SINGLE EVENT COMPARISONS OF PREDICTED AND MEASURED SOUND
AT VANCOUVER INTERNATIONAL AIRPORT

A Thesis in
Acoustics
by
Manasi Biwalkar

© 2017 Manasi Biwalkar

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2017
The thesis of Manasi Biwalkar was reviewed and approved* by the following:

Victor Sparrow  
Professor of Acoustics  
Director of the Graduate Program in Acoustics  
Thesis Advisor

Michelle Vigeant  
Assistant Professor of Acoustics and Architectural Engineering

Daniel Russell  
Professor of Acoustics

*Signatures are on file in the Graduate School.
Abstract

A recent project in the ASCENT Center of Excellence at Penn State and Purdue University is focused on validating aircraft noise models and quantifying uncertainties of both model prediction and measurement. Penn State’s contribution leans towards assessing the uncertainties in airport noise predictions due to consideration of a real meteorological data versus a vertically homogeneous atmosphere. The real atmosphere is observed to have a wind and temperature gradient which affects the sound speed profile. Refraction causes bending of sound rays and potentially affects the total intensity of the sound received at a particular location on the ground. This effect was analyzed using the NORD2000 outdoor sound propagation model. A MATLAB code was written to run the NORD2000 propagation code for three different sound speed profiles such as the homogeneous, linear and logarithmic sound speed profiles. This thesis compares simulation results obtained from the three cases in the NORD2000 code and an in-house propagation code with the sound pressure levels obtained at noise monitors. The airport considered was Vancouver International Airport in British Columbia, Canada. The comparisons with measured data help in understanding the difference in results computed using the NORD2000 propagation code when real atmospheric data was considered to calculate a sound speed profile as compared to a vertically homogeneous sound speed profile. It is shown that an improvement in predicted levels is more likely if the meteorological conditions are considered.
# Table of Contents

List of Figures vii

List of Tables ix

Acknowledgments x

## Chapter 1
**Motivation & Objectives** 1
  1.1 Motivation for aircraft research 1
  1.1.1 Developments in past decades and future 2
  1.1.2 FAA’s efforts towards aviation research 2
  1.1.3 Objectives 3
  1.2 Outline of this thesis 3

## Chapter 2
**Outdoor Sound Propagation and NORD2000 model** 4
  2.1 Outdoor sound propagation 4
  2.1.1 Spherical source in a free field 4
  2.1.2 Spherical spreading and inverse square law 5
  2.1.3 Atmospheric attenuation 6
  2.1.4 Refraction of sound 7
  2.2 Aviation Environmental Design Tool 9
  2.2.1 Computation in AEDT 9
  2.2.2 Study database 11
  2.2.3 Airport database 11
  2.2.4 Fleet database 11
  2.2.5 Weather model in AEDT 12
  2.3 Difference between real and homogeneous atmosphere 12
  2.4 NORD2000 propagation model 13
  2.5 Summary 15

## Chapter 3
**Datasets used for calculations** 16
  3.1 DISCOVER-AQ dataset 16
  3.2 BANOERAC, NINHA and SILENCE(R) datasets 17
  3.3 Vancouver International Airport 17
Chapter 4
Process input data
4.1 Introduction .......................... 20
4.2 Radar data .............................. 20
  4.2.1 Data format .......................... 20
  4.2.2 Available information & extraction of necessary data .............. 21
4.3 Noise Monitor Terminal (NMT) dataset ............................. 22
4.4 Spectral class data and NPD curves ................................. 23
4.5 Weather data ................................ 28
  4.5.1 Extracting necessary data from CFSR dataset .................... 29
4.6 Summary .................................. 31

Chapter 5
Calculation Methodology .......................... 32
5.1 Progression of methodology .......................... 32
  5.1.1 Phase 1 – Source level data and weather data input were not used ........................................ 32
  5.1.2 Phase 2 – Retrofitted input spectrum for B77L .................................. 33
  5.1.3 Phase 3 – Appropriate source level data and weather data were used .................................. 33
    5.1.3.1 Section 1 – Import input files .................................. 35
    5.1.3.2 Section 2 – RDConstruct .................................. 35
    5.1.3.3 Section 3 – NGConstruct .................................. 35
    5.1.3.4 Section 4 – Event selection .................................. 36
    5.1.3.5 Sections 5 & 6 – Analyzing events .......................... 37
5.2 Summary .................................. 44

Chapter 6
Results .................................................. 45
6.1 Comparison of results: In-house code and NORD2000 .......................... 45
6.2 Comparison of results: NORD2000 cases and measured data ............... 46
  6.2.1 Variation in results due to elevation cutoff .......................... 46
  6.2.2 Effect of NMT location topography on results ....................... 48
6.3 Discussion of results with examples .................................. 49
  6.3.1 Approach events on July 1, 2014 at 26L (south runway) triggering 4 NMTs .................................. 49
  6.3.2 Approach events on July 1, 2014 at 26R (north runway) triggering 6 NMTs .................................. 50
  6.3.3 Departure event on September 27, 2014 at 08R (south runway) ........ 52
6.4 Overall observations .................................. 54

Chapter 7
Conclusions .................................................. 56
7.1 Conclusions .................................. 56
7.2 Caveats and assumptions .................................. 56
7.3 Future work .................................. 57

Appendix A
Accessing the AEDT database using SQL .................................. 59
A.1 Accessing database tables .................................. 59
A.2 Define radar data rack in AEDT .................................. 60
## List of Figures

1.1 Historical trends in aviation [1] ................................. 1  
2.1 Spherical source in a free field [2] ................................. 5  
2.2 Log-log plot of sound absorption coefficient versus frequency for sound in air at 20°C, 1 atm pressure and with 20% relative humidity [3]. ................................. 7  
2.3 Effect of temperature gradient on sound speed [2]. In this figure, the absolute temperature is denoted by T and the height is denoted by H. ................................. 8  
2.4 Refraction of sound due to a wind profile [2] ................................. 9  
2.5 Schematic of AEDT’s background processes. ................................. 10  
2.6 Schematic of AEDT’s background processes. ................................. 12  
3.1 Main terminal at Vancouver International Airport (YVR) ................................. 18  
3.2 Map of YVR airport. X and Y axes are the longitude and latitude coordinates. ................................. 18  
3.3 Network of Noise Monitor Terminals (NMTs) around YVR. X and Y axes are the longitude and latitude coordinates. ................................. 19  
4.1 Selection of time duration for an aircraft event. t₁ and t₂ are the 10 dB down points from L₁max ................................. 23  
4.2 Spectral source levels for four aircraft under one Noise Group ................................. 24  
4.3 A-weighted source spectra for Departure Class 104 without adjustment and with maximum thrust NPD adjustment at 1 m from the source. ................................. 27  
4.4 Comparison between Environment Canada and CFSR ground temperature data for July 1, 2014 ................................. 29  
5.1 Flowchart depicting the steps for the calculation procedure ................................. 34  
5.2 YVR runway end points ................................. 37  
5.3 Map Reference Point (mrp) for YVR radar data. The latitude and longitude coordinates are (49.252534,−123.001297) and the location is shown by the red asterisk. ................................. 39  
5.4 Overlaying event tracks on Google maps ................................. 41  
5.5 This plot serves as a graphical representation of errortab showing the differences between measured and calculated L₁p for approach events on runway 26L for 4 noise monitors triggered on July 1, 2014 ................................. 43  
6.1 Wind profile at YVR on July 1, 2014 at 12 A.M. (Pacific Daylight Time) ................................. 47  
6.2 Varying topography around NMTs ................................. 48
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3</td>
<td>Arrivals on the north runway (26R) triggering 6 NMTs</td>
<td>50</td>
</tr>
<tr>
<td>6.4</td>
<td>This plot serves as a graphical representation of error tab showing the differences between measured and calculated $L_p$ for approach events on runway 26R for 6 noise monitors triggered on July 1, 2014.</td>
<td>51</td>
</tr>
<tr>
<td>6.5</td>
<td>Departure from south runway (08R). Event correlation ID 15656674</td>
<td>52</td>
</tr>
<tr>
<td>6.6</td>
<td>A-weighted source spectra for Departure Class 103 without adjustment and with maximum thrust NPD adjustment at 1 m from the source.</td>
<td>53</td>
</tr>
<tr>
<td>6.7</td>
<td>This plot serves as a graphical representation of error tab showing the differences between measured and calculated $L_p$ for a departure event from runway 08R on September 27, 2014 for two different input spectra. One adjusted for the maximum thrust NPD and another is the un-adjusted spectrum.</td>
<td>54</td>
</tr>
<tr>
<td>6.8</td>
<td>Summary of observations</td>
<td>55</td>
</tr>
<tr>
<td>A.1</td>
<td>Snapshot of the user interface for the airport selection window in AEDT.</td>
<td>60</td>
</tr>
<tr>
<td>A.2</td>
<td>Snapshot of the user interface for airport layout design toolbar.</td>
<td>61</td>
</tr>
<tr>
<td>A.3</td>
<td>Snapshot of the APT_SEGMENT table in SQL.</td>
<td>61</td>
</tr>
<tr>
<td>A.4</td>
<td>Snapshot of the SQL user interface to modify the number of editable rows in a table.</td>
<td>62</td>
</tr>
<tr>
<td>A.5</td>
<td>Snapshot of the SQL table APT_SEGMENT after modification.</td>
<td>63</td>
</tr>
<tr>
<td>A.6</td>
<td>Snapshot of an AEDT study showing the radar data flight track for a particular event that corresponds to to DISCOVER-AQ acoustic dataset</td>
<td>64</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Format of matrix \( xz \) .................................................. 14
3.1 Potential aircraft noise databases for ASCENT Project 40 ............. 16
4.1 Radar file format and content description. The data related to flight location is relative to a map reference point (mrp). ............................... 21
4.2 Format of ‘RDconstruct’ matrix ......................................... 22
4.3 Sound Pressure Level (SPL) thresholds for NMT receivers to register an activity as a noise event ............................................. 22
4.4 \( L_{AS_{\text{max}}} \) NPD data corresponding to TRENT8 Aircraft ID for maximum thrust. The ten leftmost columns correspond to the ten slant distances in feet ........ 26
4.5 1/3 octave band source spectrum for B777-300 from fleet database, Spectral class 105 ............................................................... 26
4.6 Steps for calculating input source spectra .................................. 26
4.7 Adjusted, A-weighted spectrum at 1000 ft from source .................. 27
4.8 Time window table .......................................................... 30
4.9 Aligning the direction of aircraft operations and wind velocity .......... 31
5.1 1/3 octave band spectrum retrofitted from NPD data .................... 33
5.2 Format of NGConstruct table ............................................. 36
5.3 Format of LpTable .......................................................... 42
5.4 Errortab matrix format .................................................... 43
6.1 Comparison of results from in-house code and NORD2000, homogeneous sound speed profile .................................................. 45
6.2 Comparison of results from the in-house code and case 1 of NORD2000 after removing the terrain effect ........................................ 46
6.3 Effect of elevation cutoff on Case 3 NORD2000 results .................. 47
6.4 Errortab matrix format and example ....................................... 49
6.5 Errortab for approach events on 26R triggering 6 NMTs ................. 51
Acknowledgments

Thank you to my parents for their sacrifices, unconditional love, trust and continuous support. I thank them for believing in me and my aspirations. I thank my adviser Dr. Victor Sparrow for letting me be a part of his research group and for guiding me through the project with immense patience while multitasking many roles splendidly with a calm and composed demeanor. Thank you for supporting me through the project and encouraging my extra curricular interests. It has been a great learning experience working with you. I am grateful to my committee members Dr. Dan Russell and Dr. Michelle Vigeant for their feedback and suggestions. I sincerely thank the Graduate Program in Acoustics and all my teachers for the enriching education, dedication and work ethic they have imparted on me. I am very fortunate to be a part of this institution and very grateful for the opportunities it provided for academic, professional and personal growth. I thank my friends at the Acoustics Program for enriching discussions and for being an addition to my pool of closest friends. I also thank my roommates, who helped make an apartment a home! I thank all my friends, for whom time zones were immaterial.

I thank Dr. Bill He from the U.S. Federal Aviation Administration (FAA) and, Eric Boeker and Juliet Page from the U.S. Department of Transportation Volpe Center, for their guidance and valuable inputs to this project. I also thank Rachel Min and Mark Cheng from the Vancouver International Airport Authority for providing data for this project and for their contributions and support. Penn State would also like to thank Birger Plovsing of DELTA, Inc. at Hørsholm, Denmark for the use of their NORD2000 MATLAB code.

[This work was funded by the U.S. Federal Aviation Administration (FAA) Office of Environment and Energy as part of ASCENT Project 40 under FAA Award Number 13-C-AJFE-PSU. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the FAA or other ASCENT sponsors.]
Chapter 1  
Motivation & Objectives

1.1 Motivation for aircraft research

Over the past few decades, the advances in science and technology have made aircraft much quieter. At the same time, the need for faster transportation has increased. Figure 1.1 shows that from 1975 to 2015, the population (in millions) exposed to a Day-Night Level (DNL) metric of 65 dB dropped by a factor of around $\frac{1}{20}$ from 7.00 to 0.34. During the same time period, the number of passengers traveling by plane nearly quadrupled from 200 to 760 million.

![Figure 1.1: Historical trends in aviation [1]](image)

The aviation industry is critical for the economic growth. In the United States alone, the aviation industry employs 11.8 million people, supports $1.5$ trillion annually in economic activities and constitutes a 5.4% share of the GDP [1]. The environmental effects of this industry leave an impact on noise, air quality, water quality, energy or fuel consumption and global climate.

Focusing on just the noise impact, aviation activity affects policy, land management and
urban planning. The noise contours around an airport are a deciding factor for noise zoning, which dictates land use policies. Aviation noise prediction affects many spheres of economic and environmental development and sustainability.

The primary motivation for aircraft noise research are community noise complaints, the need for compliance with the most recent standards, the impact of aviation on the environment, human health and development, sleep disturbance and annoyance. The capability to accurately predict aviation noise therefore plays a major role in the confidence of applying prediction results to related policy making decisions.

1.1.1 Developments in past decades and future

Noise is best mitigated at the source. There has been an extensive research conducted in the past on making aircraft engines quieter and efficient. Newer aircraft have turbo engines with increased bypass ratio, simplified lift mechanisms and better aerodynamic design. The research to improvise aircraft design is still ongoing. Procedural changes in take-off and approach have been suggested and implemented for jet planes such as continuous climb operation (CCO) and continuous descent operation (CDO). CCO and CDO were computationally developed techniques for optimized fuel efficiency and noise footprint. Precise navigation systems have enabled aircraft to follow very specific routes. As a result, the airspace near airports has a potential to be more concentrated. Mitigation measures and noise abatement procedures are therefore crucial to sustainability of airport operations in the vicinity of urban areas.

As the need for quicker transportation increases with globalization, aerospace companies are striving to introduce low boom supersonic aircraft for civilian transportation purposes. Unmanned helicopters, pods and new aircraft in the near future will need to be managed in a way to lower their noise impact on surrounding areas.

Various major airports around the world are making efforts not only to monitor and mitigate noise, but also to communicate with and engage in community participation through hotlines for complaints from neighborhoods in the vicinity of airports. Many airports have implemented airport curfews and noise surcharges as a means to distribute noise over time and discourage noisy events from taking place at certain times of the day.

1.1.2 FAA’s efforts towards aviation research

The Federal Aviation Administration (FAA) is the primary sponsor for the Aviation Sustainability Center (ASCENT) initiative. Also known as the Center of Excellence for Alternative Jet Fuels and Environment, ASCENT is a collaborative effort to make aviation industry sustainable and efficient. Under various ASCENT projects, research is being conducted in the areas of alternative fuel, combustion emission and air quality, and noise emissions due to aviation related activities. The ASCENT projects contribute to increasing the Research Readiness Level (RRL) for noise emission and propagation capabilities for possible future use in FAA aviation analysis tools such as the Aviation Environmental Design Tool (AEDT). This is a powerful software to analyze impact of aviation on air quality and noise. The AEDT software has enhanced capabilities to support
environmental impact studies at the national and international levels.

This research is a part of ASCENT Project 40 where the overall goal is to quantify uncertainties in aircraft noise prediction at the source, path and receiver. The Department of Aerospace engineering at Penn State is looking into the uncertainties in noise emission from aircraft as a source. The Graduate Program in Acoustics at Penn State contributes towards studying the uncertainties in propagation path due to refraction effect of temperature and wind gradient in atmosphere. The research group at the Department of Mechanical engineering at Purdue University studied the uncertainties at the receiver end due to ground impedance and doppler effect.

1.1.3 Objectives

The work represented in this thesis contributes towards achieving one of FAA's stated goals to “Reduce the U.S. population exposed to significant aircraft noise around airports”. The objective is to close in the gaps in understanding uncertainties that arise due when a homogeneous atmosphere is considered as opposed to a real atmosphere which has a certain temperature gradient and wind gradient. The gradual changes in atmospheric parameters result in a gradual change in sound speed with height and causes refraction of sound. This study determines the capabilities of existing propagation models in comparison to field measurement data for predicting aircraft noise in real world situations. Hence, there is a desire to understand the effects of this atmospheric inhomogeneity on aviation noise prediction. The second objective is to identify sets of field data to represent specific propagation scenarios. Filtering data is necessary when large datasets are available.

1.2 Outline of this thesis

The purpose of this thesis is to assess data from Vancouver International Airport and develop a methodology to use this data set for desired objectives. This thesis explains how the radar data, field measurement data and the weather data were used to analyze the effects of atmospheric inhomogeneity on noise prediction.

The remainder of this thesis is laid out as follows:

- Chapter 2 presents a literature review of some background concepts.
- Chapter 3 provides information regarding the datasets considered for this project.
- Chapter 4 presents procedures to extract selective information from various datasets and compile it in files that can be used as inputs to the calculation methodology.
- Chapter 5 discusses the calculation methodology adopted for this project.
- Chapter 6 presents the comparisons between field measured data and results from the two propagation models implemented.
- Chapter 7 discusses the assumptions, conclusions and possible future work for the project.
Chapter 2  
Outdoor Sound Propagation and NORD2000 model

This chapter discusses background information on aspects of the objectives of this research project, some basic concepts in acoustics, practical information regarding datasets, the software used, and the propagation models implemented for calculations.

2.1 Outdoor sound propagation

The propagation of sound through the atmosphere brings challenges for accurate predictions owing to complex outdoor scenarios. There are many books (See [4], [5], [6], [2], [7]) that detail the physics of acoustic wave propagation. This section provides a summary of concepts relevant to this thesis such as the spherical spreading, atmospheric attenuation and refraction of sound. The attenuation of sound due to wind turbulence, vegetation, barriers etc. is not considered in this project.

2.1.1 Spherical source in a free field

In most environmental acoustic case studies, the source is considered a monopole or point source, an object pulsating at a frequency whose wavelength is at least an order of magnitude greater than the characteristic dimension of the object. The disturbance of air particles around the object spreads spherically in all directions; this can be imagined as an expanding shell shaped wavefront. Every point on the wavefront is in phase, and the wavefront propagates outward from the source at its center.
In the case where the source is a large aircraft, the wavefronts can be assumed to be spherical, even though the source size is smaller than a wavelength and if the distance from the source is large in comparison to the dimension of the aircraft (Figure 2.1). In most practical cases, this holds true.

The space around the source is categorized into two geometrical sound fields. The near field is the space immediately enveloping the source where only a fraction of the total energy is radiated. Measurements made in the near field are of limited importance for outdoor acoustic studies. The far field is the region beyond the near field and, if devoid of obstacles, is called the free field. Generally, the far field begins at a distance greater than the wavelength of the lowest frequency emitted by the source or an order of magnitude greater than the largest dimension of the source. If the source is located in an ideal atmosphere which is a homogeneous, quiescent and, an isotropic medium, the sound pressure in the far field is inversely proportional to the distance from the source as

\[ p(r, t) = \frac{A}{r} \cdot \cos(\omega t - kr). \]  

(2.1)

2.1.2 Spherical spreading and inverse square law

The sound pressure level (SPL) in dB, corresponding to an effective sound pressure \( p_{av} \) is defined as

\[ L_p = 10 \log_{10} \left( \frac{p_{av}^2}{p_{ref}^2} \right), \] 

(2.2)

where \( p_{av} = \frac{A}{r\sqrt{2}} \) and \( p_{ref} = 2 \times 10^{-5} \text{ Pa} \), is the reference pressure.

The relationship between \( L_p \) at 2 distances, \( r_1 \) and \( r_2 \), is then given by the inverse square law

\[ L_p(r_1) - L_p(r_2) = 10 \log_{10} \left( \frac{r_2^2}{r_1^2} \right) = 20 \log_{10} \left( \frac{r_2}{r_1} \right). \] 

(2.3)
In terms of sound power level \( (L_w) \), the sound pressure level is defined as

\[
L_p = L_w - 10 \times \log_{10}(4\pi r^2) = L_w - 20 \times \log_{10}(r) - 11.
\] (2.4)

### 2.1.3 Atmospheric attenuation

Section 2.1.2 discussed spherical spreading in an ideal atmosphere. The real atmosphere consists of gases and each individual component contributes towards converting a fraction of acoustical energy into thermal energy due to internal friction and molecular relaxation. More details about classical absorption and molecular or relaxational absorption are available in the references [8].

The rate at which the sound energy is absorbed as an acoustic wave passes through the atmosphere is quantified by the air absorption coefficient or sound attenuation coefficient. It can be expressed in \([\text{dB/100m}]\) and is denoted by \( \alpha \). The important components of total absorption coefficients are classical absorption coefficient \( (\alpha'_c) \) and the absorption coefficient for molecular relaxation of the prominent constituents of the atmosphere, oxygen \( (\alpha_1) \) and nitrogen \( (\alpha_2) \). The classical absorption coefficient depends on viscosity, thermal losses and losses due to rotational relaxation. \( \alpha'_c \) is proportional to the square of frequency and while \( \alpha_1 \) and \( \alpha_2 \) have peak effect at particular frequencies called vibrational molecular relaxation frequencies [9]. The total atmospheric absorption coefficient is calculated using equation

\[
\alpha = \alpha'_c + \alpha_1 + \alpha_2.
\] (2.5)

The atmospheric attenuation can also be calculated using a plot such as the one shown in Figure 2.2. In the plot \( \alpha'_c \) is the classical absorption coefficient in \([\text{Np/m}]\) and \( \alpha_1 \) and \( \alpha_2 \) are the atmospheric coefficients due to molecular relaxation of oxygen and nitrogen at relaxation frequencies \( f_1 \) and \( f_2 \) respectively. These plots vary for different values of atmospheric parameters such as temperature and relative humidity. Figure 2.2 is used for calculating the total atmospheric attenuation coefficient for sound in air at 20°C at 1 atm pressure and with a relative humidity equal to 20%.
Figure 2.2: Log-log plot of sound absorption coefficient versus frequency for sound in air at 20°C, 1 atm pressure and with 20% relative humidity [3].

