

# **Emitter Source Geolocation from Imparted Rotor Blade Modulation**

by

Thomas Schucker

A Thesis Submitted to the Faculty of the

ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfilment of the Requirements

For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

October 2016

## **STATEMENT BY AUTHOR**

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that an accurate acknowledgement of the source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Thomas Schucker

## **APPROVAL BY THESIS DIRECTOR**

This thesis has been approved on the date shown below:

---

Tamal Bose	Date
Professor of	.
Electrical and Computer Engineering	.

## *Acknowledgements*

I would like to thank my thesis advisor Dr. Tamal Bose for the opportunity to work in his lab, and for the guidance through graduate school. I would also like to thank Matt Kruse and Charlie Cooper for coming up the research topic and support through the process. I would also like to thank the Broadband Wireless Access & Applications Center and Rincon Research Corporation for supporting this research.

*To my Mother and Father,  
my brother Danny,  
and my lovely girlfriend Clarice,  
all with love,  
Thank you for your unconditional support.*

# Contents

<b>Statement By Author</b>	<b>1</b>
<b>Acknowledgements</b>	<b>2</b>
<b>List of Figures</b>	<b>6</b>
<b>List of Tables</b>	<b>8</b>
<b>Abstract</b>	<b>9</b>
<b>1 Introduction</b>	<b>10</b>
1.1 Motivation . . . . .	10
1.2 Contribution . . . . .	10
1.3 Organization . . . . .	11
<b>2 Literature Review and Geometric Models</b>	<b>12</b>
2.1 Introduction . . . . .	12
2.2 Literature Review . . . . .	12
2.3 Geometric Models . . . . .	13
2.3.1 2D Geometric Model . . . . .	14
2.3.2 3D Geometric Model . . . . .	15
2.3.2.1 Transmitter inline with Receiver and Rotor Blade . . . . .	16
2.3.2.2 Approaching Model . . . . .	19
2.3.2.3 Receding Model . . . . .	20
<b>3 Radio Propagation Simulation</b>	<b>22</b>
3.1 Introduction . . . . .	22
3.2 Radio Propagation Model . . . . .	22
3.2.1 Free-Space Path Loss . . . . .	23
3.2.2 Reflection . . . . .	23
3.2.2.1 Specular Reflection . . . . .	23
3.2.2.2 Reflection Loss . . . . .	23

3.2.2.3	Diffuse Reflection . . . . .	24
3.2.3	Fading . . . . .	24
3.2.4	Doppler Effect . . . . .	25
3.3	Scene Model . . . . .	26
3.3.1	Rotor Blade . . . . .	26
3.3.2	Transmitter . . . . .	28
3.3.3	Receiver . . . . .	28
3.4	Ray Tracing . . . . .	29
3.4.1	Computation . . . . .	29
3.4.2	Output . . . . .	31
<b>4</b>	<b>Received Signal Simulations</b>	<b>32</b>
4.1	Introduction . . . . .	32
4.2	Analysis Techniques . . . . .	32
4.3	Receiver Position . . . . .	34
4.4	Rotor Pitch . . . . .	37
4.5	Transmitter Azimuth Angle . . . . .	39
4.6	Transmitter Elevation Angle . . . . .	43
<b>5</b>	<b>Results</b>	<b>47</b>
5.1	Introduction . . . . .	47
5.2	Transmitter Azimuth Angle Estimation . . . . .	47
5.3	Transmitter Elevation Angle Estimation . . . . .	53
<b>6</b>	<b>Conclusion and Future Work</b>	<b>61</b>
6.1	Summary . . . . .	61
6.2	Future Work . . . . .	61
<b>A</b>	<b>Ray Tracing Software</b>	<b>63</b>
<b>B</b>	<b>Class Documentation</b>	<b>66</b>
B.1	Blade_surface Class Reference . . . . .	67
B.2	Bounding_box Class Reference . . . . .	69
B.3	Point3D Class Reference . . . . .	71
B.4	Ray3D Class Reference . . . . .	73
B.5	Receiver Class Reference . . . . .	75
B.6	Rib Class Reference . . . . .	77
B.7	Rotor Class Reference . . . . .	78
B.8	Scene Class Reference . . . . .	80
B.9	Sphere Class Reference . . . . .	82
B.10	Transmitter Class Reference . . . . .	84
B.11	Triangle Class Reference . . . . .	86
B.12	Vector3D Class Reference . . . . .	88

# List of Figures

2.1	2D Geometric model with Rx at the Axis of Rotation . . . . .	14
2.2	Doppler Frequency vs. Rotor Position with varying Elevation Angle . . . . .	15
2.3	3D Geometric model with Tx at Azimuth angle of 90° . . . . .	16
2.4	Doppler profile of Tx at Azimuth angle of 90° . . . . .	17
2.5	3D Geometric model with Tx at Azimuth angle of 270° . . . . .	18
2.6	Doppler profile of Tx at Azimuth angle of 270° . . . . .	18
2.7	3D Geometric model with Tx at Azimuth angle of 180° . . . . .	19
2.8	Doppler profile of Tx at Azimuth angle of 180° . . . . .	20
2.9	3D Geometric model with Tx at Azimuth angle of 0° . . . . .	21
2.10	Doppler profile of Tx at Azimuth angle of 0° . . . . .	21
3.1	Airfoil Cross-Section . . . . .	27
3.2	Blade Surface Tessellation . . . . .	27
3.3	Transmitter Rays . . . . .	29
3.4	Disk Sampling Method . . . . .	31
4.1	Example Spectrum for one Blade revolution with transmitter at Azimuth of 45°and Elevation of 35° . . . . .	33
4.2	Example Spectrum with Doppler Profile for one Blade revolution with transmitter at Azimuth of 45°and Elevation of 35° . . . . .	34
4.3	Position of the Receiver in Relation to the Rotor Blades . . . . .	35
4.4	Max and Min Doppler vs. Receiver Position with Tx Azimuth at 135°and Elevation at 54° . . . . .	35
4.5	Doppler Envelope with receiver at 0m . . . . .	36
4.6	Doppler Envelope with receiver at 7m . . . . .	36
4.7	Rotor Pitch . . . . .	37
4.8	Max and Min Envelope Frequencies vs Rotor Pitch with Tx Azimuth at 135°and Elevation at 54° . . . . .	38
4.9	Max and Min Envelope Frequencies vs Rotor Pitch Transmitter Underneath the Helicopter . . . . .	38
4.10	Max Envelope Frequencies vs Transmitter Elevation Angle . . . . .	39
4.11	Frequency Difference vs Transmitter Elevation Angle . . . . .	40
4.12	Transmitter Azimuth Angle Relationship . . . . .	40
4.13	Max and Min Envelope Frequencies vs Transmitter Azimuth Angle with Tx Elevation at 45° . . . . .	41

