
Sheet1.Cells(13, 30).Value = Sheet1.Cells(6, 20).Value
'creates chart
Set cht = Sheet1.ChartObjects.Add(chart_rng.Left, chart_rng.Top, 500, 500)
Set ct = cht.Chart
With ct
  'determines chart layout
   .HasLegend = False
   '.Legend.Position = xlLegendPositionBottom
   .HasTitle = False
   Set series_col = ct.SeriesCollection
 'creates drive point line
   Set Drive_point = series_col.NewSeries
   With Drive_point
      .Name = "Drive"
      .XValues = ActiveSheet.Range(Cells(7, 3), Cells(7 + data_points, 3))
      .Values = Sheet1.Range(Cells(7, 4), Cells(7 + data_points, 4))
      .ChartType = xlXYScatterLinesNoMarkers
```

```
        .Format.Line.ForeColor.RGB = RGB(0, 0, 0) 'see note at bottom
    End With
  'creates trace point lines
        Set trace_value = series_col.NewSeries
        With trace_value
            .Name = "Trace value " & i
            .XValues = ActiveSheet.Range(Cells(7, 5), Cells(7 + data_points, 5))
            .Values = Sheet1.Range(Cells(7, 6), Cells(7 + data_points, 6))
            .ChartType = xlXYScatterLinesNoMarkers
            .Format.Line.DashStyle = msoLineDash
            .Format.Line.ForeColor.RGB = RGB(255, 69, 0) 'see note at bottom
        End With
     'creates drag1 point lines
        Set drag1_point = series_col.NewSeries
        With drag1_point
            .Name = "Drag1_point " & i
            .XValues = Sheet1.Range(Cells(7, 7), Cells(7 + data_points, 7))
            .Values = Sheet1.Range(Cells(7, 8), Cells(7 + data_points, 8))
            .ChartType = xlXYScatterLinesNoMarkers
            .Format.Line.ForeColor.RGB = RGB(0, 0, 255) 'see note at bottom
        End With
  'creates drag1 point lines
        Set drag2_point = series_col.NewSeries
        With drag2_point
            .Name = "drag2_point " & i
            .XValues = Sheet1.Range(Cells(7, 9), Cells(7 + data_points, 9))
            .Values = Sheet1.Range(Cells(7, 10), Cells(7 + data_points, 10))
            .ChartType = xlXYScatterLinesNoMarkers
            .Format.Line.ForeColor.RGB = RGB(0, 255, 0) 'see note at bottom
```

```
        End With
 'creates track point lines
    For i = 1 To Track_points
        Set track_value(i) = series_col.NewSeries
        With track_value(i)
            .Name = "Track point " & i
            .XValues = Sheet1.Range(Cells(7, 9 + 2 * i), Cells(7 + data_points, 9 + 2 * i))
            .Values = Sheet1.Range(Cells(7, 10 + 2 * i), Cells(7 + data_points, 10 + 2 * i))
            .ChartType = xlXYScatterLinesNoMarkers
            .Format.Line.ForeColor.RGB = RGB(Round(100 * (Sin(5 * i + 2) + 1)), Round(150 *
(Sin(3 * i + 2.5) + 1)), Round(200 * (Sin(2 * i + 2) + 1))) 'see note at bottom
        End With
    Next i
    'creates adjacent plane-Right
    Set adj_lineR = series_col.NewSeries
    With adj_lineR
        .Name = "adjacent lineR"
        .XValues = Sheet1.Range(Cells(6, 27), Cells(13, 27))
        .Values = Sheet1.Range(Cells(6, 28), Cells(13, 28))
        .ChartType = xlXYScatterLinesNoMarkers
        .Format.Line.ForeColor.RGB = RGB(0, 0, 0) 'see note at bottom
    End With
    'creates adjacent plane-Left
    Set adj_lineL = series_col.NewSeries
    With adj_lineL
        .Name = "adjacent lineL"
        .XValues = Sheet1.Range(Cells(6, 29), Cells(13, 29))
        .Values = Sheet1.Range(Cells(6, 30), Cells(13, 30))
        .ChartType = xlXYScatterLinesNoMarkers
```

```
        .Format.Line.ForeColor.RGB = RGB(0, 0, 0) 'see note at bottom
    End With
'creates wing center point lines
    'Set wing_cen = series_col.NewSeries
    'With wing_cen
        '.Name = "Wing_cen"
        '.XValues = Sheet1.Range(Cells(7, 2 * Track_points + 13), Cells(7 + data_points, 2 *
Track_points + 13))
        '.Values = Sheet1.Range(Cells(7, 2 * Track_points + 14), Cells(7 + data_points, 2 *
Track_points + 14))
        '.ChartType = xlXYScatterLinesNoMarkers
        '.Format.Line.ForeColor.RGB = RGB(255, 255, 0) 'see note at bottom
    'End With
'creates tail center point lines
    'Set tail_cen = series_col.NewSeries
    'With tail_cen
        '.Name = "tail_cen"
        '.XValues = Sheet1.Range(Cells(7, 2 * Track_points + 15), Cells(7 + data_points, 2 *
Track_points + 15))
        '.Values = Sheet1.Range(Cells(7, 2 * Track_points + 16), Cells(7 + data_points, 2 *
Track_points + 16))
        '.ChartType = xlXYScatterLinesNoMarkers
        '.Format.Line.ForeColor.RGB = RGB(255, 255, 0) 'see note at bottom
    'End With
    'creates line connecting drive and track points
    For i = 0 To data_points Step Aircraft_step
    'Track point 1
            Set connecting_lines1(0, i) = series_col.NewSeries
        With connecting_lines1(0, i
.XValues = Array(Sheet1.Cells(7 + i, 2 * Track_points + 13).Value, Sheet1.Cells(7 + i,11).Value)
```