Attenuation due to atmospheric absorption over a distance $r$ is calculated as described in international standard ISO 9613-1:1996 [10]

$$dL_{atm} = \frac{\alpha r}{d_{ref}}$$  \hfill (2.6)

$dL_{atm}$ depends on temperature and relative humidity. $dL_{atm}$ is negligible at low frequencies around 50 Hz and increases to significant values for high frequencies. Here, $r$ is the distance the sound travels and $d_{ref}$ is 100 m.

2.1.4 Refraction of sound

In a homogeneous, quiescent atmosphere, a wavefront continues to propagate along a straight line perpendicular to the wavefront. If the atmosphere is quiescent yet inhomogeneous, and has a uniform sound speed gradient, the wavefronts advance such that the particles propagate at different speeds and the line perpendicular to the wavefronts appears to bend towards the
direction of the lower sound speed. This bending of the sound waves due to a change in sound speed caused by inhomogeneities in the medium is called refraction.

For the purpose of aviation noise, altitudes up to 10 km are of importance. The International Standard Atmosphere specifies the altitude of the troposphere to 11 km [2]. The temperature variation is mostly linear in the troposphere and the spatial progression is quantified by a term dT/dZ [K/m] but denoted as dtdz in this study. Note that unless stated otherwise, in this thesis Z or z denote altitude or height of the source above the ground.

The speed of sound in air is proportional to the square root of absolute temperature. Figure 2.3(a), shows a profile with a steady decrease in temperature with elevation. This is a negative temperature gradient called a temperature lapse that usually occurs during daytime when the ground is at a higher temperature than the air above it. The speed of sound reduces with an increase in elevation. This results in upward bending of sound rays from a source in the free field. Due to upward refraction some areas, which would otherwise receive sound waves in an homogeneous atmosphere profile, do not receive any direct sound. Such areas are called shadow zones [9].

![Figure 2.3: Effect of temperature gradient on sound speed [2]. In this figure, the absolute temperature is denoted by T and the height is denoted by H.](image)

Figure 2.3(b), shows a positive temperature gradient called a temperature inversion, which usually occurs at night and early morning. Sound waves are bent towards the ground. Shadow zones are not created in the downward refraction case. Due to refraction of the sound, rays follow a curved path with a radius of curvature inversely proportional to the gradient of sound speed [2].

For outdoor sound propagation, refraction is also caused by the effect of wind speed gradients.
The strength of the wind profile is quantified in terms of wind speed in m/s and a direction at a certain height above the ground. A wind speed gradient as shown in Figure 2.4 creates a sound speed variation. The sound is bent away from higher velocity towards lower velocity regions. Figure 2.4 shows upward bending of sound in the upwind direction of the source which creates a shadow zone and downward bending in the downwind direction.

![Figure 2.4: Refraction of sound due to a wind profile](image)

### 2.2 Aviation Environmental Design Tool

The Aviation Environmental Design Tool (AEDT) is a software program developed to model aircraft performance in space and time in order to estimate noise, fuel consumption, emissions and air quality consequences [11]. It replaced the Integrated Noise Model (INM) and Emission & Dispersion Modeling System (EDMS) by integrating capabilities of both and enhancing the user interface and other computational capabilities. AEDT is now the required software program for environmental compliance of airports and air traffic spaces.

#### 2.2.1 Computation in AEDT

Figure 2.5 gives a broad overview of the background processes that take place during the simulation of a study in AEDT. Each individual project created with AEDT is called a study, to which a user can define one or more airports, aircraft events and non-aircraft equipment activity. The trajectories of aircraft events are overlayed on a GIS integrated map. AEDT then pulls relevant information from two system-provided databases (AIRPORT and FLEET) using a relational data management system (Microsoft SQL server 2008 R2) and, places only the relevant information in the study database. The input data is processed by adjusting source spectrum data to appropriate thrust settings and applying adjustments for environmental, meteorological, operational and position effects. Details about these adjustments are provided in the AEDT Technical manual [11].
Results can be computed in sixteen different metrics for user defined single point receivers and grid contours.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Process</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport data</td>
<td>✓ Accept data</td>
<td>Single Point Receivers</td>
</tr>
<tr>
<td>Fleet data</td>
<td>✓ Determine unadjusted</td>
<td>– 16 different metrics</td>
</tr>
<tr>
<td></td>
<td>noise levels</td>
<td>Contours</td>
</tr>
<tr>
<td>External input or GUI input</td>
<td>✓ Apply adjustments</td>
<td>– Fixed and dynamic grids</td>
</tr>
<tr>
<td>Study database</td>
<td>✓ Annualize results</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.5: Schematic of AEDT’s background processes.

AEDT computes noise from a series of individual aircraft operations, known as single event noise. The final result metric is calculated by combining the effect of many such single events. The following procedure highlights the key steps in the computation of result metrics [11].

1. Accept aircraft specific data – equipment, noise, position, operational data and, study specific data – weather, terrain, boundary or ambient data as input.

2. Determine the un-adjusted noise values at receptors using NPD data at predefined slant distances. Noise levels at any slant distance that fall in between these distances is calculated by linear interpolation.

3. Apply adjustments to account for environmental, meteorological, operational and position effects.

4. Compute single event metrics at user defined single point receptors or grid receptors.

5. Repeat the above process for each single event.

6. Compute noise output as per the desired output metric.

7. Annualize noise result – Annualization is a process of performing a weighted aggregation over noise results for selective or all cases to create results that represent noise over the time period of interest. For instance a single event, if repeated 50 times a day, need not be simulated 50 times. It is instead calculated once and then annualized.

This study does not cover simulation in AEDT or the comparison of measured and calculated data with predictions made in AEDT. However, AEDT constitutes an important part of learning the process of noise predictions from input data. The calculation procedures discussed in Chapter 4 are referenced to the procedures adopted by AEDT.
2.2.2 Study database

The study database is a superstructure which pulls information from system databases according to user input. It contains information and references to external source data, including input and configuration, intermediate data from calculations and output results. It is essentially a set of tables that are connected in a pre-defined hierarchy. Contents in each table are identified by the ‘primary key’ which is a column of certain identification codes which can be IATA code/ ICAO codes/ NOISE GROUP IDs etc. Each row contains information specific to that primary key. The primary keys create the relational links between different tables across databases.

2.2.3 Airport database

The airport database consists of a set of 31 tables containing a global set of airports and associated data such as runway, historical atmospheric conditions and time zones. This database is updated in the latest AEDT versions for changes in airport structure, modification to runways, taxiways, airspace and airport buildings [11].

Once a study database is created, selected airport information is added to it from the airport database. A user specified airport can also be created or modified via the graphical interface or ASIF import function.

2.2.4 Fleet database

The fleet database consists of 151 tables containing information on aircraft and non-aircraft equipment. This database contains approximately 4600 aircraft – airframe and engine combinations, and 400 non-aircraft emission source [11]. For noise purposes, aircraft are identified by the International Civil Aviation Organization (ICAO) specified Aircraft Noise Performance Database (ANP) and Eurocontrol Base of Aircraft Data (BADA). This study concerns only commercial, fixed winged aircraft whose information is available in ANP tables of the fleet database. The hierarchic relationship between various fleet database tables is given in [12].

Spectral class data — Aircraft with similar spectral shape are grouped together by engine type and/or number of engines and further categorized in groups by spectral shape and operation type. These groups are called spectral class data and were originally developed for FAA’s INM. They consist of sound pressure levels at one third octave band frequencies ranging from 50 Hz to 10,000 Hz values for four metrics discussed in section 4.4.

Noise–Power–Distance (NPD) database — The calculation methodology of AEDT relies on a set of Noise vs. Power (thrust setting) vs. Distance (NPD) datasets, which are specific to aircraft type, operation type and noise metric. For fixed-wing aircraft, NPD data consists of a set of decibel levels for various combinations of aircraft operational modes (approach, departure, overflight), engine power (thrust setting), slant distances from aircraft to receiver considering a straight flight path of infinite length parallel to the ground through a standard atmosphere [11]. The predefined NPD slant distances in [feet] are 200, 400, 630, 1000 (305 m), 2000, 4000, 6300, 10000, 16000 and 25000. During simulation calculations, corrections or adjustments are added for curved flight paths and the local ground atmosphere.
2.2.5 Weather model in AEDT

The spectral data available in the fleet database is corrected for weather data as per a reference day specified in SAE-AIR-1845 where, ambient temperature is 25 °C, relative humidity is 70% and wind speed is 0 m/s [13]. The spectral database in AEDT allows a user to take into account atmospheric absorption due to the effects of local meteorological conditions on an airport-specific basis by giving the user a choice to select atmospheric adjustment method from two other atmospheric adjustment guidelines namely, SAE-ARP-866a (which takes into account temperature and relative humidity effects) or, SAE-ARP-5534 (which takes into account temperature, relative humidity and atmospheric pressure effects) [11]. However for the purpose of noise calculations, a vertically homogeneous atmosphere is assumed.

2.3 Difference between real and homogeneous atmosphere

The vertically homogeneous atmosphere assumed in noise calculation methodology of AEDT is analyzed as a source of potential uncertainties in this research project. The weather parameters such as temperature and wind in a real atmosphere differ a lot from the assumed standard reference day atmosphere. Figure 2.6 shows the temperature as a function of height above ground at Vancouver, Canada. Every color represents a temperature profile 6 hours apart in a day on July 1, 2014. The black line represents the homogeneous temperature profile of a standard day atmosphere (25 °C).

![Figure 2.6: Schematic of AEDT’s background processes.](image)

The effect of refraction can be tested using a sound propagation model that uses a temperature...
and wind gradient. The next section describes the NORD2000 propagation model used for this research.

## 2.4 NORD2000 propagation model

NORD2000 is an outdoor sound prediction model that is based on geometrical ray theory combined with the theory of diffraction. It was mainly designed for industrial and environmental transport noise. Nevertheless, studies have been conducted to analyze the performance of the NORD2000 model for aircraft noise [14]. Initial versions of this model assumed the sound rays would follow straight lines. The model was later modified to include effects of moderate atmospheric refraction by introducing sound rays that follow a curved path. This propagation model provides sufficient accuracy for uncomplicated weather conditions [15]. These are conditions where the sound speed either increases or decreases monotonically with altitude without any abrupt changes in the sound speed gradient. Such an uncomplicated weather is approximated by a combination in a logarithmic and linear sound speed profile i.e. a log-linear profile.

The effective sound speed \( c(z) \) is calculated using equation

\[
c(z) = A \cdot \ln \left( \frac{z}{z_0} + 1 \right) + B \cdot z + C, \tag{2.7}
\]

where \( z \) is the height above the ground surface, \( z_0 \) is the roughness length and \( A, B, C \) are constants [16]. The logarithmic part of \( c(z) \) is determined only by the wind speed component, \( u \) in [m/s] at a height \( z_u \) and equals

\[
A = \frac{u(z_u)}{\ln \left( \frac{z_u}{z_0} + 1 \right)}. \tag{2.8}
\]

The linear part depends on the temperature, \( t \) in [°C] which is assumed to have a linear relation with elevation,

\[
B = \frac{dt}{dz} \cdot 10.025 \frac{t + 273.15}{. \tag{2.9}
\]

The constant \( C \) in equation 2.7 represents the homogeneous sound speed at a certain temperature, \( t \) in [°C] on the ground and is calculated from

\[
C = 20.05 \times \sqrt{t + 273.15}. \tag{2.10}
\]

The sound pressure level at the receiver (\( L_r \)) is calculated using equation

\[
L_r = L_w + \Delta L_d + \Delta L_a + \Delta L_t + \Delta L_s + \Delta L_r. \tag{2.11}
\]

\( L_w \) is the input source level and \( \Delta L_s \) are the attenuation levels which mostly are negative numbers. \( \Delta L_d \) is the attenuation due to spherical divergence of sound waves in a free field, \( \Delta L_a \) is the attenuation due to the atmospheric absorption effect based on the international standard ISO 9613(1). It considers an average day air temperature (25°C) and relative humidity (70%), unless
specified otherwise. \( \Delta L_t \) is the attenuation due to the effect of terrain. NORD2000 performs segmented calculations. If the space between the source and a receiver has an uneven terrain defined, the code calculates the attenuation for the space between two points on the \( x - z \) plane and then goes on to calculate the attenuation for the next segment. The terrains are simplified as flat, hill or valley terrain in the NORD2000 model. \( \Delta L_s \) is the attenuation due to the effect of scattering zones. \( \Delta L_r \) is the attenuation due to the effect of reflections off of screens or barriers [16]. This research is limited to applying attenuation due to the effect of spherical spreading, atmospheric absorption and effect of terrain.

A single source – path – receiver scenario can have any combination of screen and terrain. The space between two discontinuities is assessed to identify which of the above effects need to be included. For this project, only spherical spreading, air absorption, and refraction effects are considered using version version 19 of the NORD2000 propagation model, which is programmed as a MATLAB file, compro19ABC. It is important to note that although NORD2000 can include screens, terrain, ground reflections, turbulence and scattering zones, these effects were turned off for this project by assigning zero values to certain parameters [17]. The total attenuation was calculated based on the following inputs:

1. Number of points in the terrain, \( n_{xz} \). In this study, we assume a flat terrain and compute the sound at a receiver due to an aircraft event. We assume there is no obstacle in the source to receiver path. The value of \( n_{xz} \) is always equal to 2. This value will be 3 or more if we want to specify any discontinuity in the flat terrain such as a hill/valley or presence of a screen, such as a noise barrier.

2. Terrain profile matrix, \( xz \). This matrix has \( n_{xz} \) number of rows and four columns as shown in Table 2.1

Table 2.1: Format of matrix \( xz \)

<table>
<thead>
<tr>
<th>( x )</th>
<th>( z )</th>
<th>Flow resistivity</th>
<th>Ground roughness parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>( z_1 )</td>
<td></td>
<td>0 (recommended for flat terrain)</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>( z_2 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. Roughness length, \( z_0 \)

4. Ground temperature, \( t_0 \) in degree Celsius

5. Percentage relative humidity, RH

6. Coefficients \( A, B, C \)

7. Standard deviations \( s_A, s_B \) – set to zero as a default.

8. Turbulence strength, \( C_{v2} \) and \( C_{t2} \) – set to 0 as turbulence is ignored.

9. Scattering zone definitions, \( n_{scat} \) and \( xz_{scat} \) – set to 0 as the scattering effect is ignored.
The code checks for the effects of all types of attenuations and calculates the total attenuation $\Delta L$. The NORD2000 model has been validated for source elevation up to 200 m with a $\pm 2\%$ accuracy in overall A weighted sound pressure levels [14], but is said to have a good accuracy up to a 1 km altitude. For airport noise predictions, most aircraft with approach or departure type operations have lower altitudes than en-route aircraft. This model was chosen because it is time efficient, reasonably accurate and, most importantly, it accounts for the effect of curved sound rays due to refraction.

### 2.5 Summary

This chapter discussed some basic background information about the important aspects of this project in a very concise manner. The reader is requested to refer to the appropriate references for more detailed information.

The next chapter describes the radar, noise monitor and weather datasets used for this project.
Chapter 3  
Datasets used for calculations

The purpose of this study was to use real weather data to predict noise from aircraft events and compare the results with measured data. Several noise measurement datasets were considered for use in this study, are tabulated in Table 3.1, and are described in brief in sections 3.1 through 3.3. However, due to the practicalities of acquiring datasets, only the Vancouver International Airport (YVR) dataset was available at the time.

Table 3.1: Potential aircraft noise databases for ASCENT Project 40

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Focus</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISCOVER-AQ</td>
<td>Propeller Aircraft</td>
<td>U.S. Government</td>
</tr>
<tr>
<td>BANOERAC</td>
<td>Mostly en-route jets</td>
<td>EASA</td>
</tr>
<tr>
<td>NINHA</td>
<td>En-route propeller planes</td>
<td>EASA</td>
</tr>
<tr>
<td>Vancouver Intl. Airport (YVR)</td>
<td>Typical aircraft mix at large airport</td>
<td>Vancouver Airport Authority</td>
</tr>
<tr>
<td>SILENCE(R)</td>
<td>Airbus A320/A340</td>
<td>Xnoise/Airbus</td>
</tr>
</tbody>
</table>

3.1 DISCOVER-AQ dataset

The DISCOVER-AQ data set is a comprehensive combined study of air quality (combustion emissions) and noise conducted by NASA near Houston, TX in 2013. It consists of extensive air quality and noise measurement data collected at seven different measurement locations recording various types of flight maneuvers by two aircraft. This data set was used as a part of another project, ASCENT 5, before ASCENT Project 40 began.

Several events were recorded as a part of this database. Only two of those events, 33 and 35 were analyzed, and acoustic data was utilized from two measurement locations at Smith Point, TX, mnemonically named SP1 and SP2. Event 35 was a fly-over maneuver near SP1, and event 33 was a downward spiral maneuver near the SP2 location. An attempt to predict the sound pressure level due to this aviation activity was made using spherical spreading and air absorption...
(a code similar to in-house developed code discussed in Chapter 5) and simulation in the AEDT software. See the reports [18].

For this thesis, the DISCOVER-AQ data was not selected for subsequent analysis as only propeller aircraft were used. Further, Purdue University is examining this data in detail.

### 3.2 BANOERAC, NINHA and SILENCE(R) datasets

BANOERAC stands for “Background noise level and noise levels from en-route aircraft”. This study was conducted in three phases: calculation of approximate background noise levels based on population density of each country in Europe, actual measurements of background noise and aircraft noise in rural areas and analysis of results [19].

NINHA stands for “Noise Impact of aircraft with Novel engine configurations in mid- to High Altitude operations”. An advanced counter-rotating open rotor (CROR) engine was introduced in propeller planes in the 1980s. Since then, significant efforts were being taken to improve its aero-acoustic design. It was found that the noise generated by CROR engines could significantly affect communities. The NINHA project assessed noise impact from en-route airplanes fitted with CROR engines [20].

SILENCE(R) stands for “Significantly lower community exposure to aircraft noise”. This project focused on validating large scale noise reduction solutions concerning the aeroacoustic design and active noise control in engines, nacelles and airframes. [21]

These three datasets were unavailable to Penn State during 2015-2017. They may be available for use in future ASCENT noise research.

### 3.3 Vancouver International Airport

Vancouver International Airport (YVR) is located on Sea Island about 10 miles Southwest of downtown Vancouver, British Columbia in western Canada. It is the second largest airport in Canada and experiences an air traffic and fleet mix similar to many major American airports. Figure 3.1 shows the main terminal at YVR.
There are three major runways – North, South, and Crosswind runway (Figure 3.2). Most departures take off towards the west and fly over the ocean reducing the high noise impact on the communities near by. There is also a fourth runway, the Water Aerodrome, which is used by sea planes, but such operations are not considered for analysis in this thesis.

The YVR airport authority has a network of 20 noise monitors in the vicinity of the airport.
and surrounding neighborhoods. These Noise Monitor Terminals (NMTs) record sound pressure level using the Sound Exposure Level \((SEL)\) and Maximum Sound Pressure Level \((L_{\text{max}})\) metrics for noise events above 65 dBA during the day time and 55 dBA during night. Most NMTs are located on top of school buildings or in open areas. The approximate locations are marked by dots and numbered in Figure 3.3.

![Noise Monitor Terminal (NMT) locations](image)

Figure 3.3: Network of Noise Monitor Terminals (NMTs) around YVR. X and Y axes are the longitude and latitude coordinates.

The Vancouver dataset was chosen for its similar aircraft fleet to most major U.S. based airports. Extensive information regarding aircraft paths and corresponding noise events was available. In addition, since this is a non-U.S. based airport, the comments and observations made through this research study do not have any implications on U.S. airport policies.

### 3.4 Summary

This chapter presented an introduction to the datasets that were considered and could potentially be used in future ASCENT noise research projects. The Vancouver dataset used for this research study was introduced. The next chapter will introduce the inputs used for the noise analysis.
Chapter 4  |  Process input data

4.1 Introduction

An important feature of this research is the availability of real measured data from a major airport. This chapter explains, in detail, the procedures adopted to extract necessary data from the big datasets that were available. Two datasets were received from the Vancouver International Airport (YVR) authorities. The radar dataset has information about the flights in the YVR airspace and the Noise Monitor Terminal (NMT) dataset consists of sound pressure level measurements from the NMT locations. The input source levels for the different aircraft in the YVR fleet were calculated using the spectral class data and Noise–Power–Distance (NPD) data. The analysis also uses weather data which was extracted from the Climate Forecast System Reanalysis (CFSR) dataset and Environment Canada dataset. The code written to extract required data from the available datasets would be compatible with data from other locations if the format of dataset were to be consistent. The code is included in APPENDIX C and requires inputs from the different datasets discussed in this chapter.

4.2 Radar data

4.2.1 Data format

Vancouver International Airport (YVR) uses a Brüel & Kjær product, Airport Noise monitoring and Management System (ANOMS), for flight track monitoring and data acquisition. LT6 was a utility format written for ANOMS version 6 to list flight tracks in a specific format. It was developed by HMMH Inc. The more recent version of ANOMS also supports this format [22]. LT6 is an encoded text format that can be decoded by simply renaming its extension from *.lt6 to *.txt in order to view its contents. The text file can then be imported in SQL, MATLAB, Mathematica or other suitable software for further analysis. For this study, LT6 format was converted to a comma separated variables (.csv) format Microsoft Excel spreadsheet.
4.2.2 Available information & extraction of necessary data

The YVR airport authority provided seven radar data files. Each file consisted of information for flight events during one day. Radar track data had information about the operational data such as event ID, date, start and end time, airport codes, operational runway number, number of track points, X, Y and Z distances from a reference coordinate and, speed and time stamp for all of those points. It also consisted of aircraft data such as aircraft ID and aircraft type.