4.14	Max and Min Envelope Frequencies vs Transmitter Azimuth Angle with Tx Elevation at 76° . . . . .	42
4.15	Max and Min Envelope Frequencies vs Transmitter Azimuth Angle with Tx Elevation at 26.5° . . . . .	42
4.16	Max and Min Envelope Frequencies vs Transmitter Azimuth Angle with Rx at Axis of Rotation . . . . .	43
4.17	Max and Min Envelope Frequencies vs Transmitter Elevation Angle with 7.5deg Pitch . . . . .	44
4.18	Transmitter Elevation Angle . . . . .	44
4.19	Transmitter Range vs Elevation Angle at an Altitude of 200m . . . . .	45
4.20	Max and Min Envelope Frequencies vs Transmitter Elevation Angle with Tx Azimuth of 135deg . . . . .	46
4.21	figures . . . . .	46
5.1	Max and Absolute value of Min Envelope Frequencies vs Transmitter Azimuth Angle at an Elevation Angle of 45° . . . . .	48
5.2	Difference between Max and Min Envelope Calculations vs Transmitter Azimuth Angle at an Elevation Angle of 45° . . . . .	49
5.3	Correct Doppler Frequencies vs Transmitter Azimuth Angle at an Elevation Angle of 45° . . . . .	49
5.4	Correct Doppler Frequencies with Minimum vs Transmitter Azimuth Angle at an Elevation Angle of 45° . . . . .	50
5.5	Peaks of inverted Correct Doppler vs Transmitter Azimuth Angle at an Elevation Angle of 45° . . . . .	51
5.6	Average Power vs Transmitter Azimuth Angle at an Elevation Angle of 45°	51
5.7	Estimated Reflection Radius at an Elevation Angle of 45° . . . . .	54
5.8	Actual Elevation Angle and Estimated Elevation Angle vs Azimuth Angle	55
5.9	Elevation Angle Percent Error vs Transmitter Azimuth Angle . . . . .	55
5.10	Max and Min Envelope Frequencies vs Elevation Angle with Transmitter Azimuth Angle of 180° . . . . .	57
5.11	Elevation Estimate and Actual Angle with Transmitter Azimuth Angle of 180° . . . . .	57
5.12	Elevation Angle Percent Error with Transmitter Azimuth Angle of 180°	58
5.13	Uncorrected Envelope Difference vs. Transmitter Azimuth and Elevation angle of 45° with Blade pitch of 7.5° . . . . .	59
5.14	Uncorrected Doppler vs. Transmitter Azimuth and Elevation angle of 45° with Blade pitch of 7.5° . . . . .	59

# List of Tables

5.1	Azimuth Error and Azimuth Percent Error of Different Elevation Angles During Azimuth Estimation . . . . .	52
5.2	Percent Peak, Power, and Frequency Difference of Different Elevation Angles During Azimuth Estimation . . . . .	52
5.3	Average Elevation Error and Average Elevation Percent Error of Different Elevation Angles During Azimuth Estimation . . . . .	56
A.1	Simulation parameters . . . . .	64
A.2	Output file header with simulation parameter order . . . . .	65

## *Abstract*

In RF communications with a rotorcraft, such as a helicopter, the rotor blades can impart a modulation onto the received signal called Rotor Blade Modulation (RBM). This modulation is caused by the reflection of a signal off the rotating blades. The reflected signal is Doppler shifted based on where the signal is reflected along the length of the blade as well as the angle between the axis of rotation and the emitter. RBM is known to degrade the performance of RF communications on rotorcraft and can be used in RADAR applications to detect and classify aircraft, but there is little on its usefulness in other areas. This thesis looks at the ability to utilize the RBM phenomenon on the rotorcraft itself to geo-locate and track a signal emitter on the ground.

To do this a 3D RF ray tracing program was developed in C++ to produce simulations of RBM signals. The developed program is based on optical ray tracing algorithms with modified physical propagation effects for RF signals, and swapping lights and cameras for RF transmitters and receivers respectively. The ray tracer was then run over a realistic set of physical parameters to determine their effects on the received signal, this includes transmitter azimuth and elevation angle, receiver position, blade pitch, etc. along with their combinations. From the simulations the azimuth and elevation angle, of the transmitter, produce predictable modulations on the received signal. Based on the trends in the signal's modulation a DSP algorithm was distilled down that accurately determines the azimuth and elevation angle of the transmitter from simulated signal data.

# **Chapter 1**

## **Introduction**

### **1.1 Motivation**

In RF communications with a rotorcraft, such as a helicopter, the rotor blades can impart a modulation onto the received signal called Rotor Blade Modulation (RBM). This modulation is caused by the reflection of a signal off the rotating blades. The reflected signal is Doppler shifted based on where the signal is reflected along the length of the blade as well as the elevation angle between the axis of rotation and the emitter. RBM can be used in RADAR to detect and classify aircraft based on a specific signature and can degrade communication system located on the rotorcraft itself. The motivation behind this work is to use the RBM effect to locate uncooperative emitters on the ground. This is done through the use of specific knowledge of the craft itself and measuring the RBM in the received signal.

### **1.2 Contribution**

The main contributions of this thesis are as follows. A 3D Ray Tracing software is provided to simulate the RF propagation in the presence of a helicopter rotor blade. The software provides insight into the complex interaction between the rotor blade shape and the placement of the receiver and transmitters. The second contribution is an in

depth analysis of how several different variables affect the modulation on the received signal. And lastly, techniques to use the imparted rotor blade modulation to estimate the azimuth and elevation angles with respect to the location of the rotating blades.

### 1.3 Organization

In Chapter 2 we present the background and theory behind rotor blade modulation. we then formulate the basic geometry behind the RBM effect.

In Chapter 3 we Introduce the radio propagation simulation in several sections. First is the RF propagation model which defines the physical effects that are simulated within the software. Next the objects of the scene model, as well as the dimensionality, are defined in software to be representative of their physical dimensions and functionality. Lastly we describe the core ray tracing engine that traces the scene in successive frames to build the receiver output.

In Chapter 4 we present the result of the parameter swept simulations of the ray tracing software. Next the received signals from the simulation are analyzed using several DSP techniques to define the signal characteristics that will be used in the next chapter for estimation.

In Chapter 5 Those characteristics are then used to estimate the azimuth and elevation angle of the transmitter based on the given rotor and receiver parameters. Those estimates are then compared with the original position of the transmitter to determine the methods accuracy.

In Chapter 6 the summery of the results are detailed along with the current limitations. Lastly future improvements on the work will be discussed.

## Chapter 2

# Literature Review and Geometric Models

### 2.1 Introduction

This section details the basic phenomenon behind rotor blade modulation along with its significance in other applications, and the defining characteristics that allow its use in this application. The second part of this section describes the geometric models for the 2 dimensional case and several 3 dimensional cases. These models will form the basis for verification of the simulations.

### 2.2 Literature Review

The rotating blades of the helicopter modulate signals that hit the blades by reflecting them. The reflected signal, through the doppler effect, become shifted in frequency in relationship to the velocity of the reflection point (Yaman Zhang 2000). The rotor blade modulation (RBM) has a complex form and is influenced by several object parameters. These parameters range from the positioning of a signal source to the physical dimensions of the blades.

The resulting modulated signal has been used in RADAR applications. First to determine the specific characteristics of a noncooperative helicopter using time frequency analysis in (Barry D. Bullard 1991) that defines a specific signature for each parameter. The ability to detect a hovering helicopter based on the blade flashes caused by reflected RADAR pulses off the blades is defined in (J Misiurewicz 1997). The micro-doppler produced by the RADAR reflection is used in (Thayaparan, T. 2007) to determine the rotation rate of the main and tail rotor blades using the wavelet transform. Other types of aircraft produce modulations on a returned RADAR pulse, one type known as Jet Engine Modulation (JEM) is the product of a RADAR pulse reflecting off the jet engine blades (RADAR HANDBOOK). The information can similarly be used to quantify physical aspects of the aircraft when present in order to perform classification.

The doppler shifted signal caused by RBM effects the performance of common adaptive GPS anti-jamming algorithms located on a helicopter platform, unless the covariance estimation is completed within a few milliseconds. And in iterative based approaches the non-stationary signal characteristics cause degradation as well (F. Barbiero 2009). The affect of RBM on the BER of communications is examined in (Yaman Zhang 2000). Where effects of multi path and doppler cause large bit error rates at small elevation angles and large scattering coefficients described in the model.