```
.Values = Array(Sheet1.Cells(7 + i, 2 * Track_points + 14).Value, Sheet1.Cells(7 + i, 12).Value)
            .ChartType = xlXYScatterLinesNoMarkers
            .Format.Line.Weight = 3
            .Format.Line.ForeColor.RGB = RGB(128, 128, 0)
        End With
        'Track point 2
          Set connecting_lines2(0, i) = series_col.NewSeries
        With connecting_lines2(0, i)
.XValues = Array(Sheet1.Cells(7 + i, 2 * Track_points + 13).Value, Sheet1.Cells(7 + i,13).Value)
.Values = Array(Sheet1.Cells(7 + i, 2 * Track_points + 14).Value, Sheet1.Cells(7 + i, 14).Value)
            .ChartType = xlXYScatterLinesNoMarkers
            .Format.Line.Weight = 3
            .Format.Line.ForeColor.RGB = RGB(128, 128, 0)
        End With
        'Track point 3
          Set connecting_lines3(0, i) = series_col.NewSeries
        With connecting_lines3(0, i)
.XValues = Array(Sheet1.Cells(7 + i, 2 * Track_points + 15).Value, Sheet1.Cells(7 + i,15).Value)
 .Values = Array(Sheet1.Cells(7 + i, 2 * Track_points + 16).Value, Sheet1.Cells(7 + i, 16).Value)
            .ChartType = xlXYScatterLinesNoMarkers
            .Format.Line.Weight = 3
            .Format.Line.ForeColor.RGB = RGB(128, 128, 0)
        End With
        'Track point 4
          Set connecting_lines4(0, i) = series_col.NewSeries
        With connecting_lines4(0, i)
.XValues = Array(Sheet1.Cells(7 + i, 2 * Track_points + 15).Value, Sheet1.Cells(7 + i,17).Value)
 .Values = Array(Sheet1.Cells(7 + i, 2 * Track_points + 16).Value, Sheet1.Cells(7 + i, 18).Value)
            .ChartType = xlXYScatterLinesNoMarkers
```

```
            .Format.Line.Weight = 3
            .Format.Line.ForeColor.RGB = RGB(128, 128, 0)
        End With
        'Track point 5
          Set connecting_lines5(0, i) = series_col.NewSeries
        With connecting_lines5(0, i)
.XValues = Array(Sheet1.Cells(7 + i, 2 * Track_points + 15).Value, Sheet1.Cells(7 + i,19).Value)
 .Values = Array(Sheet1.Cells(7 + i, 2 * Track_points + 16).Value, Sheet1.Cells(7 + i, 20).Value)
            .ChartType = xlXYScatterLinesNoMarkers
            .Format.Line.Weight = 3
            .Format.Line.ForeColor.RGB = RGB(128, 128, 0)
        End With
        'Nose gear point
           Set Nose_gear(i) = series_col.NewSeries
        With Nose_gear(i)
            .XValues = Array(Sheet1.Cells(7 + i, 3).Value)
            .Values = Array(Sheet1.Cells(7 + i, 4).Value)
            .Format.Fill.ForeColor.RGB = RGB(255, 0, 0)
            .MarkerForegroundColor = RGB(255, 0, 0)
            .MarkerStyle = xlMarkerStyleCircle
            .MarkerSize = 8
        End With
        'drag1_point point
           Set drag1_tyre(i) = series_col.NewSeries
        With drag1_tyre(i)
            .XValues = Array(Sheet1.Cells(7 + i, 7).Value)
            .Values = Array(Sheet1.Cells(7 + i, 8).Value)
            .Format.Fill.ForeColor.RGB = RGB(0, 0, 255)
            .MarkerForegroundColor = RGB(0, 0, 255)
```

```
            .MarkerStyle = xlMarkerStyleCircle
            .MarkerSize = 8
        End With
        'drag2_point Point
            Set drag2_tyre(i) = series_col.NewSeries
        With drag2_tyre(i)
            .XValues = Array(Sheet1.Cells(7 + i, 9).Value)
            .Values = Array(Sheet1.Cells(7 + i, 10).Value)
            .Format.Fill.ForeColor.RGB = RGB(0, 0, 255)
            .MarkerForegroundColor = RGB(0, 0, 255)
            .MarkerStyle = xlMarkerStyleCircle
            .MarkerSize = 8
        End With
    Next i
    'defines axis limits and creates gridlines for graph
    .Axes(xlCategory).MinimumScale = min(0)
    .Axes(xlCategory).MaximumScale = max(0)
    .Axes(xlValue).MinimumScale = min(1)
    .Axes(xlValue).MaximumScale = max(1)
    .Axes(xlValue).MajorUnit = gridline_space
    .Axes(xlCategory).MajorUnit = gridline_space
    .Axes(xlCategory).TickLabelPosition = xlTickLabelPositionLow
    .Axes(xlValue).TickLabelPosition = xlTickLabelPositionLow
    .Axes(xlCategory).HasMajorGridlines = True
    .Axes(xlValue).HasMajorGridlines = True
End With
End Sub
```

VITA

KARTHIK KUMAR RAJENDRAN

Candidate for the Degree of

Master of Science

Thesis: FIXED PATH PULL-IN/PUSHBACK TRAJECTORIES FOR AIRLINER GROUND TRANSPORT

Major Field: MECHANICAL AND AEROSPACE ENGINEERING

Biographical:

Education:

Completed the requirements for the Master of Science in Mechanical and Aerospace Engineering at Oklahoma State University, Stillwater, Oklahoma in May, 2019.

Completed the requirements for the Bachelor of Engineering in Mechanical Engineering at Anna University, Chennai, Tamilnadu, India in 2010.