Table 4.1 presents the information along with associated line number. The highlighted gray rows indicate the information that was essential for further analysis in this project. It is important to note that the line numbers are only a reference in this table but are not enumerated in the real data.

Table 4.1: Radar file format and content description. The data related to flight location is relative to a map reference point (mrp).

<table>
<thead>
<tr>
<th>Line #</th>
<th>Explanation</th>
<th>Format/Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TRACK operation number</td>
<td>TRACK xxxxx</td>
</tr>
<tr>
<td>2</td>
<td>Event ID (correlation number)</td>
<td>MM-DD-YYYY</td>
</tr>
<tr>
<td>3</td>
<td>Track start date</td>
<td>HH:MM:SS (24 hr format)</td>
</tr>
<tr>
<td>4</td>
<td>Track start time</td>
<td>HH:MM:SS (24 hr format)</td>
</tr>
<tr>
<td>5</td>
<td>Track end time</td>
<td>HH:MM:SS (24 hr format)</td>
</tr>
<tr>
<td>6</td>
<td>Airport ID</td>
<td>ICAO/IATA code</td>
</tr>
<tr>
<td>7</td>
<td>Flight number</td>
<td>Text</td>
</tr>
<tr>
<td>8</td>
<td>Owner name</td>
<td>Text</td>
</tr>
<tr>
<td>9</td>
<td>Aircraft type</td>
<td>ICAO/IATA text format</td>
</tr>
<tr>
<td>10</td>
<td>Aircraft category</td>
<td>Single char ASCII code</td>
</tr>
<tr>
<td>11</td>
<td>Beacon</td>
<td>Text</td>
</tr>
<tr>
<td>12</td>
<td>Operation type</td>
<td>A, D, O, T</td>
</tr>
<tr>
<td></td>
<td>(A-Arrival/ D-Departure/ O-Overflight/ T-Touch &amp; go)</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Way point</td>
<td>Text</td>
</tr>
<tr>
<td>14</td>
<td>Destination airport</td>
<td>ICAO/IATA code</td>
</tr>
<tr>
<td>15</td>
<td>Runway name</td>
<td>Text</td>
</tr>
<tr>
<td>16</td>
<td>min elevation</td>
<td>ASCII in meters</td>
</tr>
<tr>
<td></td>
<td>(relative to map reference point (mrp))</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>max elevation relative to mrp</td>
<td>ASCII in meters</td>
</tr>
<tr>
<td>18</td>
<td>min range (slant distance relative to mrp)</td>
<td>ASCII in meters</td>
</tr>
<tr>
<td>19</td>
<td>max range (slant distance relative to mrp)</td>
<td>ASCII in meters</td>
</tr>
<tr>
<td>20</td>
<td>Count of track points to follow</td>
<td>ASCII count</td>
</tr>
<tr>
<td>21</td>
<td>x,y,z,v,t (relative to mrp)</td>
<td>SI units</td>
</tr>
<tr>
<td></td>
<td>mrp coordinates for YVR study: (latitude, longitude) = (49.252534,-123.001297)</td>
<td>x, y and z in [m], v in [m/s], t in [s]</td>
</tr>
</tbody>
</table>
Individual events were identified by the word ‘TRACK’ followed by operation number. Each entry in the radar datafile could be thought of as a cell with either strings or numbers or a combination of both. Since the sequence of data remained consistent, any information could be extracted from the dataset knowing its line number with respect to the line number of the word ‘TRACK’ for that event. In order to sort the events based on operation type, time of the day, and type of the aircraft, a tabulated form of radar dataset was required. The 2-D matrix ‘RDconstruct’ created in MATLAB, listed only the required information for all events. The layout of the output matrix is shown in Table 4.2. Each event had a varying number of track points and depending on the operation type of the event, either the first point (for departures) or the last point (for arrivals) would be located on the runway assigned to the aircraft operation.

Table 4.2: Format of ‘RDconstruct’ matrix

<table>
<thead>
<tr>
<th>Track number</th>
<th>Corr ID</th>
<th>Operation type</th>
<th>Runway</th>
<th>Timestamp</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
</table>

### 4.3 Noise Monitor Terminal (NMT) dataset

The NMTs are located at varying elevations (3 m to 18 m) from the ground, but mostly near the ground when compared to the elevation of the aircraft being assessed. As a result, the NMTs register noise events from other sources such as community noise and traffic noise. A YVR noise report from 2010 states that during a three month monitoring period, only 2% of the noise events registered by NMTs were related to aviation activity. A small fraction, less than 20% of this 2%, was due to aviation activity at YVR airport, while the remaining was due to aircraft operations from nearby airports [23]. This gives us an estimate of the quantity of noise data collected compared to usable events.

Due to the extensive nature of the data, sorting of the NMT data and identifying the correct pair of radar data and NMT data for an event is challenging. When an aircraft flies over, one or many NMTs under its direct or lateral path are triggered. A NMT records a noise event if the sound pressure level of an aircraft exceeds the predefined threshold for a set period of time. Table 4.3 shows the set NMT thresholds for YVR.

Table 4.3: Sound Pressure Level (SPL) thresholds for NMT receivers to register an activity as a noise event.

<table>
<thead>
<tr>
<th>Hours</th>
<th>SPL threshold</th>
<th>Duration, $\tau$ [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 AM to 10 PM</td>
<td>$\geq 65$ dBA</td>
<td>10 – 120 s</td>
</tr>
<tr>
<td>10 PM to 6 AM</td>
<td>$\geq 55$ dBA</td>
<td>10 – 120 s</td>
</tr>
</tbody>
</table>

The radar and NMT datasets contain a correlation ID, which is an eight digit number generated
if there were noise complaints. If a radar event has a zero correlation ID, it means none of the NMTs registered a noise event and there were no noise complaints associated with that event. If the radar event has a non zero correlation ID but does not match an event from NMT data, it means that the event is correlated to a noise complaint but does not qualify the criteria mentioned in Table 4.3 to be a noise event.

The NMT data provides information about the event correlation IDs, measured SEL and $L_{\text{max}}$ values at select NMTs where those events pass the noise event criteria. The duration of measurement is different for all NMTs triggered for the same event. The duration of measurement for an event, $\tau$, is the time elapsed in [s] between times $t_1$ and $t_2$ as shown in Figure 4.1. This figure shows a typical aircraft flyover time history.

![Figure 4.1: Selection of time duration for an aircraft event. $t_1$ and $t_2$ are the 10 dB down points from $L_{\text{max}}$.](image)

The equivalent sound pressure level $L_{eq}$ is calculated from the SEL data and duration of measurement, $\tau$ using the equation

$$L_{eq} = SEL - 10 \times \log_{10} \tau.$$  \hspace{1cm} (4.1)

4.4 Spectral class data and NPD curves

As previously discussed in section 2.2, AEDT and its predecessor software INM use a fleet database. Many aircraft, each with unique ACFT_ID (aircraft ID) but with similar shape of source spectrum,
are grouped together in Noise Groups identified by NOISE_IDs. In the process, individual levels for each aircraft belonging to the same noise group are normalized and represented as one sound spectrum consisting of sound pressure values at 1/3rd octave band center frequencies ranging from 50 Hz to 10 kHz. Figure 4.2 shows 4 aircraft with similar spectral shape. These are grouped together and normalized to obtain one spectrum level for all aircraft in the same noise group.

Figure 4.2: Spectral source levels for four aircraft under one Noise Group

This section details the procedural steps taken to extract aircraft source level information from different tables in the fleet database, for each event being analyzed, and to calculate a source spectrum to be used in the propagation models as input.

1. Once an event is selected for analysis, it is checked if the associated data is available in the fleet database tables. If not, the event is ignored.

2. Extracting spectral class data: Each noise group is then linked to a spectral class ID for departure and approach operation mode. A method, similar to the one implemented in AEDT, is used to identify the correct noise group for aircraft. Radar information consists of aircraft that have jet and propeller type engines. In this study only jet airplanes were considered. The procedure involves looking up a chain of information from the fleet database tables. The detailed information about accessing the SQL database is mentioned in APPENDIX A. However, all the necessary tables were compiled into a single spreadsheet.

   (a) The radar dataset contains information about the aircraft engine type (Jet/Propeller), aircraft model number (BADA_ID), and operation type(A/D/O). The ANP_AIRPLANE_ID corresponding to the BADA_ID is noted by looking up the FLT_EQUIPMENT table in the fleet database.

   (b) Then, the NOISE_ID is found by referring to the fleet database table FLT_ANP_AIRPLANES using the key ANP_AIRPLANE_ID which is also called ACFT_ID.
Then, the spectral class ID for the aircraft associated with the event can be identified from the fleet database table FLT_ANP_AIRPLANE_NOISE_GROUPS using the NOISE_ID and operation type of event. SPECT_APP (2XX) and SPECT_DEP (1XX) are the spectral class IDs for approach and departure events, respectively.

The ANP database provides unweighted sound pressure level $L_{n,ref}(d_{ref})$ [dB] at a slant distance of 1000 ft or 305 m from the source considering the effect of a reference atmosphere, where $n$ is the 1/3rd octave band center frequency ranging from 50 Hz to 10 kHz. This protected data was provided by FAA and is accessible with due permissions in the file SpectralClassData.xlsx. Each set of sound levels $L_{n,ref}(d_{ref})$ represent a group of aircraft identified by NOISE_ID. A set of calculation steps are performed to adjust the spectral data by adding or subtracting sound levels across the entire frequency range by an amount delta, which is calculated by interpolation or extrapolation. Step number 4 highlights the method to calculate delta. These values are based on data normalized using average atmospheric attenuation rates defined in SAE-ARP-1845.

3. Extracting data from Noise–Power–Distance tables

The FLT_ANP_NPD_CURVES table in the fleet database consists of overall sound pressure levels at 10 slant distances from the source. This set of values is available for each permutation and combination of the following variables:

(a) Noise_ID - The identification code of the aircraft.

(b) Thrust setting [lbs] - The weight in pounds. The higher the thrust setting, the higher are the emission levels.

(c) Operation mode - Approach(A)/ Departure(D)/ After burner(X). We will not be considering after burner related data in this study.

(d) Metric - The NPD table in the fleet database consists of overall sound pressure level for each of the following four metrics referenced to 160 knots.

- $L_{AE}$: A-weighted sound exposure level
- $L_{EPN}$: Effective perceived noise level
- $L_{ASmx}$: Maximum A-weighted sound level
- $L_{PNTS_{mx}}$: Maximum tone corrected perceived noise level

Based on the Noise ID and operation mode of the event being analyzed, we select the NPD data for $L_{ASmx}$ and maximum trust setting. Table 4.4 shows a set of NPD data for maximum available trust setting in approach and departure operation types for the Boeing 777-300. TRENT8 is the AIRCRAFT_ID and M stands for $L_{ASmx}$ at 10 slant distances from the source. Two operation types are included to show that $L_{ASmx}$ levels for take-off are higher than those for landing aircraft. However, the process described in this section is only for the departure profile of this aircraft.
Table 4.4: $L_{AS_{max}}$ NPD data corresponding to TRENT8 Aircraft ID for maximum thrust. The ten leftmost columns correspond to the ten slant distances in feet.

<table>
<thead>
<tr>
<th>Noise_ID</th>
<th>Metric</th>
<th>Op_type</th>
<th>Thrust</th>
<th>$L_{200}$</th>
<th>$L_{400}$</th>
<th>$L_{630}$</th>
<th>$L_{1k}$</th>
<th>$L_{2k}$</th>
<th>$L_{4k}$</th>
<th>$L_{6.3k}$</th>
<th>$L_{10k}$</th>
<th>$L_{16k}$</th>
<th>$L_{25k}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRENT8</td>
<td>M</td>
<td>A</td>
<td>28000</td>
<td>100.2</td>
<td>86.7</td>
<td>81.3</td>
<td>72.9</td>
<td>63.8</td>
<td>57.5</td>
<td>50.8</td>
<td>43.2</td>
<td>34.4</td>
<td></td>
</tr>
<tr>
<td>TRENT8</td>
<td>M</td>
<td>D</td>
<td>80000</td>
<td>110.5</td>
<td>104</td>
<td>99.6</td>
<td>95.1</td>
<td>88</td>
<td>80.3</td>
<td>74.4</td>
<td>67.9</td>
<td>60.5</td>
<td>52</td>
</tr>
</tbody>
</table>

Table 4.5 shows the unweighted spectral class data, as available in the fleet database for the Boeing aircraft B777-300.

Table 4.5: 1/3 octave band source spectrum for B777-300 from fleet database, Spectral class 105

<table>
<thead>
<tr>
<th>freq [Hz]</th>
<th>50</th>
<th>63</th>
<th>80</th>
<th>100</th>
<th>125</th>
<th>160</th>
<th>200</th>
<th>250</th>
<th>315</th>
<th>400</th>
<th>500</th>
<th>630</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_p$ [dB]</td>
<td>66.5</td>
<td>60.4</td>
<td>67.1</td>
<td>75</td>
<td>78.2</td>
<td>79.3</td>
<td>71.5</td>
<td>76.7</td>
<td>74.4</td>
<td>74.6</td>
<td>72.3</td>
<td>71.9</td>
</tr>
<tr>
<td>freq [Hz]</td>
<td>800</td>
<td>1000</td>
<td>1250</td>
<td>1600</td>
<td>2000</td>
<td>2500</td>
<td>3200</td>
<td>4000</td>
<td>5000</td>
<td>6300</td>
<td>8000</td>
<td>10000</td>
</tr>
<tr>
<td>$L_p$ [dB]</td>
<td>71.1</td>
<td>70</td>
<td>69</td>
<td>68.8</td>
<td>67</td>
<td>65.5</td>
<td>63.4</td>
<td>59.2</td>
<td>53.8</td>
<td>50</td>
<td>44.7</td>
<td>38.2</td>
</tr>
</tbody>
</table>

4. Adjusting spectral data as per NPD data

The following steps are taken to adjust the spectral class data as per NPD thrust settings.

(a) If the thrust setting for the event is a known parameter, the overall sound pressure level values in the NPD data should be interpolated or extrapolated to the true thrust setting.

(b) Since the actual thrust settings are not available, the maximum thrust data are used.

(c) The spectral data is then A-weighted.

Table 4.6: Steps for calculating input source spectra

<table>
<thead>
<tr>
<th>freq [Hz]</th>
<th>50</th>
<th>63</th>
<th>80</th>
<th>100</th>
<th>125</th>
<th>160</th>
<th>200</th>
<th>250</th>
<th>315</th>
<th>400</th>
<th>500</th>
<th>630</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spectral data [dB]</td>
<td>66.5</td>
<td>60.4</td>
<td>67.1</td>
<td>75</td>
<td>78.2</td>
<td>79.3</td>
<td>71.5</td>
<td>76.7</td>
<td>74.4</td>
<td>74.6</td>
<td>72.3</td>
<td>71.9</td>
</tr>
<tr>
<td>A-wt adjustment [dB]</td>
<td>-30.2</td>
<td>-26.2</td>
<td>-22.5</td>
<td>-19.1</td>
<td>-16.1</td>
<td>-13.4</td>
<td>-10.9</td>
<td>-8.6</td>
<td>-6.6</td>
<td>-4.8</td>
<td>-3.2</td>
<td>-1.9</td>
</tr>
<tr>
<td>A-weighted level [dBA]</td>
<td>36.3</td>
<td>34.2</td>
<td>44.6</td>
<td>55.9</td>
<td>62.1</td>
<td>65.9</td>
<td>60.6</td>
<td>68.1</td>
<td>67.8</td>
<td>69.8</td>
<td>69.1</td>
<td>70</td>
</tr>
<tr>
<td>freq [Hz]</td>
<td>800</td>
<td>1000</td>
<td>1250</td>
<td>1600</td>
<td>2000</td>
<td>2500</td>
<td>3200</td>
<td>4000</td>
<td>5000</td>
<td>6300</td>
<td>8000</td>
<td>10000</td>
</tr>
<tr>
<td>Spectral data [dB]</td>
<td>71.1</td>
<td>70</td>
<td>69</td>
<td>68.8</td>
<td>67</td>
<td>65.5</td>
<td>63.4</td>
<td>59.2</td>
<td>53.8</td>
<td>50</td>
<td>44.7</td>
<td>38.2</td>
</tr>
<tr>
<td>A-wt adjustment [dB]</td>
<td>-0.8</td>
<td>0</td>
<td>0.6</td>
<td>1</td>
<td>1.2</td>
<td>1.3</td>
<td>1.2</td>
<td>1</td>
<td>0.5</td>
<td>-0.1</td>
<td>-1.1</td>
<td>-2.5</td>
</tr>
<tr>
<td>A-weighted level [dBA]</td>
<td>70.3</td>
<td>70</td>
<td>69.6</td>
<td>69.8</td>
<td>68.2</td>
<td>66.8</td>
<td>64.6</td>
<td>60.2</td>
<td>54.3</td>
<td>49.9</td>
<td>43.6</td>
<td>35.7</td>
</tr>
</tbody>
</table>

(d) Calculate the overall A-weighted sound pressure level $= 80$ dBA.

(e) Delta, $\Delta$ is the difference between NPD level at 1000 feet and the overall A-weighted sound pressure level at 1000 feet. In this case, $\Delta = 95 - 80 = 15$ dB

(f) Delta is added to the spectral data to get the adjusted, A-weighted spectral class data at 305 m or 1000 ft from the source.
Table 4.7: Adjusted, A-weighted spectrum at 1000 ft from source

<table>
<thead>
<tr>
<th>freq [Hz]</th>
<th>50</th>
<th>63</th>
<th>80</th>
<th>100</th>
<th>125</th>
<th>160</th>
<th>200</th>
<th>250</th>
<th>315</th>
<th>400</th>
<th>500</th>
<th>630</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-weighted level, A-wt SC [dBA]</td>
<td>36.3</td>
<td>34.2</td>
<td>44.6</td>
<td>55.9</td>
<td>62.1</td>
<td>65.9</td>
<td>60.6</td>
<td>68.1</td>
<td>67.8</td>
<td>69.8</td>
<td>69.1</td>
<td>70</td>
</tr>
<tr>
<td>Adjusted spectrum [dBA]</td>
<td>51.3</td>
<td>49.2</td>
<td>59.6</td>
<td>70.9</td>
<td>77.1</td>
<td>80.9</td>
<td>75.6</td>
<td>83.1</td>
<td>82.8</td>
<td>84.8</td>
<td>84.1</td>
<td>85.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>freq [Hz]</th>
<th>800</th>
<th>1000</th>
<th>1250</th>
<th>1600</th>
<th>2000</th>
<th>2500</th>
<th>3200</th>
<th>4000</th>
<th>5000</th>
<th>6300</th>
<th>8000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-weighted level, A-wt SC [dBA]</td>
<td>70.1</td>
<td>70</td>
<td>69.6</td>
<td>69.8</td>
<td>68.2</td>
<td>66.8</td>
<td>64.6</td>
<td>60.2</td>
<td>54.3</td>
<td>49.9</td>
<td>43.6</td>
<td>35.7</td>
</tr>
<tr>
<td>Adjusted spectrum [dBA]</td>
<td>85.3</td>
<td>85.0</td>
<td>84.6</td>
<td>84.8</td>
<td>83.2</td>
<td>81.8</td>
<td>79.6</td>
<td>75.2</td>
<td>69.3</td>
<td>64.9</td>
<td>58.6</td>
<td>50.7</td>
</tr>
</tbody>
</table>

Although the spectrum is now adjusted, it must still match the source level at 1000 feet as our reference (maximum thrust) NPD value. The overall sound pressure level is 95.1 dBA. This is exactly equal to the value highlighted in Table 4.4 in the column for the 1000 feet distance.

The adjusted, A-weighted sound spectrum at 1000 feet from the source includes the effect of a reference atmosphere as defined in SAE-ARP-1845 [11]. This will be referred to as the new $L_{n,ref}(d_{ref})$ in the following steps.

The adjusted spectrum for departure profiles at maximum power is approximately 10 dB higher than the spectrum un-adjusted with NPD data. This trend is seen to be very consistent in departure profiles. Approach profiles however show a different trend. The adjusted spectrum levels are either equal or a few dB lower than the un-adjusted source spectrum. Figure 4.3 shows an example of adjusted and un-adjusted departure source spectra for the noise group spectral class 104.

![Figure 4.3: A-weighted source spectra for Departure Class 104 without adjustment and with maximum thrust NPD adjustment at 1 m from the source.](image)
5. The atmospheric parameters at a known location and at a known time can be different from the average reference atmosphere. Atmospheric attenuation effects must be removed from \( L_{n,\text{ref}}(d_{\text{ref}}) \) before calculating the effect of specific atmospheric parameters. Equation 4.2 is used to correct the reference spectrum by removing the effect of a reference atmosphere. The term \( \alpha_{n,\text{ref}} \) is the coefficient of atmospheric absorption as discussed in section 2.1.3. \( L_{n}(d_{\text{ref}}) \) is the A-weighted sound pressure level at 305 m from the source without the effect of atmospheric absorption.

\[
L_{n}(d_{\text{ref}}) = L_{n,\text{ref}}(d_{\text{ref}}) - \alpha_{n,\text{ref}} \times d_{\text{ref}} \tag{4.2}
\]

6. Considering the aircraft to be an omnidirectional source in the free field, Eq. 4.3 is used to calculate the A-weighted sound pressure level at \( d_{1} \), distance of 1 m from the source. The spectrum thus obtained considers atmospheric effects specific to the airport and time. The product, \( \alpha_{\text{atm}} \cdot d_{1} \) is negligible and can be ignored.

\[
L_{n}(d_{1}) = L_{n}(d_{\text{ref}}) - 20 \cdot \log_{10}\left(\frac{d_{\text{ref}}}{d_{1}}\right) - \alpha_{\text{atm}} \cdot d_{1} \tag{4.3}
\]

The calculation methodology discussed in Chapter 6 uses \( L_{n}(d_{1}) \) as the A-weighted input source sound level spectrum specific to the aircraft.

### 4.5 Weather data

Two weather datasets were available for this project: Environment Canada and the Climate Forecast System Reanalysis (CFSR) dataset. The Environment Canada dataset is available from www.weather.gc.ca, a Canadian government website that provides 24 hour current and historic weather data. This dataset consists of temperature, wind direction, wind speed and relative humidity at a particular measurement location every hour. The hourly data points are either measured or estimated at only the ground level at any location required. A coordinate point named Vancouver Intl. Airport is one of the closest to the airport vicinity but only provides information on atmospheric parameters at a single elevation of 19 m from sea level.