The role of RBM in this thesis is based on specific doppler shifts associated with where a dominant reflection occurs along the length of the blade, and the associated elevation angle of the transmitter. The received signal is then be used to create a mapping between the amount of doppler present and the position of the transmitter on the ground.

### 2.3 Geometric Models

The geometric model will begin with a two dimensional scene that will describe the geometry between the rotor blade, the transmitter and the receiver. Then the model will evolve into a selection of three dimensional models that will form the foundation for the results of the ray tracer.

### 2.3.1 2D Geometric Model

The two dimensional geometric mode is based on a few assumptions, one is that the receiver is placed as close as possible to the center of rotation, and second that all the physical objects are located within a single 2D plane. This model is represented in figure 2.1.

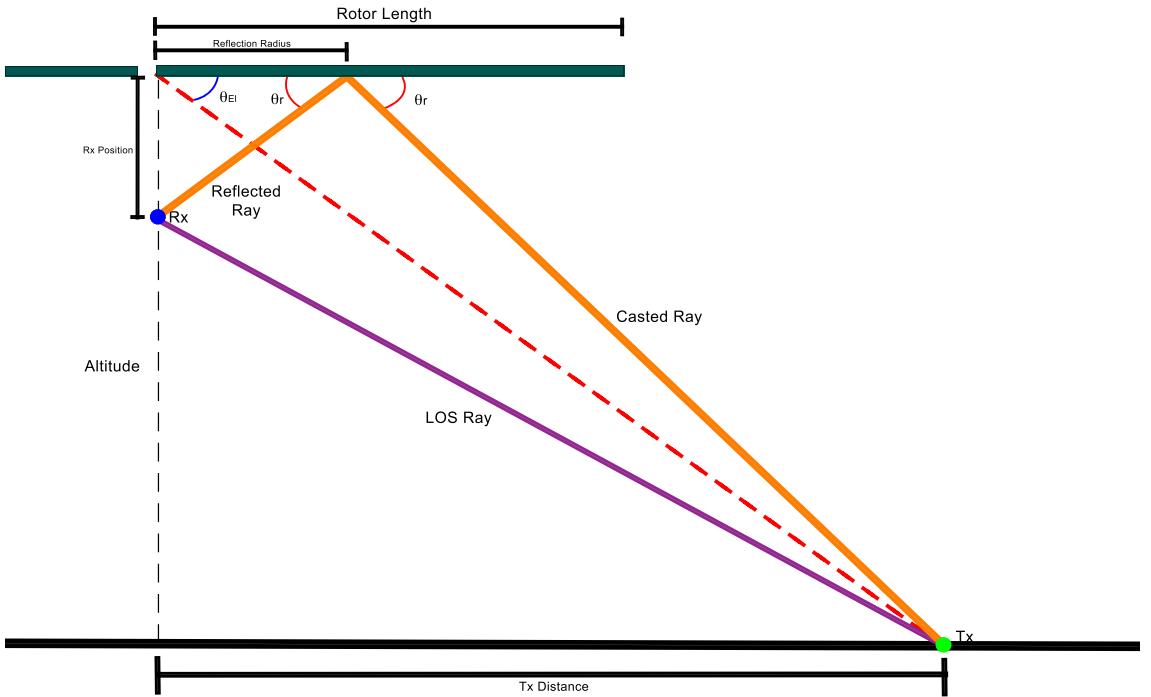


FIGURE 2.1: 2D Geometric model with Rx at the Axis of Rotation

There are three physical objects within this scene. The first is the Rotor blade which is the dark green bar depicted in the figure, next are the transmitter in green  $T_x$  and the receiver  $R_x$  in blue. The measurements depicted as black bars with denoting start and stop points reference the various distances used in the following equations. The dashed green line shows at which altitude the helicopter is hovering at, and the red dashed line transverses the transmitter and the center of rotation for the blades.  $\theta_{El}$  is the elevation angle of the transmitter with respect to the rotor blades and  $\theta_r$  denotes the specular reflection of the casted and reflected rays in orange. Lastly the line of sight (LOS) ray shows that the transmitter and receiver are in visual line of sight of each other.

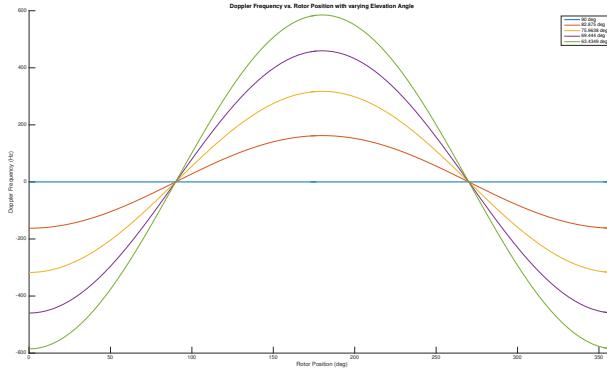


FIGURE 2.2: Doppler Frequency vs. Rotor Position with varying Elevation Angle

(equations for doppler profile)

Figure 2.2 shows the varying doppler frequency caused by the RBM at various elevation angles of the transmitter. The results of this is a sinusoidal doppler profile in which its amplitude increases as the elevation angle decreases.

Since this model has the receiver located at the axis of rotation this model can be seen as a special case of the next 3D models, this is because it is symmetric around the axis of rotation. Therefore any placement of the transmitter in 3D space can be distilled down to a 2D profile as such. This symmetry should then be mirrored in the ray tracing simulation as verification.

### 2.3.2 3D Geometric Model

The three dimensional geometric model is categorized by the position of the rotor blade and the position of the signal source. Since the rotor blade has a specific airfoil shape the models will take that into account through an approximation of that surface. The first model will be representative of having the transmitter inline with both the receiver and the blade, with the transmitter in both fore and aft positions. The next model will have the rotor approaching both the receiver and transmitter, and lastly the rotor will be receding from both the receiver and transmitter. All of the 3D models will have the receiver positioned offset from the axis of rotation.