The CFSR data is a global, high resolution 4-D dataset. It consists of estimated information for all atmospheric parameters such as temperature, wind component, humidity pressure etc on a 6-hourly basis, at 37 isobaric levels at a latitude/longitude coordinate. CFSR takes a global snapshot of all atmospheric parameters at 37 isobaric levels and records them at universal Greenwich time (UTC). All information must be corrected for the time zone. The procedure to download CFSR data from the NCAR-UCAR website is detailed in a document ‘CFSR: Not-So-Quick-Start Guide’ [24].

Both datasets have their own advantages. While the Environment Canada dataset has an excellent time resolution, CFSR data has a better spatial resolution. The propagation model adopted in this study requires a temperature gradient and wind speed at a certain elevation. The change in temperature with height \( (dT/dz) \) can be calculated only from the CFSR data.
set since temperature estimates are available at 37 isobaric levels that approximately equate height in kilometers when the location is at sea level altitude. Figure 4.4 shows a comparison of Environment Canada and CFSR temperature data for July 1, 2014. (Data at 12:00 AM on July 2, 2014 was also be included as it better represents the atmospheric scenario for the noise events occurring closer to midnight on July 1.) Such a comparison was made to understand the agreement between temperature records from two datasets. It is observed that at 12 AM and 6 AM the temperature estimated for both datasets matches well, while there is 5 to 10 degC difference for 12 PM and 6 PM data. Attempts were not made to investigate the reasons for this difference.

![Figure 4.4: Comparison between Environment Canada and CFSR ground temperature data for July 1, 2014](image)

This project focuses on the impact of sound refraction due to atmospheric temperature and wind gradients on the propagation of sound. Therefore the CFSR dataset, which has a good spatial resolution is preferred over Environment Canada data.

### 4.5.1 Extracting necessary data from CFSR dataset

There are essentially 4 data points in the CFSR dataset for any given parameter in a 24-hour period since the reanalysis data is available at 12 AM, 6 AM, 12 PM and 6 PM. Our location of interest, Vancouver, British Columbia, Canada is on Pacific Daylight Time (PDT) and lags UTC by 7 hours. The time data points are shifted by 6 hours instead of 7, for sake of convenience in calculations, to accommodate and easily identify the correct time zone. For example, 12 PM UTC is 5 AM PDT (local time at Vancouver in July). The weather data corresponding to 12
PM UTC at YVR lat/lon coordinates is in fact the data corresponding to 5 AM PDT local time at Vancouver for the same day. For convenience, a 6 hour lag is considered instead of the actual 7 hour time difference. This assumption does not make a significant difference since the data is considered to be constant over a 6 hour period.

The radar data contains the start and end time for all events. Table 4.8 shows the cut-on, cut-off time for each time zone and the CFSR data point that must be picked for correct weather data corresponding to the start time of an event being analyzed.

<table>
<thead>
<tr>
<th>Time window (cut on–cut off time)</th>
<th>Local time data point</th>
<th>UTC CFSR data point</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 AM to 3 AM</td>
<td>12 AM</td>
<td>6 AM UTC</td>
</tr>
<tr>
<td>3 AM to 9 AM</td>
<td>6 AM</td>
<td>12 PM UTC</td>
</tr>
<tr>
<td>9 AM to 3 PM</td>
<td>12 PM</td>
<td>6 PM UTC</td>
</tr>
<tr>
<td>3 PM to 9 PM</td>
<td>6 PM</td>
<td>12 AM UTC (+1 day)</td>
</tr>
<tr>
<td>9 PM to 11:59 PM</td>
<td>12 AM (+1 day)</td>
<td>6 AM UTC (+1 day)</td>
</tr>
</tbody>
</table>

Once a time specific datapoint is selected for an event, the temperature at 37 isobaric levels above that coordinate point is available. The temperature gradient, \( \frac{dT}{dz} \) is equal to the slope of a linear fit for a plot of temperature as a function of height. The extent of atmosphere column to be considered can be modified by the user. The code considers an atmospheric column up to the maximum elevation of the aircraft from the event radar data.

Wind speed information is available in the CFSR dataset as 2 components – U and V, at 37 isobaric levels. The U and V components of the wind are quantified as positive numbers in [m/s] when their vectors point from West to East and South to North, respectively. Most of the air traffic at the YVR airport is operated from the North and South runways (see Figure 3.2) which are along a 80°-260° line. The events whose paths are mostly over the sea (towards the West of YVR airport) are not considered for analysis. The events that are registered as noise events by the noise monitors are assumed to fly along the 90°-270° line, in the East – West direction. Since crosswinds are ignored, the U-component data is sufficient information as wind speed.

Each runway is identified by the runway ends (refer to [25] for information on runway nomenclature). Departures are associated with the tail-side runway end, and Approach is associated with the nose-side runway end (refer to Figure 5.2). The propagation code NORD2000, used in this study (see section 2.4), expects wind speed in the direction of propagation. It could often happen that the direction of the U-component wind might be opposite to the direction of propagation. The polarity of the U-component wind would be required to switch in such cases. Table 4.9 mentions these cases and the necessary action.
Table 4.9: Aligning the direction of aircraft operations and wind velocity

<table>
<thead>
<tr>
<th>Operation</th>
<th>Runway/Runway end/ direction</th>
<th>U-component (sign/direction)</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Departure</td>
<td>North/08L/W→ E</td>
<td>+ve/W→ E</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>South/08R/W→ E</td>
<td>−ve/W← E</td>
<td>u × −1</td>
</tr>
<tr>
<td>Approach</td>
<td>North/08L/W← E</td>
<td>+ve/W→ E</td>
<td>u × −1</td>
</tr>
<tr>
<td></td>
<td>South/08R/W← E</td>
<td>−ve/W← E</td>
<td>✓</td>
</tr>
</tbody>
</table>

The corrected U-component of wind velocity and the elevation of the nearest isobaric level (converted to m) is used as input for the NORD2000 propagation model.

### 4.6 Summary

This chapter describes several details and processes to extract information from the datasets to make it usable for the actual calculation of sound pressure levels. Section 5.1.3 describes the methodology which uses the data extracted here, as an input.

The next chapter explains the calculation procedures that were used.
Chapter 5  
Calculation Methodology

This chapter discusses the procedure adopted for analysis in the different phases of the project. The procedure was modified along the course of the project depending on the availability of data. In the first phase, weather data (temperature and wind) were not used for analysis and correct source sound pressure level data was not available. Sound pressure levels at receivers located at Noise Monitor Terminals (NMTs) were calculated using a back calculation method which is explained in Section 5.1.1. In the second phase, weather data was not considered although a sound pressure level spectra for Boeing 777 was considered instead of a back calculation approach. Section 5.1.2 discusses the assumptions and methodology for this phase. Finally, in the third phase, when the methodology for using Noise-Power-Distance (NPD) thrust settings and spectral class data for noise groups was programmed, source spectra (matching the aircraft associated with noise event being analyzed) was considered along with the weather data (temperature and wind gradients). The procedural steps followed in this phase were developed by the author and are explained in section 5.1.3. It is assumed that the reader has extracted all of the necessary information in the correct format as mentioned in Chapter 4.

5.1 Progression of methodology

5.1.1 Phase 1 – Source level data and weather data input were not used

Initially, the measured NMT data and radar dataset were the only inputs available. The goal was to calculate sound pressure level at receiver locations, at the same coordinates and height as the NMTs, and compare the calculated overall $L_p$ to the sound pressure levels recorded at the corresponding NMTs. A backcalculation methodology was developed using basic outdoor sound propagation equations to calculate an overall sound power level, $L_w$, using the measured sound pressure level at NMT. The following procedure was adopted to calculate the sound pressure level at the receiver locations.

An event that triggered multiple NMTs was preferably selected over events that triggered only one or two NMTs for the purpose of better comparing results to measured data. Then, using the location information in the radar data, slant distances from each NMT location to every available
coordinate point along the path of the aircraft were calculated. There was a unique coordinate location along the aircraft path that was closest to each NMT. The shortest NMT to aircraft distance pair was selected to calculate the overall sound power level, $L_w$. The sound exposure level (SEL), $L_p$, measured at this NMT, was then used to calculate the sound power level of the source using an equation for spherical spreading in the free field,

$$L_w = L_p + 20 \times \log_{10} r + 11.$$  

It was assumed that the source, an airplane, was an omnidirectional source propagating in a free field, and the sound power level remained constant throughout its path. The atmosphere was assumed to be vertically homogeneous. Then using the same equation, the overall sound pressure level, $L_p$ was calculated for all other active NMT locations with $L_w$ as a known parameter. The distance, $r$ was the shortest distance from each corresponding active NMT to the aircraft path. The results were then compared. This was a crude method but formed the basis of the in-house developed propagation model discussed in step number 7 of the Section 5.1.3.5.

5.1.2 Phase 2 – Retrofitted input spectrum for B77L

Using an overall sound power level as input source was not the best method. So, in a later phase of the project, a source spectra for Boeing 777-300 was used. It was a retrofitted input spectrum, normalized and A-weighted sound pressure level at 1/3rd octave band center frequencies ranging from 50 Hz to 2 kHz. The sound pressure level at center frequencies greater than 2 kHz were excluded since atmospheric attenuation was significant for these frequencies and diminished their contribution towards the overall sound pressure level.

The procedure for retrofitting NPD curves is detailed in Kieran Poulain’s thesis [26]. Table 5.1 shows the retrofitted data derived from NPD curves for a Boeing 777-300 for a departure profile with the maximum thrust setting which is the worst case scenario.

<table>
<thead>
<tr>
<th>freq [Hz]</th>
<th>50</th>
<th>63</th>
<th>80</th>
<th>100</th>
<th>125</th>
<th>160</th>
<th>200</th>
<th>250</th>
<th>315</th>
<th>400</th>
<th>500</th>
<th>630</th>
<th>800</th>
<th>1000</th>
<th>1250</th>
<th>1600</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPL[dBA]</td>
<td>116</td>
<td>121</td>
<td>130</td>
<td>134</td>
<td>136</td>
<td>140</td>
<td>141</td>
<td>145</td>
<td>147</td>
<td>147</td>
<td>147</td>
<td>148</td>
<td>148</td>
<td>148</td>
<td>149</td>
<td>150</td>
<td>130</td>
</tr>
</tbody>
</table>

The assumptions made in this phase were the same as the ones made in the first phase. In addition, a retrofitted source spectrum for a Boeing 777-300 was used for any event, irrespective of the aircraft associated with the event. The procedure for calculation discussed in the section 4.2 was used in this phase with an exception of using the input spectral data as shown in Table 5.1 instead of calculating the correct source spectra.

5.1.3 Phase 3 – Appropriate source level data and weather data were used

In this phase, the source spectra were calculated based on the aircraft associated with the event being analyzed. Real weather data from the Climate Forecast System Reanalysis (CFSR) was used to calculate the temperature gradient and wind speed. The flowchart in Figure 5.1 highlights
the important steps in the MATLAB code used for analyzing the YVR events. This description following the flowchart can be considered as a user guide to the MATLAB code developed by the author, attached in APPENDIX C. The code has been divided into sections and each section, in particular, one through four, must be executed sequentially.

Figure 5.1: Flowchart depicting the steps for the calculation procedure
5.1.3.1 Section 1 – Import input files

For the Vancouver airport study, each radar data file consisted of information for one calendar day, and each NMT data file contained data measurements for a month. The radar file originally received in *LT6 format contained extensive information on departures, arrivals, overflights and touch-and-go operations from Vancouver International Airport (YVR) and smaller airports in its vicinity. It was converted from *.LT6 to *.csv while the NMT data file was received in a *.xlsx format.

The first section of the MATLAB code imports the radar data file and the NMT data file into the MATLAB workspace. The data in both files had numeric and character data formats. In order to ensure that all of the data columns were saved in the workspace, the xlsread command was used with three output variables for retaining the data in numeric, character and alphanumeric or cell formats.

5.1.3.2 Section 2 – RDConstruct

The format of a *.LT6 file was designed for the Airport Noise monitoring and Management System (ANOMS) and is detailed in Section 4.2. The ANOMS software has the capability to customize and create files with radar information about events as per the criteria defined by the user. Since this software was not used in this project, it was necessary to extract only selective information in a single data structure for ease of processing.

The flight events could have a varying number of track points (refer to line 21 in Table 4.1). The second section of the MATLAB code creates a data matrix called RDConstruct by searching through the entire radar data sheet for the word ‘TRACK’ and tabulating the information about each event. The format of RDConstruct was shown in Table 4.2.

5.1.3.3 Section 3 – NGConstruct

In this phase of the project source spectral data corresponding to the aircraft associated with the event was used. The spectral class data and Noise-Power-Distance (NPD) data from the database of the Aviation Environmental Design Tool (AEDT) was used to calculate the input source spectra. The spectral class data consisted of normalized sound pressure levels for a group of aircraft with similar spectral shapes. Each noise group was assigned to spectral class IDs. The source spectra then needed to be calculated based on the thrust setting information available in the NPD data following the procedure detailed in Section 4.4.

The third section of the MATLAB code scans through all events from RDConstruct and identifies those events with a jet engine aircraft. The tables in the FLEET database were used to identify the spectral class IDs and a matrix NGConstruct was created. This process helped to identify the events to be excluded from analysis, if the aircraft associated was not a jet engine airplane or if the aircraft associated did not belong to any noise group. The chances of not finding a spectral class ID for a jet aircraft were very low, but this step prevented the code from running into errors at a later stage of analysis.
An example of an NGConstruct is given in Table 5.2. The first column is of the event ID, followed by the aircraft model, equipment ID, noise group to which the aircraft belongs and the spectral class ID for approach and departure operations. In this example, event #3 was excluded from NGConstruct because the aircraft associated with the event was a propeller engine type aircraft.

<table>
<thead>
<tr>
<th>Event #</th>
<th>Aircraft model</th>
<th>Equipment ID</th>
<th>Noise Group</th>
<th>spectral class ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SPECT_APP</td>
</tr>
<tr>
<td>1</td>
<td>B77L</td>
<td>777300</td>
<td>TRENT8</td>
<td>203</td>
</tr>
<tr>
<td>2</td>
<td>B77W</td>
<td>777300ER</td>
<td>GE9015</td>
<td>204</td>
</tr>
<tr>
<td>4</td>
<td>B737</td>
<td>737700</td>
<td>CF567B</td>
<td>203</td>
</tr>
</tbody>
</table>

### 5.1.3.4 Section 4 – Event selection

Before commencing the analysis, the events need be filtered, primarily on the basis of operation type and the associated runway. The user is prompted to provide this information and is informed of the number of events that comply with the preliminary criteria. Every runway is identified by two runway ends as shown in Figure 5.2. The approach events are associated with the runway end that the airplane noise points to, while the departure events are associated with the runway end behind them. This information helps the user to input the desired runway end. An example of the MATLAB command prompt is shown below along with a sample user input in red.

**Select a runway end. South (08R,26L) North (08L,26R) :** 08R

**Dep=1, Arr=2, Overflight=3(runway is immaterial). Select operation :** 1

**Number of events for this operation type from selected runway end =**

In the fifth section of the MATLAB code, the user could specify a single event to be analyzed by providing an input event number which must not exceed the number of events for the given input criteria (operation type and runway end). An array ‘idx_active’ then stored the row numbers or index numbers for the selected events. The code also checked if the first point of departure events and the last point of approach events were accurately picked. These coordinate locations, termed as points of interests (see Section 4.2) could be used to check if the coordinate locations did lie on the selected runway by marking them on Google map plots. The great circle method was used to calculate latitude, longitude coordinates using the distances from a reference coordinate (see APPENDIX B).
5.1.3.5 Sections 5 & 6 – Analyzing events

The analysis of events was separated into two sections. The fifth section of MATLAB code computed single event analysis based on the event number provided by the user. The event number was required to be less than the total number of events shortlisted by the MATLAB code in the previous section. The sixth section of the MATLAB code computed multiple events of the same operation type, runway end, number of NMTs triggered and time window (optional). An example of the MATLAB command prompt and a user input (in red) is given below.

Number of NMTs as a selection criteria : 5
1 -> 12 AM to 3 AM
2 -> 3 AM to 9 AM
3 -> 9 AM to 3 PM
4 -> 3 PM to 9 PM
5 -> After 9 PM

Pick a time window : 5

The procedural steps for the multiple event analysis algorithm is given below. The steps 2 through 8 are essentially the algorithm for single event analysis and are repeated for every eligible event in the sixth section of MATLAB.

1. Each event listed in the idx_active array was assessed based on the input selection criteria. If the aircraft associated with the event was not a jet engine type aircraft, if the aircraft did not belong to any of the noise groups and thus did not have any input source spectra available, if the event start time did not fall in between the time window specified by the
user (if any) or if the number of input NMTs triggered by the event did not match the input specified by the user, then, the event was excluded from the analysis. All other events were considered for the following steps.

2. The correlation ID (corrID) and row numbers corresponding to the event were identified in the NMT data sheet. Each row had SEL, $L_{\text{max}}$ values and the duration in seconds for each NMT receiver triggered by the event.

3. If the user did not define a time window, the start time of the event was identified and an appropriate time window based on Table 4.8 was selected and weather parameters such as temperature gradient, wind velocity and height were calculated from the weather dataset CFSR. If the user specified a time window, only the weather parameters corresponding to that time window were calculated.

4. Based on the aircraft information from the radar data, the adjusted and A-weighted sound pressure levels were calculated as per the process described in Section 4.4, at 1 m from the source. The code also provided an option to choose the input source levels between source spectra adjusted to the maximum thrust setting (saved as ADJlpA1m), unadjusted source spectra (saved as lp_spectrumA) and, retrofitted spectra for Boeing 777-300 (input used in second phase of the project, saved as lw). The input source spectra used by the code was saved under variable name ipL. It must be ensured that the chosen input was saved as ipL and not overwritten later in the run.

5. The radar track data for the event, available in sets of $x$, $y$ and $z$ distances in [m] from a map reference coordinate were used to calculate (latitude, longitude) coordinates. The map reference coordinate for Vancouver International Airport was located far from the airport itself, in the City of Burnaby, to the East of Vancouver City. The location coordinates were (49.252534, −123.001297). The red asterisk in Figure 5.3 shows the exact location of the map reference coordinate.
6. The coordinates for all track points were calculated using the great circle method, described in the APPENDIX B.

7. Calculations using an in-house developed code

The methodology used for calculation of sound pressure level at NMT locations in the first phase of the project was improvised to develop the in-house code. The following steps were taken to calculate overall sound pressure level at each NMT triggered by the event under consideration.

(a) Slant distances between NMT locations and all points along the flight track were calculated and the minimum distance $d_{CP}$ was identified.

(b) Sound pressure levels, $L_n(d_{CP})$ were calculated at $1/3$ rd octave band center frequencies ranging from 50 Hz to 10 kHz, considering the effect of spherical spreading and air absorption using equation

$$L_n(d_{CP}) = L_n(d_1) - 20 \cdot \log_{10}(d_{CP}) + dL_a. \quad (5.1)$$

and then, the overall A-weighted sound pressure level, $L_{p,SS}$ was calculated using equation

$$L_{p,SS} = 10 \times \log_{10} \left( \sum_{n=1}^{n=24} 10^{L_n / 10} \right). \quad (5.2)$$

(c) The above steps were repeated for the next active NMT.
8. Calculations using the NORD2000 code

The MATLAB script compro19ABC.m was the execution file for the NORD2000 propagation model which calculated the total attenuation \((dL)\) due to terrain effects and air attenuation. The compro19ABC.m code was provided to Penn State by Birger Plovsing of DELTA, Inc., Hørsholm, Denmark.

The effective sound speed as a function of height was calculated using equation

\[
c(z) = A \cdot \ln \left( \frac{z}{z_0} + 1 \right) + B \cdot z + C
\]

(see Section 2.4). Depending on which of the coefficients \(A\), \(B\) and \(C\) from equation 2.7 were non zero, the sound speed profile could be homogeneous, linear or log-linear. A function RunCompro was written by the author to execute compro19ABC.m for these three different input conditions. These were treated as three cases and the overall sound pressure level was calculated for each case.

In the first case, the coefficients \(A\) and \(B\) were equated to zero, to obtain a vertically homogeneous sound speed profile equal to constant \(C\) which was calculated using the ground temperature in Equation 2.10. The overall sound pressure level calculated for this case was denoted by \(L_{pN2kC1}\). In the second case, coefficient \(A\) was equated to zero and coefficient \(B\) was calculated using equation 2.9 using a polyfit function to compute \(dtdz\). A linear sound speed profile based on a linear temperature profile was used to calculate the overall sound pressure level, \(L_{pN2kC2}\). In the third case, a log-linear sound speed profile was obtained by considering the effect of wind velocity, \(u\) at a certain height \(z_u\) to calculate coefficient \(A\) using equation 2.8, in addition to the effect of linear temperature gradient. The resulting sound pressure level was denoted by \(L_{pN2kC3}\).

9. Overlay event tracks on a map

The MATLAB function plot_google_map.m was used to plot event tracks on Google maps. This function was available on the MATLAB documentation website [27].

Usually in a multiple event analysis run, many events appear on the google plot with more NMTs than the initial specified criteria. This is common because all the qualifying events have a different set of active NMTs. For instance, if the selected runway was 26L, the operation type was Approach (i.e. 2) and the number of NMTs as an input criteria was 4, the MATLAB code found three events for analysis. The plot for this multiple event run (see Figure 5.4) shows six NMTs.
Figure 5.4: Overlaying event tracks on Google maps

Note that the Google maps API key could expire after many runs, in which case, a new API key can be used.

10. **Tabulate results**

It was essential to compile a table with only the necessary results and select which overall sound pressure levels would be compared to derive an inference. There were measured values at the NMTs in both SEL and $L_{\text{max}}$ metrics, calculated results from the inhouse code, and three sets of results from the NORD2000 propagation code. Since the multiple event analysis was a repeated form of the single event analysis, the results for each single event run were tabulated in a two dimensional matrix called ‘LpTable’, while the results for multiple event run was tabulated in a three dimensional matrix called ‘complex3’. If complex3 has $l \times m \times n$ dimensions, $n$ is the page number. Every page would then be a $l \times m$ matrix for one event. $l$ is the row number storing information about a single NMT for a particular event, and $m$ is the column number corresponding to the information from Table 5.3.