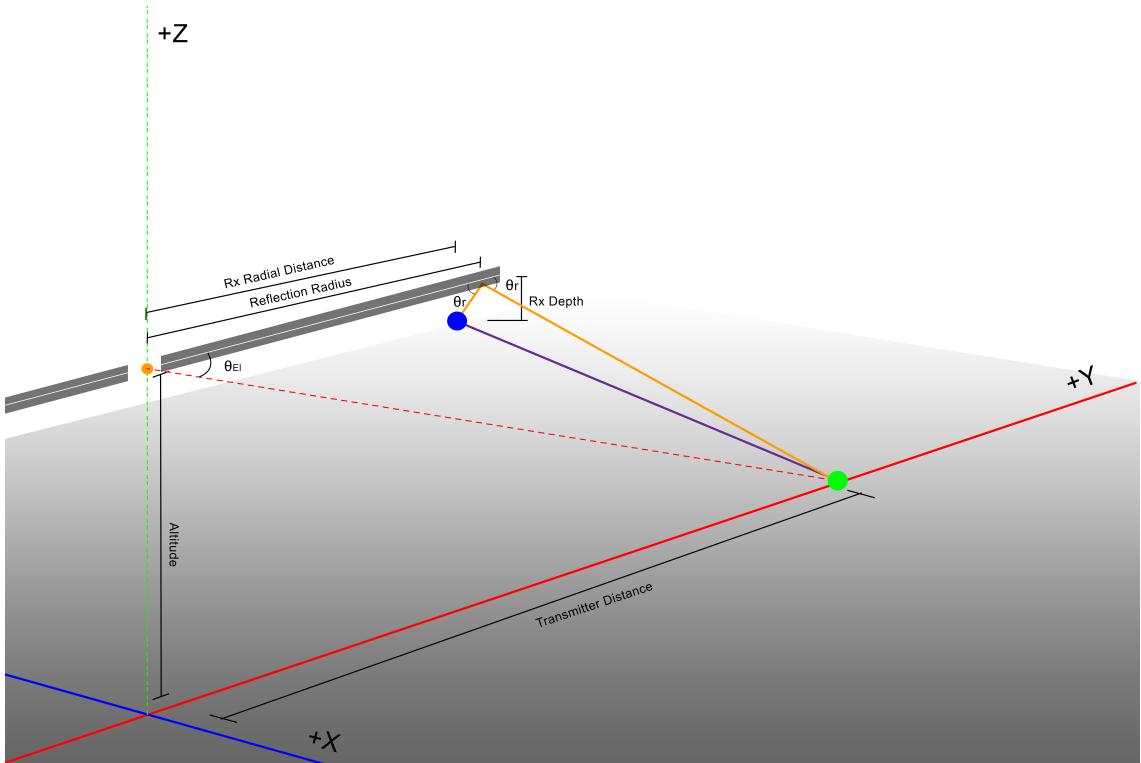


FIGURE 2.3: 3D Geometric model with Tx at Azimuth angle of 90°

### 2.3.2.1 Transmitter inline with Receiver and Rotor Blade

The first two models as mentioned above fall under the criteria of the transmitter being inline with both the rotor blade and the offset receiver.

Figure 2.3 depicts the transmitter at a 90° azimuth angle, which places it inline with the position of the receiver and the current location of the rotating blades.

The objects in 2.3 are similar to the ones previously stated in the 2D case. The X and Y axis are specifically drawn in blue and red respectfully, with the Z axis now the green dashed line and altitude is now denoted with a measurement bar. The receiver offset in the +Y direction is now stated. and the Blades are now colored gray. All other variables and colors denote their objects as previously stated in the 2D case.

(equations doppler profile 90 deg)

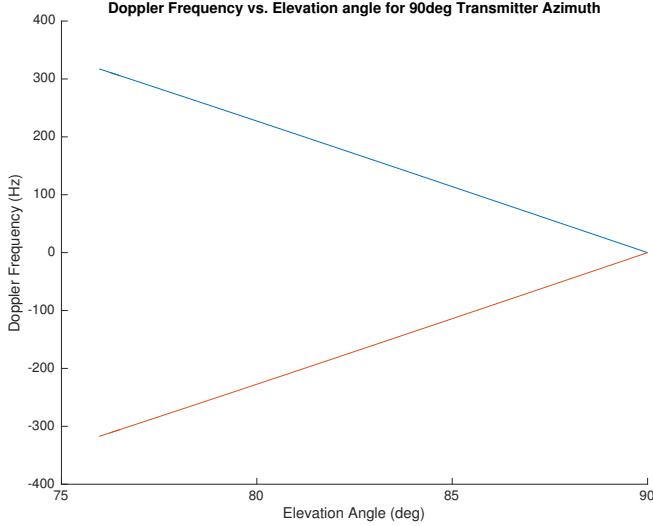


FIGURE 2.4: Doppler profile of Tx at Azimuth angle of 90°

Figure 2.4 shows the maximum and minimum doppler values versus elevation angle for a transmitter at an azimuth angle of 90°. The elevation angle is limited by the reflection angle, caused by the limited distance between the tip of the blade and the location of the receiver. This limitation constrains the doppler profile causing inaccurate transmitter elevation estimation at this transmitter azimuth position.

Figure 2.5 depicts the transmitter at a 270° azimuth angle, which places it inline with the position of the receiver and the current location of the rotating blades. This placement of the transmitter behind the helicopter is different from the situation in figure 2.3 because the value of  $\theta_r$  is now not limited by the tip of the rotor and can go to smaller elevation angles.

(equations doppler profile 270 deg)

Figure 2.6 shows the maximum and minimum doppler values versus elevation angle for a transmitter at an azimuth angle of 270°. The data from the figure is similar to the data from the previous figure 2.3 but does not exhibit the limitation on the reflection angle, allowing for smaller elevation angles and larger doppler shifts to occur.

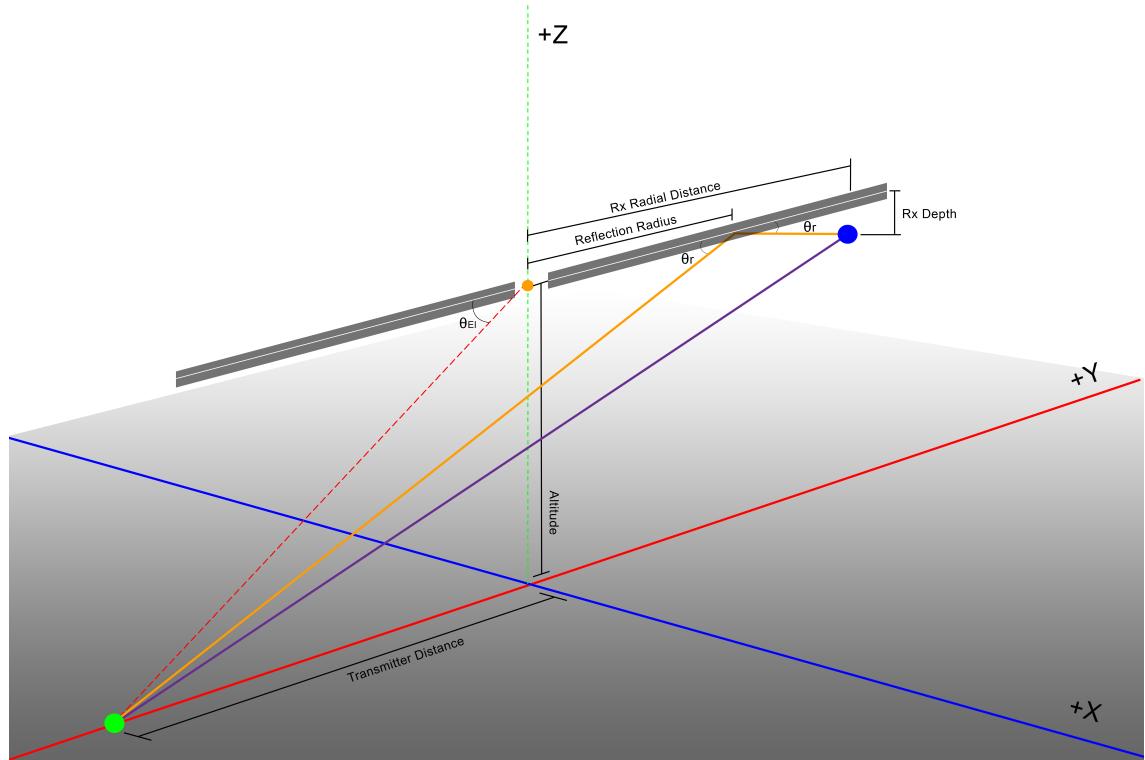


FIGURE 2.5: 3D Geometric model with Tx at Azimuth angle of 270°

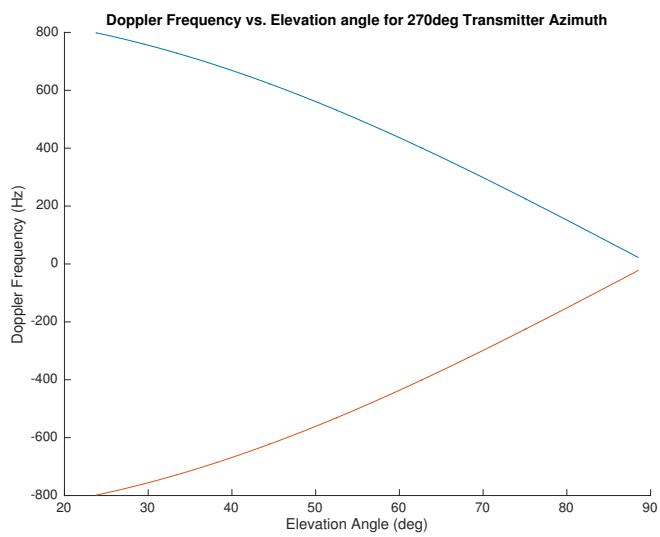


FIGURE 2.6: Doppler profile of Tx at Azimuth angle of 270°

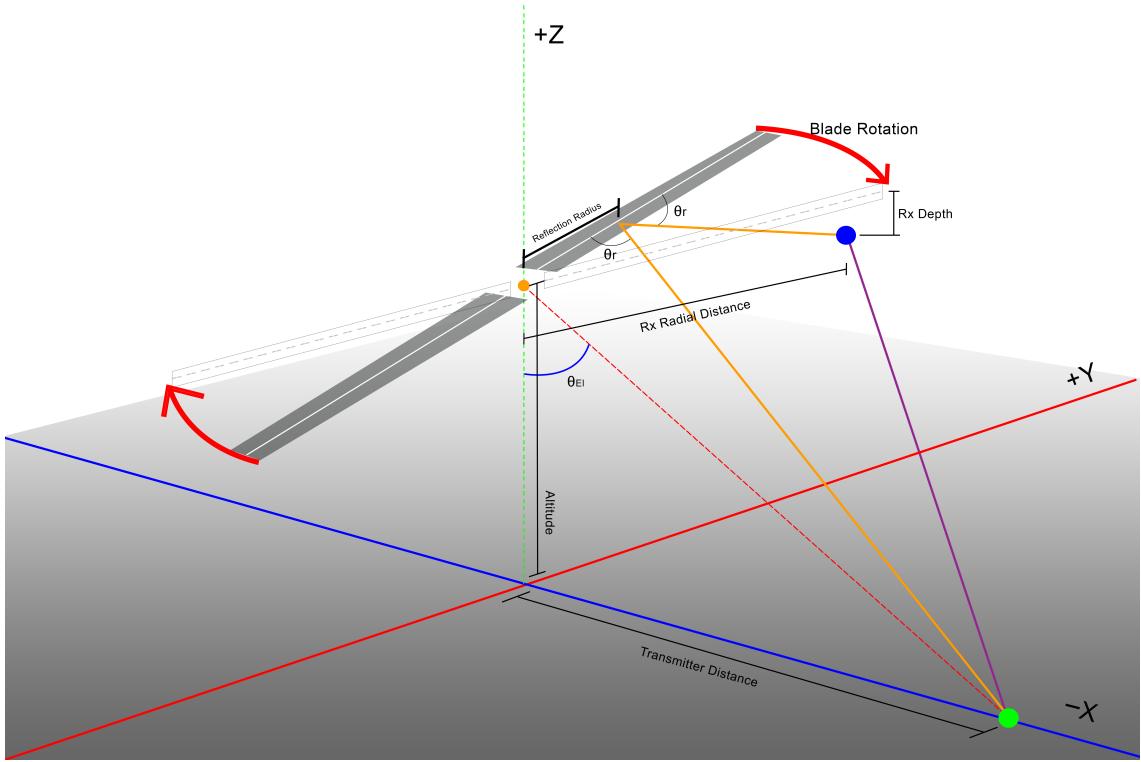


FIGURE 2.7: 3D Geometric model with Tx at Azimuth angle of 180°

### 2.3.2.2 Approaching Model

The approaching model shown in figure 2.7 depicts the transmitter at an azimuth of 180 ° and the axis of X has now been rotated to be -X. The direction of the rotor blades is depicted ,in green, to follow the previous assumptions.

This case of the 3D model places the transmitter at 180 ° so that the oncoming blades move toward both the receiver and the transmitter at the same time. This particular placement will cause a positive doppler profile because the blades will be moving in the direction of both objects, compressing the reflected wave accordingly.

(equations doppler profile 180 deg)

Figure 2.8 shows the maximum and minimum doppler values versus elevation angle for a transmitter at an azimuth angle of 180°. The data from the figure shows the maximum of the doppler profile decrease as the elevation angle gets larger.

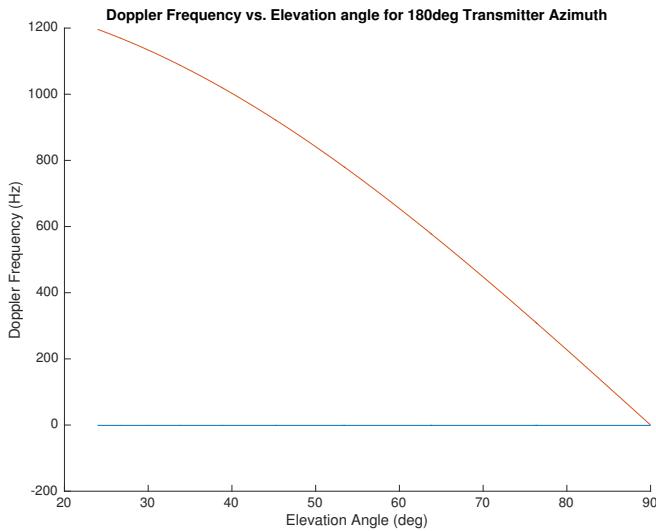


FIGURE 2.8: Doppler profile of Tx at Azimuth angle of 180°

### 2.3.2.3 Receding Model

The receding model shown in figure 2.9 depicts the transmitter at an azimuth of 0 ° where the axis positions back to their original places and the direction of the rotor blade is now depicted in red.

This case of the 3D model places the transmitter at 0 ° so that the receding blades move away from both the receiver and the transmitter at the same time. This particular placement should provide a negative doppler profile because the blades will be moving in the opposite direction of both objects, stretching the reflected wave accordingly.

(equations doppler profile 0 deg)

Figure 2.10 shows the maximum and minimum doppler values versus elevation angle for a transmitter at an azimuth angle of 0°. The data from the figure shows the minimum of the doppler profile increase as the elevation angle gets larger.