The LpTable consists of the NMT number, the shortest distances between points along the flight path and NMTs, the measured sound exposure level (SEL), maximum A-weighted sound pressure level, $L_{\text{max}}$ and the calculated equivalent A-weighted sound pressure level, $L_{\text{eq}}$. It also consists of results from the propagation models and finally the difference between the calculated results for the homogeneous and log-linear sound speed profiles cases and the measured $L_{\text{max}}$ value.
11. Compare results

The single event section was written to quickly glance at and compare values. However, the multiple event run results that were tabulated in the 3-D matrix were not very easy to understand by visual checks. Also, we needed to quantify the error, the difference between the calculated and measured values for different cases mentioned in point numbers 6 & 7. The aim was to investigate the effect of including a log-linear sound speed profile as against a homogeneous profile. Hence, the results of Case 1 and Case 3 of the NORD2000 propagation code were compared with the measured sound level. The mean error was calculated as follows:

(a) The difference between the overall sound pressure level from Case 1 of the NORD2000 propagation model and the measured \( L_{\text{max}} \) was computed for each active NMT. This was denoted by err1. Essentially, the values in column 6 of Table 5.3 were subtracted from corresponding values in column 9.

(b) The difference between the results from Case 3 of the NORD2000 code and measured \( L_{\text{max}} \) was computed. This was denoted by err2 and was obtained by subtracting the values in column 6 from values in column 11.

(c) The mean of err1 and err2 was then calculated for each NMT. In the example case, the input criteria was to select events that triggered four NMTs. Three events were selected by the MATLAB code for analysis. Not all of the three events triggered the same set of four NMTs, so the total NMTs to be analyzed was six.

(d) A matrix ‘errortab’ was created to list the errors for each NMT for all the events selected for analysis in a multiple event run. The first column of this matrix lists all active NMTs across all events that were analyzed. A few NMTs were triggered by most of these events but a few were unique to a particular event.

(e) Two pairs of columns were appended to the errortab table for every event analyzed in a sequence. The first of the two digits following ‘err’ in columns 2 through 7 denote the type of error (err1 or err2) and the second digit denote the event number. In this case, there were three events, each having a column for err1 and err2. Finally, the two columns containing the averages of err1 and err2 for all the events were appended. The format of errortab is given in Table 5.4.
Table 5.4: Errortab matrix format

<table>
<thead>
<tr>
<th>NMT#</th>
<th>Error for Event 1</th>
<th>Error for Event 2</th>
<th>Error for Event 3</th>
<th>Mean of errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>err11</td>
<td>err21</td>
<td>err12</td>
<td>err13</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>err1_avg</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>err2_avg</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The mean of errors in [dB] were then plotted for each NMT. The blue squares in Figure 5.5 represent the mean error1 while the red circles represent mean error2. A positive error was an over-prediction, which meant that the calculated results were higher than the measured values. A negative error was an under-prediction. If the red circle was closer to the zero line as compared to its corresponding blue square, it meant that the error had reduced owing to the consideration of a log-linear sound speed profile instead of a homogeneous sound speed profile.

\[
\Delta L_p = L_p - L_{max}
\]

Case Blue: Homogeneous Sound Speed
Case Red: Log-Lin Sound Speed

Figure 5.5: This plot serves as a graphical representation of errortab showing the differences between measured and calculated $L_p$ for approach events on runway 26L for 4 noise monitors triggered on July 1, 2014.
5.2 Summary

This chapter described the calculation procedures and the data handling methods for this project. The next chapter describes the results.
Chapter 6
Results

The tabulated results for a single event analysis was stored in the MATLAB workspace as LpTable and the results for a multiple event run was saved as complex3. The results consisted of calculations from two propagation models, the in-house code and NORD2000 (N2k). Also, NORD2000 had three calculation scenarios as discussed in Section 5.1.3.5. This chapter discusses the comparisons between results from different cases, propagation models and measured data. Some trends seen in the results were understandable while a few results were contrary to expectations.

6.1 Comparison of results: In-house code and NORD2000

When NORD2000 was first used in this project, an important aspect of understanding its results was an attempt to see if the results matched the output of the in-house developed code. At this stage, only a homogeneous sound speed profile (case 1) was considered in NORD2000. At first glance, the difference between the overall sound pressure level results of the in-house code (LpSS) and the NORD2000 code considering a homogeneous sound speed profile (LpN2k-C1) was prominent, up to 3 dB. Table 6.2 is an example showing the comparison between results for an approach event on the 26L runway. This event is one of the three events from the multiple event run (see Figure 5.4), that triggered the following set of four NMTs.

Table 6.1: Comparison of results from in-house code and NORD2000, homogeneous sound speed profile

<table>
<thead>
<tr>
<th>NMT#</th>
<th>In-house code results</th>
<th>NORD2000 Case 1 results</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>66</td>
<td>69</td>
</tr>
<tr>
<td>6</td>
<td>73</td>
<td>74</td>
</tr>
<tr>
<td>4</td>
<td>77</td>
<td>78</td>
</tr>
<tr>
<td>2</td>
<td>82</td>
<td>83</td>
</tr>
</tbody>
</table>

These results were expected to match. After investigating, it was found that the NORD2000
propagation model implemented in this study using the RunCompro.m executable, considers the terrain effect in addition to spherical spreading and air attenuation. Attenuation due to terrain \((dL_{ter})\) entails calculation of ground effect as a result of flat/valley/hill type of terrain between the source and receiver. In this study a flat terrain case was assumed. A nominal but non-zero attenuation spectrum was obtained for a terrain effect owing to a ground roughness parameter. However, the in-house code calculates the sound pressure level at each receiver NMT location only considering the effect of spherical spreading and air attenuation, ignoring any terrain effects.

When the terrain effect was neglected while executing the NORD2000 model, the results matched perfectly, not just for this example but for any event. This was a good sanity check. However, the terrain effect was removed only for such comparison. The NORD2000 code was used as is for the final calculations.

Table 6.2: Comparison of results from the in-house code and case 1 of NORD2000 after removing the terrain effect

<table>
<thead>
<tr>
<th>NMT#</th>
<th>In-house code results</th>
<th>NORD2000 Case 1 results (without terrain effect)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>66</td>
<td>66</td>
</tr>
<tr>
<td>6</td>
<td>73</td>
<td>73</td>
</tr>
<tr>
<td>4</td>
<td>77</td>
<td>77</td>
</tr>
<tr>
<td>2</td>
<td>82</td>
<td>82</td>
</tr>
</tbody>
</table>

6.2 Comparison of results: NORD2000 cases and measured data

The main goal of this study was to quantify the effect of using real atmosphere data instead of a homogeneous atmosphere. The Case 1 and Case 3 results of NORD2000 represent homogeneous and real approximated meteorological data, respectively.

6.2.1 Variation in results due to elevation cutoff

The process to determine the temperature gradient and wind velocity from CFSR data was discussed in section 3.5.3. The CFSR data has a good vertical resolution so it was required to analyze which height the atmospheric column needed to be. It was observed that the height of the atmospheric column affected the results. For an example case discussed in this section, the wind speed profile for July 1, 2014 is shown in Figure 6.1. Unlike an ideal gradual gradient, the wind closer to the ground was observed to change sporadically in speed and direction.
When the data up to the maximum source height (approximately 800 m) was being considered, the wind velocity was observed to be 3.29 m/s westwards. When the mean source height was used as a limit, the wind velocity was 0.1 m/s eastward. Up to approximately 300 m, the wind gradient was positive and stronger than the wind gradient between 400 m to 800 m. Not shown in the figure, the temperature gradient was $-0.06$ when the maximum source height was used as a cutoff and 0.21 when the mean height was used. The Case 3 result for both are tabulated in Table 6.3.

### Table 6.3: Effect of elevation cutoff on Case 3 NORD2000 results

<table>
<thead>
<tr>
<th>NMT#</th>
<th>Slant distance</th>
<th>Measured $L_p$</th>
<th>Calculated $L_p$ N2k-Case3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SEL</td>
<td>$L_{\text{max}}$</td>
</tr>
<tr>
<td>15</td>
<td>0.9</td>
<td>81.5</td>
<td>72.4</td>
</tr>
<tr>
<td>6</td>
<td>0.5</td>
<td>82.1</td>
<td>72.2</td>
</tr>
<tr>
<td>4</td>
<td>0.4</td>
<td>79.9</td>
<td>68.6</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>78.8</td>
<td>69.1</td>
</tr>
</tbody>
</table>

There was not a definite conclusion as to which of the two options gave a result more compatible with the measured data. In this case, the results from using the mean source height as cutoff agreed better with the $L_{\text{max}}$ measurements at the NMTs, compared to the results from using the maximum source height.

However, the more realistic approach seemed to be the one where the entire atmospheric
column from source height to receiver was considered. Therefore, all calculations were done with maximum source height as the elevation limit for wind and temperature data consideration.

### 6.2.2 Effect of NMT location topography on results

The NMT dataset did not include the elevation or the latitude/longitude coordinate. The coordinates were approximated by looking up the names of the locations, mentioned in the NMT data, on Google maps and the height of NMTs were approximated using the pictures of the NMTs. The heights of the receiver microphones at NMT locations varied significantly, ranging between 4 m to 12 m above the ground. Some were located on rooftops with potential reflective surfaces near by such as NMT 14 shown in Figure 6.2a, while some were located on a clear field at a considerably higher elevation from reflecting surfaces such as NMT 15 shown in Figure 6.2b.

![Figure 6.2: Varying topography around NMTs](image)

(a) NMT#14 (~6 m height)  
(b) NMT#15 (~2 m height)

The input ground roughness parameter was kept constant for the NORD2000 analysis. Accordingly, the reflective surfaces were not accounted for and this could have affected the results.
6.3 Discussion of results with examples

The three examples showcased in this section were from different days. Each example discusses the operation type and illustrates the trajectories of events on a map, errortable, and scatterplot to represent the comparison between the Case 1 and Case 3 results with the measured $L_{\text{max}}$ values.

6.3.1 Approach events on July 1, 2014 at 26L (south runway) triggering 4 NMTs.

The results are considered for the example discussed in the Chapter 5 where the situation considered was approach events at 26L (south runway) triggering 4 NMTs on July 1, 2014. A time window was not specified. Figure 5.4 represents the events on a Google map. Table 6.4 shows the error i.e. the difference between calculated and measured sound pressure levels.

$$\text{err1}_\text{avg} = \text{mean}(L_{pN2kC1} - L_{\text{max}})$$

$$\text{err2}_\text{avg} = \text{mean}(L_{pN2kC3} - L_{\text{max}}).$$

<table>
<thead>
<tr>
<th>NMT#</th>
<th>Error for Event 1</th>
<th>Error for Event 2</th>
<th>Error for Event 3</th>
<th>Mean of errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>err11</td>
<td>err21</td>
<td>err12</td>
<td>err22</td>
</tr>
<tr>
<td>2</td>
<td>13.9</td>
<td>13.9</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>9.4</td>
<td>9.4</td>
<td>1.8</td>
<td>1.8</td>
</tr>
<tr>
<td>6</td>
<td>1.8</td>
<td>1.8</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>-17.2</td>
<td>-16.2</td>
</tr>
<tr>
<td>15</td>
<td>-3.4</td>
<td>-6.4</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

There are always a few NMTs that are triggered by all events in one multiple events run, while some NMTs that are triggered only by a few of the events. However, when calculating the mean error, the sum of all the err1s and all the err2s were divided by the total number of events that complied with the selection criteria. In this example, three events complied with the input criteria. The NMTs that recorded all 3 events were NMT# 2, 4 and 6. The error averages calculated for these three common NMTs, is a statistically true average. The other NMTs, NMT# 11, 14 and 15 were each unique to one of the three events. The code currently does not have the ability to differentiate between NMTs that are triggered by all events and those that are not. The averages for NMT# 11, 14 and 15 are reduced to one third of the actual value of the error and therefore, are statistically incorrect. This glitch is also discussed in Section 7.3.

The result is represented graphically in Figure 5.5. For most NMTs, err2_avg is less than err1_avg, which means that the results from Case 3 of the NORD2000 model which includes the effect of meteorological data, has a better agreement with measured sound levels, than the results from Case 1 of the NORD2000 model which assumes a homogeneous sound speed profile.
6.3.2 Approach events on July 1, 2014 at 26R (north runway) triggering 6 NMTs.

This run was selected as an example because one of its events loops around and triggers a few NMTs twice. Figure 6.3 shows the Google map with flight path.

![Google map with flight path](image)

**Figure 6.3:** Arrivals on the north runway (26R) triggering 6 NMTs

In the NMT data, the event that loops around had multiple records for NMT#14. Both records were 15 minutes apart and had different SEL values, but the same $L_{\text{max}}$ value and duration. The code was incapable of identifying multiple passes of the same event over the same NMT. It considered the closest slant height over the entire path of the event, which is a caveat.

The aim was to check the difference between results of Case 1 and Case 3 of NORD2000. The average of errors for NMT# 13, 14 and 15 were lesser in magnitude for the NORD2000 log-linear sound speed profile case, than the corresponding average of errors for a homogeneous sound speed profile case. This meant that the results for Case 3 of NORD2000 agreed better with measured data than the results for Case 1 of NORD2000 for these three NMTs out of the six NMTs that were triggered. This is an indication that using meteorological data in NORD2000 gave better than using a homogeneous sound speed profile. The results for NMT# 6 and 11 did not show the same trend.
Table 6.5: Erortab for approach events on 26R triggering 6 NMTs

<table>
<thead>
<tr>
<th>NMT#</th>
<th>Error for Event 1</th>
<th>Error for Event 2</th>
<th>Mean of errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>err11 err21</td>
<td>err12 err22</td>
<td>Err1_avg Err2_avg</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>-3.7</td>
<td>-5.7</td>
</tr>
<tr>
<td>6</td>
<td>-18.8</td>
<td>-0.6</td>
<td>-17.6</td>
</tr>
<tr>
<td>11</td>
<td>6.7</td>
<td>0</td>
<td>-0.6</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>-6.7</td>
<td>-12.2</td>
</tr>
<tr>
<td>14</td>
<td>-14</td>
<td>-12</td>
<td>-12.2</td>
</tr>
<tr>
<td>15</td>
<td>-2.2</td>
<td>-1.2</td>
<td>-5.4</td>
</tr>
</tbody>
</table>

Figure 6.4 shows a graphical representation of results from Table 6.5. It was observed that the results for NMT# 11 were over predicted while the results for all other active NMTs were under predicted. It is easier to visually understand the mean of errors for Case 3 of NORD2000, marked in red circles are closer to the the zero error line than the blue squares that represent the mean error for Case 1 of NORD2000 for NMT# 13, 14 and 15.

Figure 6.4: This plot serves as a graphical representation of erortab showing the differences between measured and calculated $L_p$ for approach events on runway 26R for 6 noise monitors triggered on July 1, 2014.
6.3.3 Departure event on September 27, 2014 at 08R (south runway).

It was observed for many departure type operations that the Case 3 and Case 1 NORD2000 results were both highly overpredicted. Upon investigation it was inferred that this could be because of the maximum thrust level adjustment of source spectra as explained in Section 4.4, which increases the source level spectra by almost 10 dB over the unadjusted source spectra. This section discusses one such example of a single event analysis of a departure event from 08R runway end on September 27, 2014. The path for this event is shown in Figure 6.5.

![Figure 6.5: Departure from south runway (08R). Event correlation ID 15656674](image)

The aircraft associated with this event was an Airbus A319 belonging to the departure profile spectral class SPECT_DEP 103. Figure 6.6 shows the source levels over a frequency range of 50 Hz to 10 kHz. The blue line represents the input spectrum adjusted for the maximum thrust setting and the red line represents the input spectrum without NPD thrust setting adjustment.
Figure 6.6: A-weighted source spectra for Departure Class 103 without adjustment and with maximum thrust NPD adjustment at 1 m from the source.

The analysis for this event was performed with two different source spectra inputs to understand the difference in results due to maximum thrust level adjustment. Comparisons between measured and calculated overall sound pressure levels were made for Case 1 and Case 3 of NORD2000, using an input source spectrum that was un-adjusted and using an input source spectrum adjusted for maximum thrust setting, which was a default procedure implemented for all analysis throughout the third phase of this project.

The errorplot in Figure 6.7 shows two sets of results. The red and blue pair are the results for adjusted source spectra and purple and yellow pair are the results for un-adjusted spectra. It was observed that for all active NMTs, the overall sound pressure values calculated using the maximum thrust NPD data were higher than the results obtained from the same case (Case 1 or Case 3) of the NORD2000 model, calculated using the un-adjusted input spectrum. This trend was seen consistently for the departure profile results.
Figure 6.7: This plot serves as a graphical representation of errortab showing the differences between measured and calculated $L_p$ for a departure event from runway 08R on September 27, 2014 for two different input spectra. One adjusted for the maximum thrust NPD and another is the un-adjusted spectrum.

### 6.4 Overall observations

This study focused on examining the trends seen in single events with similar characteristics. No attempt to average between multiple events, such as overall daily averages, were made. Nevertheless, some trends are apparent.

First, it was noticed that if the input source spectrum used was either retrofitted as discussed in Section 5.1.2, or if an overall sound power level was used instead of a spectrum along with a homogeneous sound speed profile, the calculated predictions had a poor agreement with the measured NMT data. Secondly, the agreement between measured and calculated data seemed to be somewhat improved if the specific aircraft input spectrum calculated as per Section 4.4 was used, still maintaining a homogeneous sound speed profile. Finally, even better overall results were seen if the correct source spectrum was used along with real atmospheric temperature and wind data. The improvement was prominent if the temperature and wind speed profiles were strong. These observations are summarized in the Figure 6.8
<table>
<thead>
<tr>
<th><strong>Input spectrum</strong></th>
<th><strong>Input atmosphere</strong></th>
<th><strong>NMT Vs. prediction</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall SWL Back-calculated from NMT SEL</td>
<td>Const Temp &amp; No wind = Homogeneous $c(z)$</td>
<td>Poor agreement</td>
</tr>
<tr>
<td>1/3rd octave SPL Correct aircraft i/p spectrum from AEDT database</td>
<td>Const Temp &amp; No wind = Homogeneous $c(z)$</td>
<td>Satisfactory agreement</td>
</tr>
<tr>
<td></td>
<td>Temp gradient &amp; Wind data = Log-Lin $c(z)$</td>
<td>Overall Better Agreement</td>
</tr>
</tbody>
</table>

Figure 6.8: Summary of observations
Chapter 7  
Conclusions

7.1 Conclusions

The data from the Vancouver International Airport was used for a single event aircraft noise analysis. The methodology and codes used to extract information could be replicated for a dataset from any airport in the world. The understanding gained in learning the process was valuable.

In most of the cases, especially the cases involving approach type operations, the results were better when real atmospheric data was used when compared with results obtained with a homogeneous atmospheric profile. For the events analyzed at Vancouver, the wind speeds and temperature gradients were not too strong and their effects in the progressive change in each NORD2000 case were not always dominant. However, significant changes could be expected in situations where a very strong wind profile and temperature profile existed. In most cases, the results showed a significant change in error due to the inclusion of atmospheric effects.

It was very critical to use the source spectrum data appropriate for the aircraft corresponding to the event. The data provided did not consist of real thrust setting information due to which, maximum thrust settings were assumed for calculation of source input specta. Although, having this information might help in getting more accurate calculation predictions.

7.2 Caveats and assumptions

Before discussing the scope of further research related to this study, the highlights of the assumptions in the present project are provided:

1. The NMT elevations are approximated and a realistic topology is not considered.

2. Crosswinds are neglected, although they can play a major role.

3. It is assumed that all departures are eastward and all arrivals are westward. In reality, the flight paths are always at an angle to east-west. There is a rare chance that this might not be true for a few cases and those cases are not analyzed.
4. The aircraft are assumed to be omnidirectional point sources in the free field. There is data available on directivity of aircrafts in the Aviation Environmental Design Tool (AEDT) database [11].

5. It is assumed that there is no background noise.

7.3 Future work

Currently, the MATLAB code is programmed to compute the mean error. This method is a good for NMTs that are repeated over all events that qualify in the same multiple events run. If there are 15 events that comply with the input criteria for event selection, each triggered 5 NMTs and only 3 of those were triggered by all 15 events; the method to calculate the mean error would add all the errors and divide the sum by 15. This is not a good method to analyze the other 2 unique NMTs that were triggered by only one or a few of the 15 events. MATLAB cannot process NaN (not-a-number) values in the matrix, which is why they were replaced with zeros, and yet, while calculating the mean, the sum is divided by 15 rather than the actual number of events that triggered the NMT being analyzed. The code could be improved to correct this issue.

The departure operation type events are the loudest among the three operation types. As discussed in Section 6.3.3, it was observed that the agreement between measured and predicted values was better when an un-adjusted input source spectrum was used as compared to the case when the input source spectrum was adjusted to the maximum thrust setting. Using the maximum thrust setting is not an ideal or realistic scenario. Obtaining a real thrust setting for use in the analysis could improve the results in the future predictions.

The measured NMT data was available in Sound Exposure Level (SEL) and maximum sound pressure level ($L_{\text{max}}$). The equivalent sound pressure level ($L_{\text{eq}}$) could be calculated from the SEL level using the duration of measurement using Equation 4.1. The metric chosen to compare measured data was $L_{\text{max}}$ however, the entire process could be tried with the SEL metric or $L_{\text{eq}}$.

The methodology implemented to pick the time window and computation of the temperature gradient slope and wind direction (see Section 4.5.1) made a few assumptions such as a six hourly data was considered rather than an hourly data, the crosswinds were neglected and the wind direction was assumed to be perfectly East-West. It was also observed that the cut-off height for consideration of the atmospheric column could also change the input parameters to the NORD2000 code. Improving the accuracy of these assumptions could help in obtaining better comparable results.

Currently, the results from different multiple event analysis runs are not compared with one another. Although, a statistical analysis of the results from multiple events might show a trend in errors at particular NMTs. Any trends can be then correlated to NMT specific data or help us identify input factors that need to be refined.

For the purpose of this project, only the NORD2000 propagation model was used as it is perhaps the simplest method available to include the effects of refraction due to wind and temperature. Future work might involve the use of ray-trace or parabolic equation models. The comparison of the predicted results from the propagation models to the simulated results from
AEDT and the comparison of measured NMT data to simulated AEDT results could help verify the prediction models [14]. Comparison with AEDT will also help in analyzing the difference in results when considering the directivity of the aircraft.
Appendix A
Accessing the AEDT database using SQL

This Appendix chapter provides information on a few basic SQL commands that will be useful for accessing information from the database tables discussed in Chapter 4. It also describes the procedure for modifying the track co-ordinated in AEDT through the SQL database management software, which might be useful in the future when AEDT would be used for analysis of aircraft events with radar information.

A.1 Accessing database tables

AEDT version 2c., is built on the Microsoft .NET Framework. It is supported by a global database of airports, airspace and fleet information [11]. The central data architecture of AEDT relies on a relational database management system, Microsoft SQL Server 2008 R2. The three databases as explained in section 2.2 can be accessed using the SQL Server Management Studio using the following procedure:

1. Upon staring the SQL Server Management Studio, connect to the AEDT server by browsing the list for ‘server name’. It appears in the format ‘ACS-AIRPORTNOIS\AEDT’.
2. Click on ‘new query’, which opens a command window where we can type out the code.
3. Refer to the Fleet Database Description Document (Fleet DDD) [12] which consists of details about the information available in each table. Note the names of the tables required.
4. In the new query, type the short code given below
   
   Select *
   from [FLEET].[dbo].[table name]
   
   The * is equivalent to ALL columns. It may be replaced by specific column names in [square brackets] to hide rest of the columns in the table when the code is executed.
5. Click in ‘!Execute’ to compile the command.
The fleet database tables which were used in the YVR study were copied to a single Microsoft Excel spreadsheet, ‘SpectralClassData.xlsx’ for convenience. Although, an alternative way could also be to integrate SQL and MATLAB and extract information from the database tables directly.

A.2 Define radar data rack in AEDT

It is practically not feasible to input a long flight path with many coordinate locations into AEDT manually since the latitude, longitude and height would need to be typed manually. There are ways to export the coordinates of the modeled track to an Excel sheet, although any edits/additions made to that sheet do not reflect back in the track data. All information regarding tracks modeled in AEDT is stored and made available in the SQL database [28].

1. One can use the Microsoft SQL Server management system to add/edit tracks. Here are some simple steps to follow. In AEDT version 2c, create a track using the following commands

(a) Create a new study OR import an existing study.
(b) Check if the desired airport has been imported. If not, click on the ‘Airports’ tab and add an airport to the study.
(c) Click on the ‘+/-’ sign in the ‘airports’ window and a new tab, ‘[Airport Name/code] Default Layout’ will appear as shown in Figure A.1.

![Figure A.1: Snapshot of the user interface for the airport selection window in AEDT](image)

(d) Right click on this tab and click on ‘Design’. A tool window appears as shown in figure A.2
Figure A.2: Snapshot of the user interface for airport layout design toolbar

(e) Click on ‘Add Dep/Appr Track’ to create a new departure or approach event track.

(f) Specify at least the first 2 coordinate points and assign a significant and identifiable name to the track.

2. Open Microsoft SQL Server Management Studio 2008 R2 and make sure the database is connected in SQL. Use the syntax specified in section A.2, access the information from tables APT_TRACK, APT_SUBTRACK and APT_SEGMENT in the order mentioned below. Note that these tables are not in the fleet database so we replace FLEET with the name of the AEDT study.

(a) Referring to the study database table APT_TRACK, get the TRACK_ID corresponding to the TRACK_NAME of interest. Note that the TRACK_TYPE would be ‘P’ i.e. Point type input.

(b) Referring to the table APT_SUBTRACK, get the SUBTRACK_ID corresponding to the TRACK_ID.

(c) Explore the table APT_SEGMENT using the same syntax. The table will contain the coordinates of the first two points that were used to define the track. Each point will have a SEGMENT_ID which is unique for every point.

3. Create a table in Microsoft Excel with the exact same sequence of columns and fill the rows such that it would be a continuation to the existing APT_SEGMENT table. If two co-ordinate points were initially defined, start the table with the third point. Also assign the appropriate SEGMENT_ID. Figure A.3 illustrates the APT_SEGMENT table. Only the first 7 columns are of importance to model the flight track.

Figure A.3: Snapshot of the APT_SEGMENT table in SQL

Points to be remembered while creating the Excel sheet are:

(a) The Excel sheet to be created must start with a SEGMENT_ID number that would follow the last entry in that column for the desired SUBTRACK_ID i.e. 2218 in this case.
(b) The SUBTRACK_ID is unique to a track, while the SEGMENT_NUM and the SEGMENT_ID increments by 1 for each co-ordinate of the same track.

(c) The SEGMENT_TYPE will be ‘P’. PARAM_1 is the latitude co-ordinate column and PARAM_2 is the longitude co-ordinate column.

(d) The first six columns can not have a NULL value. ALTITUDE can however be edited later if such data are unavailable since this column accepts NULL values.

(e) SEGMENT_ID is auto generated. SQL remembers the last SEGMENT_ID which was assigned to a segment, even if the segment was immediately deleted. That is the reason SEGMENT_ID jumps from 4 to 2216 in the snapshot of the segment table shown in Figure A.3. No matter what SEGMENT_ID we try to create, SQL will overwrite it. This won’t affect the modeling of our track. Although, not including some SEGMENT_ID might create avenues for errors.

4. After the table to be appended is created in MS Excel, the existing APT_SEGMENT table can be modified. Right click on the table in Object Explorer and click on ‘Edit Top 2000 Rows’. The default number of rows that can be edited is 200. This can be changed to any desired value by assigning a higher value to the field ‘Value for Edit <n> Rows command’ as shown in Figure A.4.

![Figure A.4: Snapshot of the SQL user interface to modify the number of editable rows in a table.](image)

5. Once values are appended to the table as shown in Figure A.5, it looks the same as it would have been in case we were to input each point in AEDT manually.
6. The modifications are then saved and the result can be checked in AEDT. The track which initially had only two points must now have been updated. Figure A.6 shows one such example.
Figure A.6: Snapshot of an AEDT study showing the radar data flight track for a particular event that corresponds to the DISCOVER-AQ acoustic dataset.
Appendix B

The Great Circle Method

The radar data consists of x,y,z locations of an airplane in reference to a fixed point. It is not necessary that the co-ordinates of this point are the airport reference. In the case of the YVR radar data, the radar reference is approximately 15 km to the North-East of the airport island. In order to plot the tracks over a Google map and to find the distances between location of aircraft and the NMT locations, a simpler version of the great circle method was used to calculate the distances between the aircraft and the receivers [29]. The following procedure discusses the assumptions and the theory behind this calculation methodology.

1. The great circle method is used to find the shortest distance between two points on the surface of a sphere, measured along the surface of the sphere.

2. It is assumed that the earth is a perfect sphere and the slightly ellipsoidal shape is ignored. It is assumed that the lateral distance between two longitude lines decreases evenly as the latitude increases, or as we move towards the poles from the equator. This relationship is simplified to a cosine function as shown in the equation B.2

3. At the equator, 1° longitude and 1° latitude equals to 60 nautical miles. In the equations B.1 and B.2, $X_0$ and $Y_0$ are the longitude and latitude co-ordinates of the radar reference point or the map reference point (mrp) as mentioned in Table 4.1. The $x$ and $y$ values are distances in meters from the mrp and are given in the radar data. The coordinates in degrees can be calculated using simplified haversine formula [30]:

\[
\text{Latitude coordinate} = Y \text{ coordinate} = Y_0 + \frac{y[m]}{60 \times 1852} \quad (B.1)
\]

\[
\text{Longitude coordinate} = X \text{ coordinate} = X_0 + \frac{x[m]}{60 \times 1852 \times \cos(Y \text{ coordinate})} \quad (B.2)
\]

Note that there are exactly 1852 meters per nautical mile. Also, the nautical mile is a minute or on-sixtieth of a degree of latitude.

4. The height, $z$ is readily available in the radar data.
5. The same formula are modified to calculate $x$ and $y$ distances in [m] when the coordinates are known and distance is to be calculated.
Appendix C
MATLAB codes

C.1 Main execution code: ExecutionFileYVR.m

This code is partitioned in sections. The description for each section is given in section 5.1.3.

```matlab
%% SECTION 1
% This program plots tracks after extracting data from radar file
% Author - Manasi Biwalkar
% Fall 2016 & Spring 2017
% Import Excel file.
addpath('C:\Users\mxb1096\Documents\Research_material\YVR Noise Analysis ... project\RadarDataFiles');
datafile = 'RadarDataSep27.xlsx';
NMTdata = 'NMTSep2014.xlsx';
% Take in all data in one big matrix format
[rad_num,rad_txt,rawData] = xlsread(datafile); %rawData is a matrix of the ...
dataFile
CFSR_YVR = load('C:\Users\mxb1096\Documents\Research_material\YVR Noise Analysis ... project\RadarDataFiles\WeatherData\CFSRreq240638\CFSv2YVR2014.mat');
%% SECTION 2 --> Making of RCONSTRUCT matrix
% nx9 matrix : n - Number of identified tracks
% Columns Track ID, CorrID, Operation type, Runway, PointOfInterest(1st pt
% for D and O | LastPt for A), X, Y, Z and Speed at PointOfInterest
%% Look for the word 'TRACK' and find the index
idx_track = find(strncmp('TRACK', rawData,5)); %Precision to 5 characters
idx_CorrID = idx_track+1; %
idx_op = idx_track+11; %index for Operation type. Found in txt (D/A/O)
idx_runway = idx_track+14; %Found in txt
idx_1pt = idx_track+20; %1st point of track Found in num
```
idx_count = idx_track+19;  %Number of points in the event
idx_lastpt = idx_lpt+cell2mat(rawData(idx_count,1))-1;

date = rawData(idx_track+2);
EvStTime = datenum(cell2mat(rawData(idx_track+3)));  %Event Start Time acc to radar
EvEnTime = datenum(cell2mat(rawData(idx_track+4)));  %Event End Time acc to radar
EvDuratn = datenum(EvEnTime) - datenum(EvStTime), 'HH:MM:SS');

AircraftMod = rawData(idx_track+8);  %Aircraft Model
AircraftTyp = rawData(idx_track+9);  %Aircraft Type P/B/J/H

RDConstruct=[rawData(idx_track),rawData(idx_CorrID),rawData(idx_op),...
  rawData(idx_runway),rawData(idx_lpt,5),rawData(idx_lpt),...
  rawData(idx_lpt,2),rawData(idx_lpt,3)];
%
K,rawData(idx_lpt,4)

% in RDConstruct, rawData(idx_lpt,1 or 2) is the X_m and Y_m distance for
% 1st point in the track. To ensure that the first point is selected,
% rawData(idx_lpt,5) is added to the table. this point is the 1st time
% stamp. So, it has to be zero. Check it!

check=cell2mat(rawData(idx_lpt,5));  %Its the matrix (nx1) of time stamps of 1st point.
find(check)  %Will return index of non zero elements in X. If all values are 0 ...
  (desired);
% ans will be an empty matrix
fprintf('If you got Empty matrix, everything is GOOD!')

% We don't need the 1st point of Arrival tracks. We need the LAST point of
% arrivals. So, we update RDConstruct.

idx_a=find(strncmp('A', RDConstruct(:,3),1));  % Find index of all indexes of ...
% Arrivals in RDConstruct

for ii=1:length(idx_a)
  RDConstruct(idx_a,6)=rawData(idx_lastpt(idx_a),1);  %X
  RDConstruct(idx_a,7)=rawData(idx_lastpt(idx_a),2);  %Y
  RDConstruct(idx_a,8)=rawData(idx_lastpt(idx_a),3);  %Z
  RDConstruct(idx_a,5)=rawData(idx_lastpt(idx_a),5);  %timestamp
end

% In case we want to save .xlsx sheet of RDConstruct. Might need to
% include/exclude last header depending on whether direct distance was
% calculated.
% heading={'Marker','Corr ...
  ID','Operation','Runway','timestamp','X_m','Y_m','Z_m','speed','dir_dist'};
% RDConstruct= [heading;RDConstruct];
% xlsxwrite('RDConstruct.xlsx',RDConstruct)
%
% SECTION 3 --> Making of NGConstruct

[EQUIP_num,EQUIP_txt,FLT_EQUIP] = xlsread('SpectralClassData.xlsx',1);
[ARPL_num,ARPL_txt,FLT_ARPL] = xlsread('SpectralClassData.xlsx',2);
[NID_num,NID_txt,FLT_NOISGRP] = xlsread('SpectralClassData.xlsx',3);
[temp,temp,LwData] = xlsread('SpectralClassData.xlsx',4);

idx_Elig = find(strcmp('J',ArcrftTyp));
z=num2cell(zeros(length(idx_Elig),1));
NGConstruct = [num2cell(idx_Elig), ArcrftMod(idx_Elig,1), z, z, z, z];

for ii=1:length(idx_Elig)
    idx_bada = ...
          find(strncmp(char(NGConstruct(ii,2)),EQUIP_txt(:,1),length(char(NGConstruct(ii,2)))));
      if isempty(idx_bada)
          continue
      end
    AIRPLANE_ID = cell2mat(FLT_EQUIP(idx_bada(1)+1,5));
    NGConstruct(ii,3) = FLT_EQUIP(idx_bada(1)+1,5);
    if isnumeric(AIRPLANE_ID)
        idx_arpID = find(ARPL_num(:,1)==AIRPLANE_ID);
        NOISE_ID = FLT_ARPL(idx_arpID+1,14);
    else
        idx_arpID = find(strncmp(AIRPLANE_ID,FLT_ARPL(:,1),length(AIRPLANE_ID)));
        NOISE_ID = FLT_ARPL(idx_arpID,14);
    end
    if length(NOISE_ID)>1
        NOISE_ID = NOISE_ID(1);
    end
    NGConstruct(ii,4) = NOISE_ID;
    idx_ng = find(strncmp(char(NOISE_ID),NID_txt(:,1),length(char(NOISE_ID))));
    SPECT_APP = FLT_NOISGRP(idx_ng(1),2);
    NGConstruct(ii,5) = FLT_NOISGRP(idx_ng(1),2);
    SPECT_DEP = FLT_NOISGRP(idx_ng(1),3);
    NGConstruct(ii,6) = FLT_NOISGRP(idx_ng(1),3);
end
disp('Noise group data is compiled!')

% SECTION 4 --> This section creates a list of events of the same operation type ...
% from the same runway
runway=input('Select a runway end. South (08R,26L) North (08L,26R) : ','s' );
da=input('Dep=1, Arr=2, Overflight=3{runway is immaterial}. Select operation : ');

if da==1;
    operation='D';
elseif da==2;
    operation='A';
elseif da==3;
    operation='O';
else
    sprintf('Incorrect da input!')
    return
end

% Method#1
% Find index numbers from RDConstruct that are a specific operation and from
if da==1;
    idx_d = find((and(strncmp(operation, RDConstruct(:,3),1),strncmp(runway, ...
        RDConstruct(:,4),3)),cell2mat(RDConstruct(:,2))));
    idx_active=idx_d;
elsif da==2;
    idx_a = find((and(strncmp(operation, RDConstruct(:,3),1),strncmp(runway, ...
        RDConstruct(:,4),3)),cell2mat(RDConstruct(:,2))));
    idx_active=idx_a;
elseif da==3;
    idx_o = find(and(strncmp(operation, RDConstruct(:,3),1),cell2mat(RDConstruct(:,2))));
    idx_active=idx_o;
end
sprintf('% Number of events for this operation type from selected runway end = ...
    \%g',length(idx_active))
X_md=cell2mat(RDConstruct(idx_active,6));
Y_md=cell2mat(RDConstruct(idx_active,7));

% Radar Reference coordinate
X_o = -123.001297;
Y_o = 49.252534;

% Interest points : are 1st points for Departures. last points for arrivals
Y_intrstPts = Y_o+(Y_md./(60*1852));
X_intrstPts = X_o+(X_md./(60*1852.*cosd(Y_intrstPts)));

NMT = [ 49.175020, -123.152601, 18 ;
        49.186098, -123.150813, 4 ;
        49.166899, -123.164517, 4 ;
        49.183261, -123.115452, 4.5;]
        49.189229, -123.085043, 4.5;]
        49.171005, -123.056888, 4.5;]
        49.234280, -123.176844, 4.5;]
        49.217671, -123.152957, 10;]
        49.274834, -123.236714, 5.5;]
        49.209830, -123.136489, 15;]
        49.196551, -123.111156, 12;]
        49.197399, -123.205339, 5.5;]
        49.206136, -123.179256, 4.5;]
        49.169056, -123.902371, 12;]
        49.157465, -123.905125, 5.5;]
        49.219994, -123.967796, 4.5;]
        49.138023, -123.141789, 6;]
        49.016830, -123.075767, 6;]
        49.152834, -122.816857, 5.5;]
        49.038085, -122.876015, 6 ]; % (;,3) is in m

% Mean of all interest points
x_Ipt_m=mean(X_intrstPts); y_Ipt_m=mean(Y_intrstPts);

% SECTION 5 --&gt; Analyze SINGLE events
% Find by nth event way
% In the list of departure tracks from a specific runway, chose 'n'th event
n=1;
current_track=idx_active(n);
%Finds index number of the current track in RDConstruct
corrID = cell2mat(RDConstruct(current_track,2)) %Finds correlation ID for that track
% % Find by corrID
% corrID = 15446331
% n = find(idx_active==find(cell2mat(RDConstruct(:,2))==corrID)) %This corrID is ...
in the nth row of RDConstruct

% Get Spectral Class data for current track (lw). Length of lw must be
% equal to length of f.
% 10^1.7 = 50 Hz; 10^-3.3 = 2kHz; 10^-4 Hz (full available spectrum)
% Define frequency and frequency spectra
bno = 17:40;
f = 10.^((bno/10);
if ~ismember(current_track,cell2mat(NGConstruct(:,1)))
disp('LW data unavailable')
return
else
idx_noiseID = find(cell2mat(NGConstruct(:,1))==current_track);
if da==2
    idx_sc = find(cell2mat(LwData(:,1))==cell2mat(NGConstruct(idx_noiseID,5)));
else
    idx_sc = find(cell2mat(LwData(:,1))==cell2mat(NGConstruct(idx_noiseID,6)));
end
% 3:end if bno=17:40 | 3:19 if bno=17:33
lp_sc = cell2mat(LwData(idx_sc,3:end));

AirAtt = [3.3 3.3 3.3 6.6 6.6 9.8 13.1 13.1 19.7 23 29.5 36.1 45.9 59 75.4 98.3 ... 131.1...
170.5 229.5 311.5 360.7 524.6 721.3 983.6]/100;
% Reference spectrum corrected to remove air attenuation Eq. 10-41 (Pg258 AEDT Tech ... manual)
lp10_41 = lp_sc + AirAtt.*3.05;
addpath('C:\Users\mb1096\Documents\Research_material\YVR Noise Analysis ... project\RadarDataFiles\Codes');
WA = AWT(f);
% Corrected spectrum adjusted to 1m from S using reference atmosphere. Eq. 10-42 ...
(Pg258 AEDT Tech manual)
lp_spectrumA = round(10.*{lp10_41 - (20*log10(1/305)) - (AirAtt/100) + WA /10;\l}
lp_spectrum = round(10.*(lp10_41 - (20*log10(1/305)) - (AirAtt/100)))/10;
lw_sc = lp_spectrumA +20*log10(1)+11; %If we want equivalent Power spectra
% ----------------------------------------------------------Adjust for normalization------------------------
cur_NoiseID = NGConstruct(idx_sc,4);
adj_LpscA = DeNormalize(operation,cur_NoiseID,lp_sc);
if adj_LpscA == 0

71
disp('NFD data unavailable. This run is processed using Un-adjusted input ... spectrum');
  ipL = lp_spectrumA;

else
  temp1 = adj_lpscA + AirAtt.*3.05;
  ADJlpAlm = round(10.*(temp1 - (20*log10(1/305)) - (AirAtt/100))) / 10;
  semilogx(f,ADJlpAlm,'b'); hold on; semilogx(f,lp_spectrumA,'r');
  xlim([min(f) max(f)])
  xlabel('Frequency [Hz]'); ylabel('SPL [dBA]')
  title('A weighted SPL at 1 m from aircraft; 'Adjusted with maximum NFD thrust ... setting and without adjustment'))
  legend('Adjusted spearctrum', ' Without adjustment', 'Location', 'Best')
  ipL = ADJlpAlm;
  % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  end
  ipL_unadj = lp_spectrumA;
  % ipL = ADJlpAlm;

end

% % Default spectra from K. Pouline's code. OVERRIDE from spectral class
% bno = 17:33;
% f = 10.^(bno/10);
% lw = [116 121 130 134 136 140 141 145 147 149 150 150];
% ipL = lw;

cur_date = datenum(date(idx_active(n))); %num format. interested in floor(cur_date)
cur_time = datenum(EvStTime(idx_active(n),:)); %num format. Interested in digits ... after decimal pt.
% cur_dt = [datestr(cur_date), ' ', datestr(cur_time)]; %string format of date and time
z=['12:00 AM'; '06:00 AM'; '12:00 PM'; '06:00 PM']
tzPick = cur_time - floor(cur_time);
if tzPick<0.125 %Upto 3:00 AM
  tz=1;
else tzPick<0.375 % 3 AM to 9 AM
  tz=2;
else tzPick<0.625 % 9 AM to 3 PM
  tz=3;
else tzPick<0.875 % 3 PM to 9 PM
  tz=4;
else % After 9 PM, take weather data from next day's 12 AM
  tz=1;
  cur_date=cur_date+1;
end
% % Finds the unique values. Column #1
lookup_table = unique(NMT_num(:,1));
rows = find(NMT_num(:,1)== corrID);
active_NMT = NMT_num(rows,2);
SEL = NMT_num(rows,4);
Lmax = NMT_num(rows,5);

NMTtau = NMT_num(rows,7);
Leq = round(10*(SEL - 10*log10(NMTtau)))/10;
data= cell2mat(rawData(idx_lpt(idx_active(n)):idx_lastpt(idx_active(n))-1,1:5));
X_m = data(:,1); %in meters
Y_m = data(:,2); %in meters
Z_m = data(:,3); %in meters
vel = data(:,4); %in meters/sec
tm = data(:,5); %in sec

% Original radar track for corr_ID(n)
Y = Y_o+(Y_m./(60*1852));
X = X_o+(X_m./(60*1852.*cosd(Y)));

sprintf('Radar data available, track is updated!')