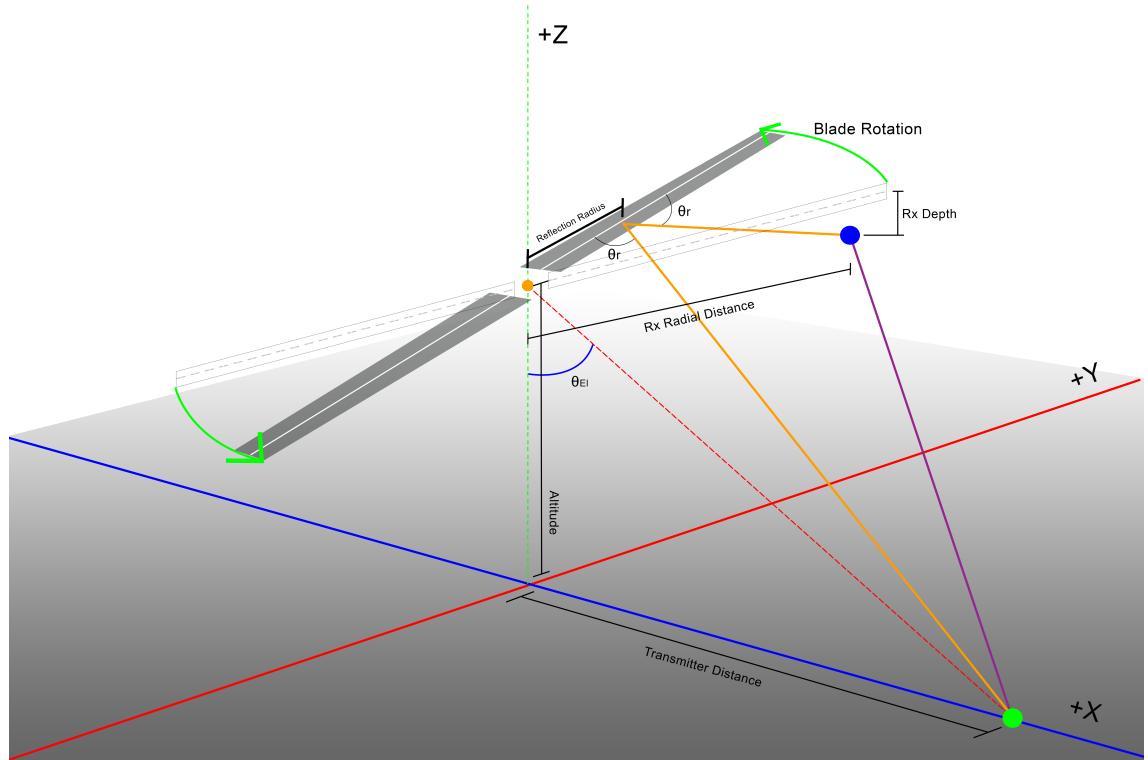


FIGURE 2.9: 3D Geometric model with Tx at Azimuth angle of 0°

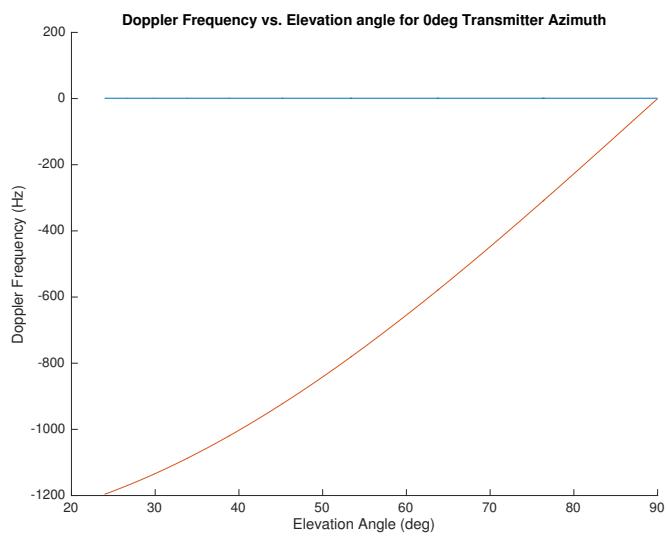


FIGURE 2.10: Doppler profile of Tx at Azimuth angle of 0°

## Chapter 3

# Radio Propagation Simulation

### 3.1 Introduction

The simulation software is broken down into three main components, the radio wave propagation model, the scene model, and the ray tracing engine. The wave propagation model is a collection of equations that describe the physical phenomena that affect the propagation when represented by rays. The model used is based upon classical geometric optics to work with the ray tracing engine. The scene model is comprised of a three dimensional environment in order to define the physical rotor blade surface. The receiver and transmitter will also be defined in the same environment with simulated characteristics to match real life conditions. The ray tracing engine uses recursive execution described in (Whitted, 1979), replacing cameras and lights for receivers and transmitters respectively in a three dimensional environment.

### 3.2 Radio Propagation Model

The propagation model defined in this project is based on the following physical phenomena, free-space path loss, reflection, and the doppler effect. They are all formally established models described in full in (Bertoni, 2000; Parsons, 2000; Willis, 1991)(OK). Although several effects are described within the model several were left out to simplify

the computation due to the overhead of implementation in three dimensional space and because of the assumptions made about the scene model.

### 3.2.1 Free-Space Path Loss

Free-Space path loss is calculated using the Friis transmission and is incorporated into the model by keeping track of the distance a ray travels and adjusting its power is

$$P_r = G_r G_t P_t \left( \frac{\lambda}{4\pi R} \right)^2 \quad (3.1)$$

where  $P_r$  is the power at the receiver,  $P_t$  is the power at the transmitter,  $G_r$  and  $G_t$  are the antenna gains,  $\lambda$  is the wavelength, and  $R$  is the distance between the receiver and transmitter antennas. In the simulations described in this paper,  $G_r$  and  $G_t$  are assumed to have a value of 1.

## 3.2.2 Reflection

### 3.2.2.1 Specular Reflection

In the case of a perfect reflection the angle of the reflection is calculated by

$$\theta_r = \cos^{-1}(\vec{n} \bullet (-\vec{d}_i)) \quad (3.2)$$

where  $\theta_r$  is the angle of the reflected ray,  $\vec{n}$  is the normal vector to the surface, and  $\vec{d}_i$  is the direction vector of the incident ray.

### 3.2.2.2 Reflection Loss

Power loss due to surface reflection is described through the following coefficient

$$\Gamma = \frac{\cos(\theta_i) - \sqrt{\varepsilon} \cos(\theta_t)}{\cos(\theta_i) + \sqrt{\varepsilon} \cos(\theta_t)} \quad (3.3)$$

where  $\theta_i$  is the angle of incidence with the object's surface normal,  $\varepsilon$  is the dielectric constant for that surface, and  $\theta_t$  is the angle of refraction as described by Snell's law [3.4](#).

$$\theta_t = \sin^{-1} \left( \frac{\sin(\theta_i)}{\sqrt{\varepsilon}} \right) \quad (3.4)$$

The final reflected power is

$$P_{reflected} = P_i |\Gamma| \quad (3.5)$$

where  $P_i$  is the incident power before reflection losses.

### 3.2.2.3 Diffuse Reflection

A perfect diffuse reflection, also known as Lambertian reflection, will reflect rays in all directions in a uniform hemisphere (J. Lambert, "Photometria Sive de Mensura et Gradibus Luminis Colorum et Umbrae", 1760). The hemisphere is located at the point of the ray hit, producing rays away from the surface. To approximate a particular type of surface the rays power is attenuated by the surface's bi-directional reflectance distribution function (BRDF) (Suffern, 2007; Pharr, 2010; Glassner, 1995)(OK). There are several ways of calculating the BRDF, but the simplest is the Phong BRDF which will be used in this ray tracing implementation(Phong, 1998)(OK). This method for the BRDF is regularly used in the rendering of computer graphics to simulate the reflection of light off surfaces. Since it was not designed for longer wavelengths used in RF communications an assumption is made that small scale features of the surface, relevant to light waves, produce a reflectance distribution similar for larger wavelengths.

### 3.2.3 Fading

The transmitted signal power experiences fading described by

$$F = \cos\left(2\pi\frac{d}{c} + t_0\right) \quad (3.6)$$

where  $F$  is the fading coefficient,  $d$  is the total distance the particular ray has traveled,  $c$  is the speed of light, and  $t_0$  is the starting time of the current frame. The power can then be calculated using

$$P_{faded} = P F \quad (3.7)$$

where  $P_{faded}$  is the adjusted power due to fading and  $P$  is the signal power. The superposition of each ray's faded power becomes the received signal.

### 3.2.4 Doppler Effect

The Doppler effect is characterized by (FIX Update to Bistatic Doppler)

$$f = \left(\frac{c + v_r}{c + v_s}\right) f_0 \quad (3.8)$$

It is assumed that both the receiver and the transmitter are not moving, therefore the only doppler imparted on the signal will be from the rotation of the blades. The associated Doppler is calculated using

$$f = f_0 + \frac{2\Delta v}{\lambda} \quad (3.9)$$

where  $f_0$  is the initial transmitted frequency,  $\lambda$  is the wavelength, and  $\Delta v$  is the difference in velocity.

$$\lambda = \frac{c}{f_0} \quad (3.10)$$

where  $\lambda$  is the wavelength of the transmitted signal based off of the speed of light  $c$  and its initial frequency  $f_0$

$$\Delta v = r\omega_r(\vec{d}_p \bullet \vec{d}_r) \quad (3.11)$$

where  $\Delta v$  is the difference in velocity between the point of reflection along the length of the rotor blade in the direction of the reflection characterized by equation 3.11. Where  $r$  is the radius along the length of the blade where the reflection occurs.  $\omega_r$  is the angular velocity of the rotor.  $\vec{d}_p$  is the normalized vector that is perpendicular to the rotor in its direction of travel and  $\vec{d}_r$  is the normalized vector that is in the direction of the resulting reflected ray.

### 3.3 Scene Model

The scene model is the 3D environment in which the physical objects are placed. The model is comprised of three types of objects, the rotor blades, transmitter and receiver. The rotor blade is the most complex object due to its airfoil shape. The transmitter is a point source that projects the rays into the scene, and the receiver is described in a similar fashion with a surrounding boundary layer. There are no other objects within the scene, this is because the effect of the rotor on the received signal is to be analyzed for defining characteristics in the best case scenario.

#### 3.3.1 Rotor Blade

The rotor blade is the most complex object within the scene due to its airfoil shape. The airfoil shape is designed to produce lift for the helicopter, and designing one is outside the scope of this project. Fortunately there are databases that provide real helicopter airfoil designs in x,y coordinates. These designs form a 2D cross-section of the airfoil shape that will be extruded into the length of the rotor blade. Figure 3.1 shows the airfoil cross-section that will be used to create the rotor surface.

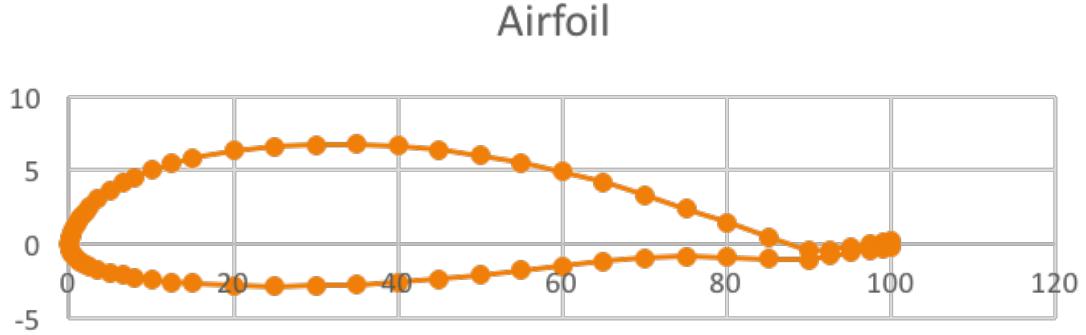


FIGURE 3.1: Airfoil Cross-Section

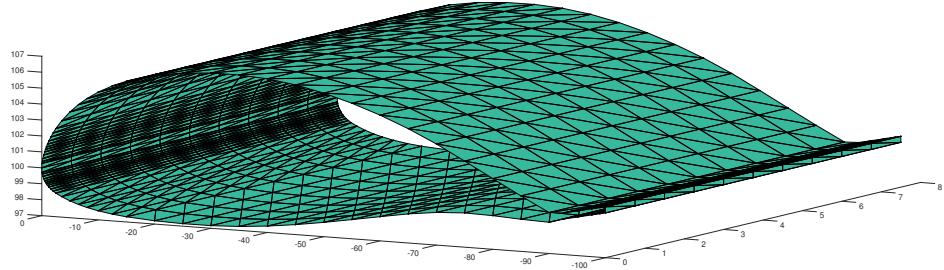


FIGURE 3.2: Blade Surface Tessellation

The 2D airfoil sections will form the ribs of the rotor blade but the surface of the blade will be made up of primitive surfaces to approximate the curved airfoil shape. The primitive shapes are triangles that span between the ribs and go around to cover the surface. This was done using a lightweight mesh generation method using the a priori knowledge of the blade shape. The designed algorithm, located in [B.1 \*create\\_surface\(\)\*](#) method, takes advantage of the blade being composed of rib sections with consistent shape. The surface normals are then adjusted to match the correct directionality. Figure [3.2](#) shows the created surface for the rotor blade that will be used in the scene.

All aspects of the rotor blade are parameterized. Starting with the airfoil shape, which is defined by a set of 2D coordinates. The length and the number of rib sections can be configured along with the number of blades and corresponding altitude. As well as the RPM. This allows the scene to replicate any type of rotor configuration and position it

at a specific altitude.

### 3.3.2 Transmitter

The transmitter is modeled as a point source, this means that all the radio waves are projected from this point into the scene. It is assumed that the transmitter is located on the ground within the x,y plane in relation ship to the rotor blades. The transmitter is configured with a center frequency and is assumed to produce a tone at that frequency. Therefor signal produced has no inherent modulation applied to it. Being that the Transmitter is emitting radio waves into a 3D space it is assumed to be transmitting in an omnidirectional fashion, but because the rotor and the receiver are the only two other objects within the scene radio waves will only be propagated in their direction. This limitation allows for a higher concentration of rays to be casted, resulting in a higher resolution picture and eliminates computational overhead for the tracing engine. Figure 3.3 shows the rays, in blue, casted only in the vicinity of the rotor blade and receiver.

### 3.3.3 Receiver

The receiver object is modeled as a point in 3D space with a surrounding boundary layer in the shape of a sphere. This is because the receiver is assumed to have an omnidirectional antenna which will receive radio waves from all directions uniformly. The receiver will record any ray that hits the boundary layer within each frame of the simulation to create a sample of data at that instance in time. The sample is produced using superposition of all the rays that are acting on the receiver at that instance in time which are then extrapolated to produce a clear result. The receiver is configurable allowing the user to set a desired center frequency along with the sampling rate of the simulated hardware. The sampling rate will determine several simulation parameters in the ray tracer due to the need to associate this rate with the change in the rotor position for the next frame. The consequence of this is that increasing the sample rate will produce higher resolution results but will increase the computational overhead of the simulation.

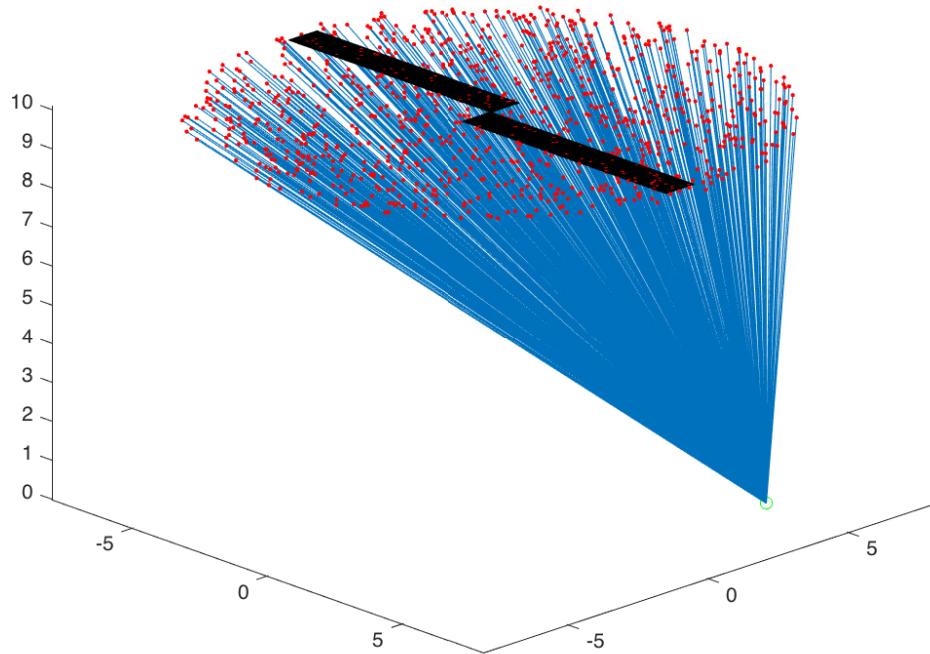


FIGURE 3.3: Transmitter Rays

## 3.4 Ray Tracing

The ray tracing simulation is conducted in the Monte Carlo fashion acting on the scene and propagation models. This method of raytracing is established for creating realistic images in (Suffern, 2007; Pharr, 2010). The simulation is performed over successive iterations. Each iteration is a discrete element called a frame. Where each frame will be updated to represent the change in the position of the rotor blade.