% Get weather data for target time window.
idx_cfsr = find(datenum(CFSR_YVR.t) == datenum(cur_dt)+1);
% +1 will access the next 6 hr time window in CFSR data set to account for UTC-7:00 ...
time zone.
% To get clear idea refer UTC_PST.mat. Convert to datesr to check the time zone
Zg_mean = mean(CFSR_YVR.Zg,1); % Mean elevation from sea level [m]
idx_isoT = find(Zg_mean < 1000);
Tgrad = CFSR_YVR.T(idx_cfsr,idx_isoT)-273.15;
Zgrad = Zg_mean(1,idx_isoT)./10^3;
RH = CFSR_YVR.Hs(idx_cfsr,1)*10000
% figure()
% plot(Tgrad,Zgrad)
% titleesub('Temperature gradient for ',cur_dt)
S1Int=polyfit(Tgrad,Zgrad,1);
dtz=SlInt(1)
idx_isoW = find(Zg_mean < 1000);
Wgrad = CFSR_YVR.U(idx_cfsr,idx_isoW);
Zgrad = Zg_mean(1,idx_isoW)./10^3;
u = Wgrad(end)
z = Zgrad(end)*10^3;
LpSS = zeros(1,length(active_NMT)).';
horzdist = LpSS; Sht = LpSS; CP = LpSS; LpN2kc1 = LpSS; LpN2kc2 = LpSS; LpN2kc3 = LpSS;
LpN2kc1_unadj = LpSS; LpN2kc3_unadj = LpSS;
addpath('C:\Users\mxb1096\Documents\Research_material\YVR Noise Analysis ...
project\RadarDataFiles\Codes');
[LpSS,horzdist,Sht,CP] = SphSpr(active_NMT,NMT,X,Y,Z_m,f,ipL,RH,Tgrad(1));
% LpAedt = AEDTmet(active_NMT,NMT,X,Y,Z_m,f,ipL0_41,AirAtt);
if or((and(u<0 , da==1), and(u>0 , da==2))
   u=-u
end
for kase = 1:3
    if kase==1
        LpN2kc1 = RunCompro(ipL,NMT,kase,active_NMT, horzdist,Sht,f,RH,Tgrad(1));
        LpN2kc1_unadj = RunCompro(ipL_unadj,NMT,kase,active_NMT, ...
                          horzdist,Sht,f,RH,Tgrad(1));
    else
        dtdz = input('dtdz = ');  
        if kase==2
            LpN2kc2 = RunCompro(ipL,NMT,kase,active_NMT, ...
                                horzdist,Sht,f,RH,Tgrad(1),dtdz);
        elseif kase == 3
            u = input('wind speed component in direction of propagation (+ ... 
                      downwind, - upwind) = ');  
            zu = input('height of measured wind = ');  
            LpN2kc3 = RunCompro(ipL,NMT,kase,active_NMT, ...
                                horzdist,Sht,f,RH,Tgrad(1),dtdz,u,zu);
        end
    end
end

LpTable=zeros(length(active_NMT),9);
LpTable=[active_NMT,horzdist./1000,Sht./1000,CP./1000,SEL,Lmax,Leq,LpSS,LpN2kc1,LpN2kc2,LpN2kc3];
tableADJ =0;
if adj_LpscA==0
    tableADJ = LpTable;
end
LpTable = real(LpTable);
LpTable_unadj=zeros(length(active_NMT),9);
LpTable_unadj=[active_NMT,horzdist./1000,Sht./1000,CP./1000,SEL,Lmax,Leq,LpSS,LpN2kc1_unadj,LpN2kc2,LpN2kc3_unadj];
LpTable_unadj = real(LpTable_unadj);
figure();
scatter(active_NMT(:,1),LpTable(:,9)-LpTable(:,6),'filled','s','MarkerEdgeColor','k'); ...
    err1 blue square
hold on
scatter(active_NMT(:,1),LpTable(:,11)-LpTable(:,6),'filled','MarkerEdgeColor','k'); ...
    err2 red circle
scatter(active_NMT(:,1),LpTable_unadj(:,9)-LpTable_unadj(:,6),'filled','s','MarkerEdgeColor','k'); ...
    err1 blue square
scatter(active_NMT(:,1),LpTable_unadj(:,11)-LpTable_unadj(:,6),'filled','MarkerEdgeColor','k'); ...
    err2 red circle
xlim([min(active_NMT)-1 max(active_NMT)+1]);
grid on
xlabel 'NMT number'
ylabel '{\Delta L_p}\_{L_p}''
title('Comparison Calculated L_p for Case 3 N2k: With and without NPD adjustment');

% Now we break X-axis to get fancy!
% Decide up to where to break axis?? Break at 6 for sure.
% Start again l-NMT# that appears first while ascending from 7 through 13
startX = 5+ find(ismember([7,8,9,10,11,12,13,14,15],active_NMT),1);
breakaxis([? startX]);

% Plots and graphs for Single event
% figure(2)
hold all
plot(X,Y,'-b', 'MarkerSize',5)
plot(X_o,Y_o,'*r', 'MarkerSize',8)
NMT_marker = plot(NMT(active_NMT,2),NMT(active_NMT,1),'-b','MarkerSize',10);
str = num2str(round(SEL));
str = num2str(active_NMT);
text(NMT(active_NMT,2),NMT(active_NMT,1),str)
plot_google_map
alpha(.5)

if da==1;
  op='Departure from';
elseif da==2;
  op='Arrival at';
elseif da==3;
  op='Overflight no';
runway=' ';
end

str= sprintf('Corr ID: %u, %s runway %s,corrID,op,runway);
strl=sprintf('Position of track relative to Radar Ref');
title({str,str1},'fontsize',12)

% SECTION 6 --> Analyze MULTIPLE events
skippedForNMT = 0;
% Intergrated with functions SphSpr.m and RunCompro.m

% Initialize variable UsableEvents = zeros (l x # of events to be analyzed)
% The zero at nth event will be replaced by corrID only if it satisfies the
% xx number of NMTs needed for that run
UsableEvents=zeros(length(idx_active),1);

% Duplicate NMT coordinate matrix. This is created so that NMTs aren't
% overwritten while plotting
dynamicNMT=NMT;

% Criteria for this run is enlisting events that trigger exactly xx NMTs
xx=input('Number of NMTs as a selection criteria : ');

% Initialize variables to be used later
LpSS = zeros(length(idx_active),xx); horzdist = LpSS; Sht = LpSS; LpN2kcl = LpSS;
LpN2kc2 = LpSS; LpN2kc3 = LpSS; SEL = LpSS; Leq = LpSS; trigNMT = LpSS; CP = LpSS; ...
Lmax = LpSS;

% % Get inputs for N2k Manually
% disp('For Nord2000 calculations...')
% dtdz = input('Temperature gradient, dtdz = ');
% u = input('wind speed component in direction of propagation (+ downwind, - ... upwind) = ');
% zu = input('height of measured wind = ');
% Find the events mentioned in NMT data. Since multiple rows can have same
% Corr ID, make a list of CorrIDs without repetition.
% lookup_table = unique(NMT_num(:,1));
% % Define frequency and frequency spectra
% bno = 17:40;
% f = 10.^((bno/10);
% % ---------------------------------------------------------------
% % Ask for time window input - ***ALSO UNCOMENT "ELEMINATION ROUND 2"***
% z='[12:00 AM'; '06:00 AM'; '12:00 PM'; '06:00 PM';
% % Tst = input(['1 going to 3 AM','
% % '3 AM to 9 AM','
% % '9 AM to 3 PM','
% % '3 PM to 9 PM','After 9 PM']);
% % Tst = input(['1 going to 3 AM','
% % '3 AM to 9 AM','
% % '9 AM to 3 PM','
% % '3 PM to 9 PM','After 9 PM']);

% if Tst<5
% want_ct = [char(date(1)),' ',z(Tst,:));
elseif Tst==5
% want_ct = [datestr(datumen(date(1))+1),' ',z(1,:)); %This is our target ...
% Date-Time string
% end
% Get weather data for target time window.
% idx_cfsr = find(datumen(CFSR_YVR.t) == datumen(want_ct)+1);
% +1 I will access the next 6 hr time window in CFSR data set to account for ...
% UTC-7:00 time zone.
% % To get clear idea refer UTC_PST.mat. Convert to datestr to check the time zone
% Zg_mean = mean(CFSR_YVR.Zg,1); % Mean elevation from sea level [m]
% idx_isoT = find(Zg_mean < 787);
% Tgrad = CFSR_YVR.T(idx_cfsr,idx_isoT)-273.15;
% Zgrad = Zg_mean(1,idx_isoT)./10^3;
% figure();
% plot(Tgrad,Zgrad)
% title('Temperature gradient for ','cur_dts'))
% SIInt=polyfit(Tgrad,Zgrad,1);
% dtdz=SIInt(1);
% idx_isoW = find(Zg_mean < 787);
% Wgrad = CFSR_YVR.U(idx_cfsr,idx_isoW);
% Zgrad = Zg_mean(1,idx_isoW)./10^3;
% u = Wgrad(end);
% zu = Zgrad(end).*10^3;
% if or((u<0 , da==1), (u>0 , da==2))
% u=-u
% end
```matlab
% START THE LOOP!
for n=1:length(idx_active)

current_track = idx_active(n);
corrID = cell2mat(RDConstruct(current_track,z));
rows = find(NMT_num(:,1)==corrID);
active_NMT = NMT_num(rows,2);

% ELIMINATION ROUND 1 ###################################################################
% Check if All events in idx_active are Jet type airplanes. If not, go to next event!
if ~ismember(current_track,cell2mat(NGConstruct(:,1)))
    temptext = sprintf('lw data unavailable for %g',n);
disp(temptext)
continue
end

% ###########################################################################
% check time of the day for gradient info
cur_date = datenum(date(idx_active(n))); % num format. Interested in floor(cur_date)
cur_time = datenum(EvStTime(idx_active(n),:)); % num format. Interested in digits ...
    % after decimal pt.
tzPick = cur_time - floor(cur_time);
if tzPick<0.125 % Upto 3:00 AM
tz=1;
elseif tzPick<0.375 % 3 AM to 9 AM
tz=2;
elseif tzPick<0.625 % 9 AM to 3 PM
tz=3;
elseif tzPick<0.875 % 3 PM to 9 PM
tz=4;
else % After 9 PM, take weather data from next day's 12 AM
tz=1;
cur_date=cur_date+1;
end
z=['12:00 AM';'06:00 AM';'12:00 PM'; '06:00 PM'];
cur_dt = [datestr(cur_date),','+num2str(tz,3)]; % Date-Time string for this event

% MULTIPLE PLOTS WITH XX NMTS (NO time window criteria)
idx_cfar = find(datenum(CFSR_YVR.t) == datenum(cur_dt)) + 1;
% +1 will access the next 6 hr time window in CFSR data set to account for UTC-7:00 ... 
    % time zone.
% To get clear idea refer UTC_PST.mat. Convert to datestr to check the time zone
Zg_mean = mean(CFSR_YVR.Zg,1); % Mean elevation from sea level [m]
idx_isoT = find(Zg_mean<1000);
Tgrad = CFSR_YVR.T(idx_cfar,idx_isoT)-273.15;
Zgrad = Zg_mean(1,idx_isoT).*10^3;
RH = CFSR_YVR.Hs(idx_cfar,1)*10000;
```
temp1

if adj_LpscA
cur_NoiseID
lp_spectrumA
addpath lp10_41
AirAtt

% Define frequency and frequency spectra
bno = 17:40;
f = 10.^((bno/10);
idx_noiseID = find(cell2mat(NGConstruct(:,1))==current_track);
if da ==2
idx_sc = find(cell2mat(LwData(:,1))==cell2mat(NGConstruct(idx_noiseID,5))); 
else
idx_sc = find(cell2mat(LwData(:,1))==cell2mat(NGConstruct(idx_noiseID,6))); 
end
3:end if bno==17:40 | 3:19 if bno=14:33
lp_sc = cell2mat(LwData(idx_sc,3:end));

% Reference spectrum corrected to remove air attenuation Eq. 10-41 (Pg258 AEDT Tech manual)
AirAtt = [3.3 3.3 3.3 6.6 6.6 9.8 13.1 13.1 19.7 23 29.5 36.1 45.9 59 75.4 98.3 ... 131.1... 170.5 229.5 311.5 360.7 524.6 721.3 983.6] / 100;
lp10_41 = lp_sc + AirAtt.*3.05;
addpath('C:\Users\mxb1096\Documents\Research_material\YVR Noise Analysis ... project\RadarDataFiles\Codes');
WA = AWt(f);

% Corrected spectrum adjusted to 1m from S using reference atmosphere. Eq. 10-42 ...
% (Pg258 AEDT Tech manual)
lp_spectrumA = round(10.*(lp10_41 - (20*log10(1/305)) - (AirAtt/100) + WA)) / 10;

% -------------------Adjust for normalization------------------------
cur_NoiseID = NGConstruct(idx_sc,4);
adj_LpscA = DeNormalize(operation, cur_NoiseID, lp_sc);
if adj_LpscA == 0
temptext = sprintf('Skipped %s : NPD data unavailable. ', num2str(corrID));
disp(temptext)
% ipL = lp_spectrumA;
continue
else
temp1 = adj_LpscA + AirAtt.*3.05;
ADJlpA1m = round(10.*(temp1 - (20*log10(1/305)) - (AirAtt/100)) / 10;
% figure(1)
% plot the difference
% semilogx(f,ADJlpAlm,'b');hold on; semilogx(f,lp_spectrumA,'r');
% xlim([min(f) max(f)])
% xlabel('Frequency [Hz']);ylabel('SPL [dBA]')
% title('A weighted SPL at 1 m from aircraft';'after removing air attenuation ...  
% effect of reference atmosphere'))
% legend('Adjusted spectrum',' Without adjustment', 'Location','Best')
% end

% % retrofitted Lw [dBA] for B777 (Spectral class 105) from K. Poulins' code. ...
% OVERRIDE from spectral class if
% required. If this is being used, GOTO SphSpr.m and make necessary
% changes for
% bno = 17:33;
% f = 10.^/(bno/10);
% ipL = [116 121 130 134 140 141 145 147 147 148 148 148 149 150 130];

% % To override all lw calculations, for checking specific input type,
% % UNCOMMENT THIS!-
% % ipL = lw_sc;

% define x number of active NMTs required for considering an event for
% analysis length(active_NMT)=xx
if length(active_NMT)~=xx
  %Doesn't match? Skip this event, its outside of desired time window
  temptext = sprintf('Skipped %s : it has %g ...
  NMTs',num2str(corrID),length(active_NMT));
  disp(temptext)
  skippedForNMT = skippedForNMT+1;
else
  temptext = sprintf('%s COMPLIES all criteria.',num2str(corrID));
  disp(temptext)
  % Duration of events can be found in NMT data [sec]
  tau = NMT_num(rows,?);'
  trigNMT(n,1:xx) = active_NMT; % Record active NMTs
  UsableEvents(n,1) = corrID; % Record corrID
  SEL(n,1:xx) = round(NMT_num(rows,4),1);
  Leq(n,1:xx) = round(10.*(SEL(n,:)-10*log10(tau)))/10;
  Lmax(n,1:xx) = round(NMT_num(rows,5),1);
  %Create 0 values at active NMT indices (This is just to make sure that NMT texts
  %are not written over and over again on the plot)
  dynamicNMT(active_NMT,1)=0;

% !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% ipL=lp_spectrumA;
ipL = ADJlpAlm;
% !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% Original radar track for corr_ID(n)
Y = Y_o+(Y_m./(60+1852));
X = X_o+(X_m./(60+1852.*cosd(Y)));
% disp('Start LpSS calc')
% Calculate LpSS i.e. SPL (dB) due to spherical spreading and air attenuation
% RH=70; T=20;
addpath('C:\Users\mxb1096\Documents\Research_material\YVR Noise Analysis ...
project\RadarDataFiles\Codes');
[LpSS(n,1:xx),horzdist(n,1:xx),Sh(t(n,1:xx)),CP(n,1:xx)] = ...
    SphSpr(active_NMT,NMT,X,Y,Z_m,f,ipL,RH,Tgrad(1));

% MULTIPLE PLOTS WITH XX NMTS (NO time window criteria)
idx_cfsr = find(datenum(CFSR_YVR.t) == datenum(cur_dt)+1);
% +1 will access the next 6 hr time window in CFSR data set to account for UTC-7:00 ...
time zone.
% To get clear idea refer UTC_PST.mat. Convert to datestr to check the time zone.
Zg_mean = mean(CFSR_YVR.Zg(1)); % Mean elevation from sea level [m]
idx_isoT = find(Zg_mean < max(Sht(n,:)));
Tgrad = CFSR_YVR.T(idx_cfsr,idx_isoT)-273.15;
Zgrad = Zg_mean(idx_cfsr,idx_isoT)./10^3;
RH = CFSR_YVR.Hs(idx_cfsr,1).*10000;
S1Int=polyfit(Tgrad,Zgrad,1);
dtdz=S1Int(1)
idx_isoW = find(Zg_mean < max(Sht(n,:)));
Wgrad = CFSR_YVR.U(idx_cfsr,idx_isoW);
Zgrad = Zg_mean(idx_cfsr,idx_isoW)./10^3;
u = Wgrad(end);
zu = Zgrad(end).*10^3
% Change the polarity of wind direction in case its defined in the ...
% direction opposite to direction of propagation.
if or((u<0 , da=-1), (u>0 , da=2))
    u=-u
end

% Calculate LpN2k i.e. Run Compro19ABC Nord 2000 outdoor sound propagation code.
% This code can run on 3 cases depending on the kind of sound speed
% profile. (1) Homogeneous c. (2)Linear c. (3)Log-Lin c.
% addpath('C:\Users\mxb1096\Documents\Research_material\YVR Noise Analysis ...
project\NORD2000\Modified files');
for kase = 1:3
    if kase==1
        LpN2kcl(n,1:xx) = RunCompro(ipL,NMT,kase,active_NMT, ...
         horzdist(n,:),Sh(t(n,:),f,RH,Tgrad(1));
    elseif kase==2
        LpN2kc2(n,1:xx) = RunCompro(ipL,NMT,kase,active_NMT, ...
         horzdist(n,:),Sh(t(n,:),f,RH,Tgrad(1),dtdz);
    elseif kase == 3

80
LpN2kc3(n,1:xx) = RunCompro(ipL,NMT,kase,active_NMT,...
    horzdist (n,:),Sht(n,:),f,RH,Tgrad(1),dtdz,u,zu);

end
end

% disp('done with LpN2k calc')
% Start plotting on map
hold on
if da==1
    plot(X,Y,'Color',[0.2 0.1 0.7], 'MarkerSize',5, 'LineWidth',1.3) %blue
elseif da==2
    plot(X,Y,'Color',[0.2 0.7 0.1], 'MarkerSize',5, 'LineWidth',1.3) %green
elseif da==3
    plot(X,Y,'Color',[0.7 0.2 0.2], 'MarkerSize',5, 'LineWidth',1.3) %red
end
clear X
clear Y
end

plot(X_o,Y_o,'.*r', 'MarkerSize',8)

% For title of plot
if da==1;
    op='Departures from';
elseif da==2;
    op='Arrivals at';
elseif da==3;
    op='Overflights';
    runway='';
end

% Making a fancy movie!
M(count)=getframe;
count=count+1;
end
% disp('Start plotting Google map')
addpath('C:\Users\mb1096\Documents\Research_material\YVR Noise Analysis ...
    project\RadarDataFiles\Codes');
plot_google_map
alpha(0.5)

% We want to avoid overwriting NMT numbers on the plot.
% If an NMT was triggered, there is a 0 at corresponding index in
% dynamicNMT
% Say if 5 events passed the criteria of xx active NMTs. Each event can have a
% different set of xx NMTs that it triggers. So UsedNMTs can be xx or more!
UsedNMTs = find(~dynamicNMT(:,1));
UsableEvents=UsableEvents(UsableEvents>0); % Delete all corr ID = zeros from list.
plot(NMT(UsedNMTs,2),NMT(UsedNMTs,1),'.b', 'MarkerSize',10);
text(NMT(UsedNMTs,2)-0.007,NMT(UsedNMTs,1)-0.007, num2str(UsedNMTs), 'FontSize',12);
% Feel like plotting all NMTS? Uncomment and dont complain!
% plot(NMT(:,2),NMT(:,1),'b', 'MarkerSize',10)
str1=sprintf('%g %s %s on %s that activate %g ... 
NMT(s)', length(UsableEvents), op, runway, char(date(2)), xx);

title(str1, 'fontsize', 12);

% text(X_o-0.075, Y_o, 'Radar reference pt \rightarrow', 'Color', 'red')

zoomHandle = zoom; % Will refresh maps and maintain trasparency effect of alpha ...
      after zoom
set(zoomHandle, 'ActionPostCallback', @update_google_map);
panHandle = pan; % Will refresh maps and maintain trasparency effect of alpha ...
      after pan
set(panHandle, 'ActionPostCallback', @update_google_map);
hold off

txtop = sprintf('Done!');

% This following bit magicstr is creating dynamic variable names to finally
% store list of corrIDs of specific runway, specific operation with
% 'xx' NMTs
if da==1
    magicstr = ['R', runway, 'D', int2str(xx), 'NMT'];
else if da==2
    magicstr = ['R', runway, 'A', int2str(xx), 'NMT'];
else if da==3
    magicstr = ['O', int2str(xx), 'NMT'];
end

% This is a list of corr IDs that activate more than xx NMTs. xx can be
% defined in the if statement above. If UsableEvents is empty, none of the
% events meet set criteria. This is just to help narrow down on events that
% activate say more than 2 or 3 NMTs.
% UsableEvents=UsableEvents(UsableEvents>0); % Delete all corr ID = zeros from list.
assignin('base', magicstr, UsableEvents); % This assigns values in UsableEvents ...
      to a variable name that is made by magicstr

trigNMT = trigNMT(all(trigNMT,2),:);
horzdist = horzdist(all(horzdist,2),:);
Sht = Sht(all(Sht,2),:);
CP = CP(all(CP,2),:);
Lmax = Lmax(all(Lmax,2),:);

% UNCOMMENT all assignins below to save all LpS with magicstr name
% separately
SEL = SEL(all(SEL,2),:);
% assignin('base', ['SEL', magicstr], SEL);

Leq = Leq(all(Leq,2),:);

LpSS = LpSS(all(LpSS,2),:);
% assignin('base', ['LpSS', magicstr], LpSS);

LpN2kcl = LpN2kcl(all(LpN2kcl,2),:);
% magicstr1 = ['LpN2k', int2str(kase), magicstr];
% assignin('base', magicstr1, LpN2k);
LpN2kc2 = LpN2kc2(all(LpN2kc2,2),:);
LpN2kc3 = LpN2kc3(all(LpN2kc3,2),:);

% Fancy movie creations!
% movie(M,2,10);
v=VideoWriter('movie_deps.mp4');
v.FrameRate = 5;
open(v);
writeVideo(v,M);
close(v);

% Construction of complex3 --> 3D matrix
complex3=zeros(xx,13,length(UsableEvents));

for qq = 1:page
    complex3(:,1,qq) = trigNMT(qq,:);
    complex3(:,2,qq) = horzdist(qq,:)/1000;
    complex3(:,3,qq) = Shl(qq,:)/1000;
    complex3(:,4,qq) = CP(qq,:)/1000;
    complex3(:,5,qq) = SEL(qq,:);
    complex3(:,6,qq) = Lmax(qq,:);
    complex3(:,7,qq) = Leq(qq,:);
    complex3(:,8,qq) = LpSS(qq,:);
    complex3(:,9,qq) = LpN2kc1(qq,:);
    complex3(:,10,qq) = LpN2kc2(qq,:);
    complex3(:,11,qq) = LpN2kc3(qq,:);
    complex3(:,12,qq) = complex3(:,9,qq) - complex3(:,6,qq); %err1
    complex3(:,13,qq) = complex3(:,11,qq) - complex3(:,6,qq); %err2
end

% To plot only the common NMTs, we first arrange these NMTs in an order. and
% sort their corresponding SEL, LpSS and LpN2k
for ww=1:page
    complex3(:,:,ww) = sortrows(complex3(:,:,ww));
end
allNMT = unique(complex3(:,1,:));
complex3 = real(complex3);
temptext = sprintf('%s events SKIPPED for unequal nos. of ... NMTs',num2str(skippedForNMT));
disp(temptext)
assignin('base','allNMT',allNMT,page);disp('allNMT = unique(complex3(:,1,:));
complex3 = real(complex3);
temptext = sprintf('%s events SKIPPED for unequal nos. of ... NMTs',num2str(skippedForNMT));
disp(temptext)
assignin('base','allNMT',allNMT,page);disp('allNMT = unique(complex3(:,1,:));
complex3 = real(complex3);
temptext = sprintf('%s events SKIPPED for unequal nos. of ... NMTs',num2str(skippedForNMT));
disp(temptext)
assignin('base','allNMT',allNMT,page);disp('allNMT = unique(complex3(:,1,:));
complex3 = real(complex3);
temptext = sprintf('%s events SKIPPED for unequal nos. of ... NMTs',num2str(skippedForNMT));
disp(temptext)
assignin('base','allNMT',allNMT,page);disp('allNMT = unique(complex3(:,1,:));
complex3 = real(complex3);
temptext = sprintf('%s events SKIPPED for unequal nos. of ... NMTs',num2str(skippedForNMT));
disp(temptext)
assignin('base','allNMT',allNMT,page);disp('allNMT = unique(complex3(:,1,:));
complex3 = real(complex3);
temptext = sprintf('%s events SKIPPED for unequal nos. of ... NMTs',num2str(skippedForNMT));
disp(temptext)
assignin('base','allNMT',allNMT,page);disp('allNMT = unique(complex3(:,1,:));
complex3 = real(complex3);
temptext = sprintf('%s events SKIPPED for unequal nos. of ... NMTs',num2str(skippedForNMT));
disp(temptext)
assignin('base','allNMT',allNMT,page);disp('allNMT = unique(complex3(:,1,:));
complex3 = real(complex3);
temptext = sprintf('%s events SKIPPED for unequal nos. of ... NMTs',num2str(skippedForNMT));
disp(temptext)
assignin('base','allNMT',allNMT,page);disp('allNMT = unique(complex3(:,1,:));
complex3 = real(complex3);
temptext = sprintf('%s events SKIPPED for unequal nos. of ... NMTs',num2str(skippedForNMT));
disp(temptext)
assignin('base','allNMT',allNMT,page);disp('allNMT = unique(complex3(:,1,:));
complex3 = real(complex3);
temptext = sprintf('%s events SKIPPED for unequal nos. of ... NMTs',num2str(skippedForNMT));
disp(temptext)
assignin('base','allNMT',allNMT,page);disp('allNMT = unique(complex3(:,1,:));
complex3 = real(complex3);
temptext = sprintf('%s events SKIPPED for unequal nos. of ... NMTs',num2str(skippedForNMT));
disp(temptext)
assignin('base','allNMT',allNMT,page);disp('allNMT = unique(complex3(:,1,:));
complex3 = real(complex3);
temptext = sprintf('%s events SKIPPED for unequal nos. of ... NMTs',num2str(skippedForNMT));
disp(temptext)
assignin('base','allNMT',allNMT,page);
elseif $ww==2$
    errortab = outerjoin(tab1,tab2,'MergeKeys',true);
else
    errortab = outerjoin(errortab,tab,'MergeKeys',true);
end

% uncomment inner join & comment outer join to get a errortab with ONLY
% the NMTs that are common to all qualifying events.
% elseif $ww==2$
%    errortab = innerjoin(tab1,tab2);
% else
%    errortab = innerjoin(errortab,tab);
% end
end

% Replace NaNs by 0s
vars = errortab.Properties.VariableNames;
et = errortab(:,vars);
et(isnan(et))=0;
errortab(:,vars)=et;

% Find average error in err1
errortab.Err1_avg = round(10*(mean(errortab(:,2:2:page*2),2))/10); %All odd columns ...
from err1
% Find average error in err2
errortab.Err2_avg = round(10*(mean(errortab(:,3:2:page*2),2))/10); %All even columns ...
from err2

figure()
scatter(errortab(:,1),errortab(:,end-1),'filled','s','MarkerEdgeColor','k'); %err1 ... blue square
hold on
scatter(errortab(:,1),errortab(:,end),'filled','MarkerEdgeColor','k'); %err2 red circle
set(gca,'xtick',sort(unique(errortab(:,1))));
% [2 4 6 14 15]
xlim([min(allNMT)-1 max(allNMT)+1]);
grid on
% legend('Error1','Error2','Location','best')
title('Mean Error')
xlabel('NMT number'
ylabel('\Delta L_p'

% Other plots such as a comparison of measured and calculated L_p
% Now we break X-axis to get fancy!
% Decide up to where to break axis?? Break at 6 for sure.
% Start again 1-NMT# that appears first while ascending from 7 through 13
startX = 5+ find(ismember([7,8,9,10,11,12,13,14,15],active_NMT),1);
breakxaxis([startX]);
% set(gca,'Layer','Top')
legend('Red','Blue','Location','West');

% Innerjoin errortab
clear errortabIJ

for ww=1:page
    dynamicStr = ['tab',int2str(ww)];
    tab = table(complex3(:,1,ww),complex3(:,12,ww),complex3(:,13,ww),...
                'VariableNames',{'NMT',['err1',int2str(ww)],[err2',int2str(ww)]});
    assignin('base',dynamicStr,tab);
    if ww==1
        continue
    elseif ww==2
       errortabIJ = innerjoin(tab1,tab2);
    else
       errortabIJ = innerjoin(errortabIJ,tab);
    end
end

% Replace NaNs by 0s
vars = errortabIJ.Properties.VariableNames;
et = errortabIJ(:,vars);
et(isnan(et))=0;
errortabIJ(:,vars)=et;

% Find average error in err1
errortabIJ.Err1_avg = round(10*(mean(errortabIJ(:,2:2:page*2),2))/10); %All odd ...
columns from err1
% Find average error in err2
errortabIJ.Err2_avg = round(10*(mean(errortabIJ(:,3:2:page*2),2))/10); %All even ...
columns from err2

figure()
scatter(errortabIJ(:,1),errortabIJ(:,end-1),'filled','s','MarkerEdgeColor','k'); ... %err1 blue square
hold on
scatter(errortabIJ(:,1),errortabIJ(:,end),'filled','MarkerEdgeColor','k'); %err2 ... red circle
set(gca,'xtick',sort(unique(errortabIJ(:,1))));

% Plot output spectrum
index = 1:xx;
indexC = get(gca,'ColorOrder');
for ii=1:4
    hold all
plot(index, SEL(ii,:), 'Color', indexC(ii,:))

plot(index, LpSS(1,:), 'Color', indexC(ii,:))

end

legend('SEL', 'SphSpr', 'N2k')

xlabel('Active NMTs')

ylabel('Lp')


% Plot interest points

figure(1)

hold all

plot(x_intrstPts, y_intrstPts, '.', 'MarkerSize', 14)

plot(x_Ipt_m, y_Ipt_m, '.', 'MarkerSize', 14)

plot(X_o, Y_o, '*', 'MarkerSize', 8)

plot_google_map

alpha(.7)

if da==1;
    op='Departures from';
else
    op='Arrivals at';
end

str = sprintf('%%g %s runway %s', length(Y_md), op, runway);
title(str, 'fontsize', 14)
text(X_o-0.075, Y_o, 'Radar reference pt \rightarrow', 'Color', 'red')

% All NMT locations

hold on;
plot(NMT(:,2), NMT(:,1), '.', 'MarkerSize', 10)
plot_google_map
alpha(0.5)
title('Noise Monitor Terminal (NMT) locations', 'fontsize', 14);
for ii=1:length(NMT);
    text(NMT(ii,2), NMT(ii,1), num2str(ii))
end
hold off

%% Comparison of inputs for B777 Class DEP105

%% Define frequency and frequency spectra

bno = 17:40;
f = 10.^(bno/10);

addpath('C:Users\mxb1096\Documents\Research_material\YVR Noise Analysis ...
project\RadarDataFiles');
[temp, temp, LwData] = xlsread('SpectralClassData.xlsx', 4);

AirAtt = [3.3 3.3 3.3 6.6 6.6 9.8 13.1 13.1 19.7 23 29.5 36.1 45.9 59 75.4 98.3 ...
131.1... 170.5 229.5 311.5 360.7 524.6 721.3 983.6]/100; % Ait att SAE 1845 [db/100m]
This function calculates the aircraft spectral data adjusting it to the maximum thrust setting as described in section 4.4.

C.2 Supporting functions

The main code calls many supporting functions.

C.2.1 DeNormalize.m

This function calculates the aircraft spectral data adjusting it to the maximum thrust setting as described in section 4.4.

```matlab
function adj_LpscA = DeNormalize(operation, cur_NoiseID, lp_sc)
    addpath('C:\Users\mb1096\Documents\Research_material\YVR Noise Analysis ...
         project\RadarDataFiles\Codes');
    WA = AWT(f);
    % Corrected spectrum adjusted to 1m from S using reference atmosphere. Eq. 10-42 ...
    (Pg258 AEDT Tech manual)
    lp_spectruman = round(10.*lp_sc - (20*log10(1/305)) - (AirAtt/100) + WA))/10;
    lp_spectruman = round(10.*(lp_sc - (20*log10(1/305)) - (AirAtt/100))/10;
    lw_sc = lp_spectruman * 20*log10(1)+11;
    co = get(gca,'ColorOrder');
    semilogx(f,lp_sc,'LineWidth',1.3)
    hold on
    semilogx(f,lp_spectruman,'LineWidth',1.3)
    semilogx(f,lp_spectruman,’-‘,'LineWidth',1.3)
    semilogx(f,lw_sc,’-‘,'LineWidth',1.3)
    semilogx(f,lw_sc,’-‘,'LineWidth',1.3)
    xlim([50 10000])
    legend(’L_w : B777-300 NPD data from Kieran Code [dBA]’,’L_p : DEP-105 data at 305 ...
    m from S. [dB]’, ...
    ’L_p : DEP-105 data corrected to 1 m [dB]’,’L_A : DEP-105 data corrected to 1 m ...
    [dB]’, ’Lw’, ‘Location’, ’best’) hold off
    xlabel(’1/3rd Octave band frequency [Hz]’);
    ylabel(’Sound Level [dB]’)
```

87
if isempty(idx_unqNPD)==1;
adj_LpscA = 0;
else
cur_NPD = cell2mat(NPDcurve(idx_unqNPD,5:end));
end
bno = 17:40;
f = 10.^((bno/10);
WA = AWt(f);
% De-normalization of spectrum
lp_scA = lp_sc + WA;
lpOA = overall(lp_scA);
\Delta = cur_NPD(4)-lpOA;
adj_LpscA = lp_scA + \Delta;
end

C.2.2 SphSpr.m

The function SphSpr.m is the in-house developed propagation model as described in point number 6 of section 5.1.3.

function [Lp,horzdist,Sht,ClosestPt] = SphSpr(active_NMT,NMT,X,Y,Z_m,f,ipL,RH,T) % ...
  uses real atm
% function [Lp,horzdist,Sht,ClosestPt] = SphSpr(active_NMT,NMT,X,Y,Z_m,f,ipL) % ...
  uses standard atm
% How many NMTs are active for this track?
dim = length(active_NMT);

% Initialize these matrices to null.
% Each matrix has # of columns = # of active NMTs
% and # of rows = # of points in the track.
Y_dist = zeros(length(Y),dim); X_dist = Y_dist; Slant_dist = Y_dist;
% ClosestPt will store minimum Slant distance from each active NMT to some
% point in the flight trajectory. This point won't always be the same.
% 'I' will store the location or row number of ClosestPt
ClosestPt = zeros(1,dim); I = zeros(1,dim); Lp = zeros(1,dim);
horzdist = zeros(1,dim); Sht = zeros(1,dim);

for nmtNo= 1:dim
% Find the distance (m) of all points along the track from active NMT#1 to end.
Y_dist(:,nmtNo) = (NMT(active_NMT(nmtNo),1)-Y).*60*1852;
X_dist(:,nmtNo) = (NMT(active_NMT(nmtNo),2)-X).*60*1852.*cosd(Y);
Slant_dist(:,nmtNo) = sqrt(power(Y_dist(:,nmtNo),2) +
                              power(X_dist(:,nmtNo),2)+power(Z_m,2));
[ClosestPt(nmtNo),I(nmtNo)] = min(Slant_dist(:,nmtNo));
25 horzdist(1,nmtNo) = sqrt(power(Y_dist(I(nmtNo),nmtNo),2) + ...
26 power(X_dist(I(nmtNo),nmtNo),2));
27 Sh(1,nmtNo) = Z_m(I(nmtNo));
28 end
29
30 [cp,in] = min(ClosestPt);
31 WA = AWt(f);
32
33 for nmtNo = 1:dim
34 % If input is a Sound POWER level spectrum ...(Check if Lw is A
35 % weighted--> Include/Exclude WA??)
36 % dLa = airatt(ClosestPt(nmtNo),20,70,f); %Original use. Acc to ISO 9613-1. Has ...
37 % all -ve values
38 % lp_spectr = round10.*(ipL - (20*log10(ClosestPt(nmtNo))) - 11 + dLa))/10; ...
39 %Original SS + Air abs +A wt
40 % If input is a Sound PRESSURE level spectrum at 1 m from Source...
41 % (Check if Lp is A weighted-->Include/Exclude WA??)
42 dLa = airatt(ClosestPt(nmtNo),T,RH,f); %Original use. Acc to ISO 9613-1. Has ...
43 % all -ve values
44 lp_spectr = ipL - 20*log10(ClosestPt(nmtNo)) + dLa ;
45 % Use this for LpAedt calculations
46 % dLa = AirAtt.*10^((ClosestPt(nmtNo)); %Experimental use. Acc to SAE-AIR-1845. ...
47 % Has all +ve values
48 % lp_spectr = round10.*(lw - (20*log10(ClosestPt(nmtNo)/305)) - dLa))/10; ...
49 %Experimental SS + Air abs
50 sum = 0;
51 for ii=1:length(lp_spectr)
52 randm = 10^((lp_spectr(ii)/10);
53 sum = sum + randm ;
54 end
55 Lp(nmtNo) = round10*log10(sum));
56 end
57
58 horzdist = horzdist.';
59 Sh = Sh.';
60 ClosestPt=ClosestPt.';
61 Lp = Lp.';

C.2.3 RunCompro.m

The function RunCompro.m identifies one of the three cases (homogeneous/linear or log-linear) for calculation based on the number of input parameters to this function. It is described in point number 7 of section 5.1.3.

1 % This function is to run NORD 2000 by modifying the examples to suit YVR
2 % track analysis.
% Author - Manasi Biwalkar
% Spring 2017

function LpN2k = RunCompro(lw,NMT,kase,active_NMT,dirdist,Sht,f,RH,t0,varargin) ...
% uses real atm
function LpN2k = RunCompro(lw,NMT,kase,active_NMT,dirdist,Sht,f,varargin) %uses ...
standard atm
addpath('C:\Users\mxb1096\Documents\Research_material\YVR Noise Analysis ...
project\NORD2000\Modified files');
% Inputs - horzdist : straight line distance in x-y plane
% found in Back_Calcs.m
% Sht : Source ht. also from Back_Calcs.m
% lw : Sound Power level of source
% kase : Case No. 1-Homogenous; 2-Linear; 3-LogLin
% THIS PART MUST BE CHECKED FOR INPUTS EACH TIME!!!
% monitor = input('active NMT index = ');
xz=2; % number of points in terrain profile 2 for FLAT TERRAIN
% xz matrix
% columns --> | x | z | flow resistivity | ground roughness |
% rows --> no of points to be specified
xz(1,1)=0; % x1
xz(1,2)=0; % z1
xz(1,3)=200000; % flow resistivity, grass
xz(1,4)=0; % ground roughness (a value of zero is recommended until further notice)
% xz(2,1)=horzdist(moniter); % x2
xz(2,2)=0; % z2
xz(2,3)=0; % not used
xz(2,4)=0; % not used
% compro calculation parameters
% hsv = Sht(monitor); % vertical source height
% hrv = 1; % vertical receiver height
z0 = 0.1; % roughness length (wind profile)
su = 0; % standard deviation of fluctuating wind speed component (0 for ...
instantaneous noise levels)
t0 = 20; % temperature at the ground (degrees Celsius)
sdtdz=0; % standard deviation of fluctuating temperature gradient (normally =0)
Cv2 = 0; % turbulence strength, wind, typical value
Ct2 = 0; % turbulence strength, temperature, typical value
% RH = 70; % relative humidity in % (0 means ignore air absorption)
nacat=0; % scattering zone definition (not used)
xzscat=0; % scattering zone definition (not used)
%...
```matlab
% lw = input('SoundPowerLevel = '); % For single Lw values
% kase = input([{'Case 1: Homogeneous sound speed','\nCase 2: Linear Profile'...'
\nCase 3:Log-Lin profile','\nEnter Case No: '}]);
% if kase == 2
dtdz = input('dtdz = ');
% elseif kase == 3
u = input('wind speed component in direction of propagation (+ ... downwind, - upwind) = ');
% zu = input('height of measured wind = ');
% end
% UNCOMMENT IF FILE IS A FUNCTION
if kase == 1
% DEFAULT VAL. INCASE of Case 2 or 3, varargin function will rewrite this
% later in the function. DO NOT COMMENT OUT !!
dtdz = -0.01; % temperature gradient for neutral weather
zu = 10; % height of measured wind
u = 1; % wind speed component in direction of propagation (+ downwind, - upwind)
else
% varargin = length(varargin);
dtz = varargin{1};
zu = 10; % height of measured wind
u = 1; % wind speed component in direction of propagation (+ downwind, - upwind)
if kase == 3
% if nargin > 1
u = varargin{2};
zu = varargin{3};
end
end
% =========================================================================
% determine vertical sound speed profile parameters
% A--> Log profile coeff
% B--> Linear profile coeff
% C--> sound speed at t0 deg Cel
% sA,sB-->std. deviations in A, B, C
Lp = zeros(length(active_NMT),1);
for monitor = 1:length(active_NMT)
xz(2,1)=dirdist(monitor); % x2 or Distance of R along horizontal (from S)
hsv = Sht(monitor); % vertical source height
hrv = NMT(active_NMT,monitor,3);
% % Define frequencies and find A wt
% bno = 17:33;
% f = 10.^((bno/10);
% WA = AM(f);
C = coft(t0);
A = u/log(zu/z0+1);
sA=su/log(zu/z0+1);
```
B = dtdz * 201.0013/C; % lydhastighedsgradienten ved t0 = dtdz * 10.025/sqrt(t+273.15)
sB = sdtdz * 201.0013/C;

if kase == 1
    A = 0; B = 0;
elseif kase == 2
    A = 0;
end

% define mean temperature in deg C
tmean = t0 + dtdz * (hsv + hrv)/2;
tair = tmean;

dL = compro19ABC(nxz, xz, hsv, hrv, z0, A, sA, B, sB, t0, Cv2, Ct2, tair, RH, nscat, xzscat, f);

% calculation of sound pressure level
R = sqrt((xz(nxz,1) - xz(1,1))^2 + (xz(nxz,2) + hrv - xz(1,2) - hsv)^2); % ...
    source-receiver distance

% LdB = lw - 11 - 20*log10(R) + dL ;
LdB = lw - 20*log10(R) + dL ;

% LdB = lw - 20*log10(R/305) + dL ; %To be used with lp10_41 AEDT method as ...
input lw.
    input lw.

% LdBA = LdB + WA ;
LdBA = LdB;

sum = 0;
for ii = 1:length(LdBA)
    randm = 10^((LdBA(ii))/10);
    sum = sum + randm;
end

Lp (monitor) = round(10*log10(sum)); % original
            Lp = round(10*log10(sum));
end
LpN2k = Lp;
end
Bibliography


URL https://www.mathworks.com/matlabcentral/fileexchange/27627-zoharby-plot-google-map


URL http://www.movable-type.co.uk/scripts/latlong.html

94