### 3.4.1 Computation

The ray tracing engine first sets up the environment creating the surfaces and placing the objects in the scene. Then the number of frames that will be computed. The number of frames is determined by the number of rotations that the rotor makes within the

simulation, the speed of the rotor in RPM and the sampling frequency of the receiver defined before the beginning of the simulation. This is described by

$$Frames = \frac{60f_s}{RPM} Rev \quad (3.12)$$

where *Frames* is the number of frames that will be computed in the simulation,  $f_s$  is the sampling frequency of the receiver, *RPM* is the revolutions per minuet of the rotor, and *Rev* is the number of revolutions that the rotor makes in the simulation.

In each frame a set number of rays are casted into the scene in a uniformly distributed circle in the direction of the rotor and receiver using

$$v_{ray} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{pmatrix} \sqrt{r} \cos \theta_u + d_x \\ \sqrt{r} \sin \theta_u + d_y \\ d_z \end{pmatrix} \quad (3.13)$$

The casting technique reduces the computation overhead by limiting the search space. It also creates higher resolution result by focusing the limited number of rays in the direction of the other objects shown in figure 3.4.

When a ray is casted into the scene it has an origin point, a direction vector, a frequency and an initial power level. The ray then travels through the scene till it encounters another object. If there is a collision with the receiver it is recorded. If the ray collides with a rotor blade a new ray is calculated based on the normal to the geometric surface. The new ray's power is updated as well as an updated frequency since the blade is moving. The power level is adjusted based on the BRDF of the specific material. Due to only the rotor and receiver in the scene, besides the transmitter, rays that are found to hit the rotor the reflection ray is calculated. Then the angle between the reflection and a ray between the point of reflection and the receiver is used to create a dedicated ray in that direction with a decayed power and adjusted frequency accordingly.

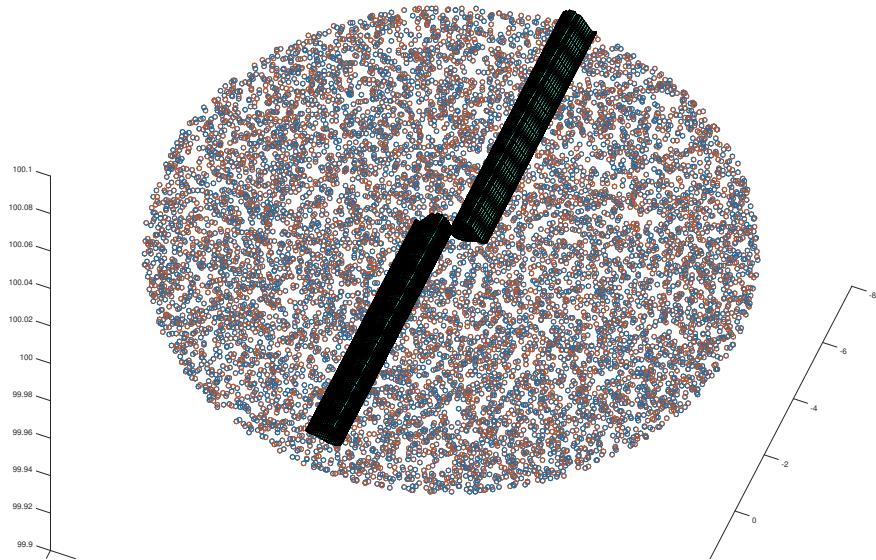


FIGURE 3.4: Disk Sampling Method

### 3.4.2 Output

The superposition of these receiver contributions during each frame creates the received signal data. The received signal data will then be visualized in spectrograms over the length of the simulated time. The simulations do not render in real time but creates a CSV file that can then be loaded into MATLAB for signal processing.

## Chapter 4

# Received Signal Simulations

### 4.1 Introduction

The following simulations performed will define the search space on the effect of rotor blade modulation. Since the software developed allows for full parameterization of the objects within the scene the simulations will focus on a few that effect the different object's positions in 3D space and the resulting received signal. The parameters that will be changed are the position of the receiver, pitch of the rotor blade, and transmitter position. The effect of each parameter will be analyzed to determine what information can be derived for localization.

### 4.2 Analysis Techniques

The data from the .CSV files generated by the raytracing program is loaded into MATLAB scripts to process the data. The technique that is used is the spectrogram function that computes the short time Fourier transform. This allows for computations to be done in the time frequency domain and provide an accurate picture of how the signal is changing with time. The following figure shows an example of the computed spectrograms.

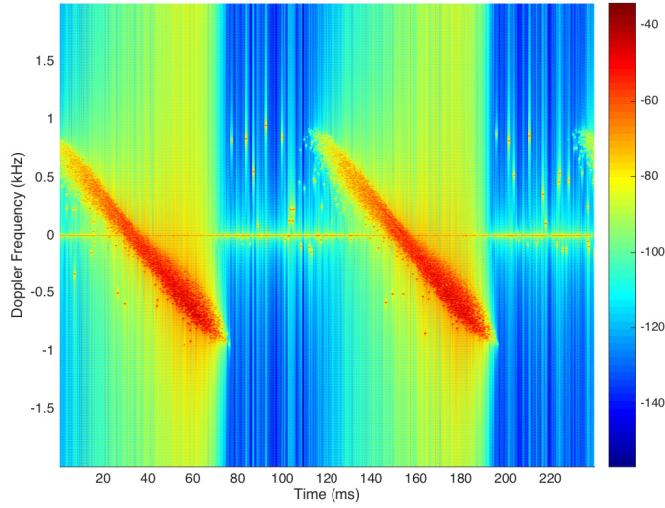


FIGURE 4.1: Example Spectrum for one Blade revolution with transmitter at Azimuth of  $45^\circ$  and Elevation of  $35^\circ$

After the computation of the short time Fourier transform the doppler envelope is calculated by performing a threshold on the power, in both the positive and negative doppler, at each time slice. The envelope value for that time slice is found by approaching zero frequency and stopping as the power reaches a predetermined level from both the negative and positive values. The power level to set the threshold is calculated differently based on whether the elevation or the azimuth parameter is being swept. For the azimuth angle sweep, which will be shown later in section 4.5, the maximum power is found in each time slice then those values are averaged. Then for each successive angle the same average is taken and averaged with all the others

This technique is used because the elevation angle is remaining constant, so the distance with the helicopter platform should be constant as well. When varying the elevation angle in section 4.6 the same method cannot be used because the distance is changing so only the maximum power is found in each time slice then those values are averaged without any successive averaging.

The Doppler envelopes are displayed in figure 4.2, the lower in blue and the upper in red. The minimum is calculated from the lower envelope, and the maximum from the upper

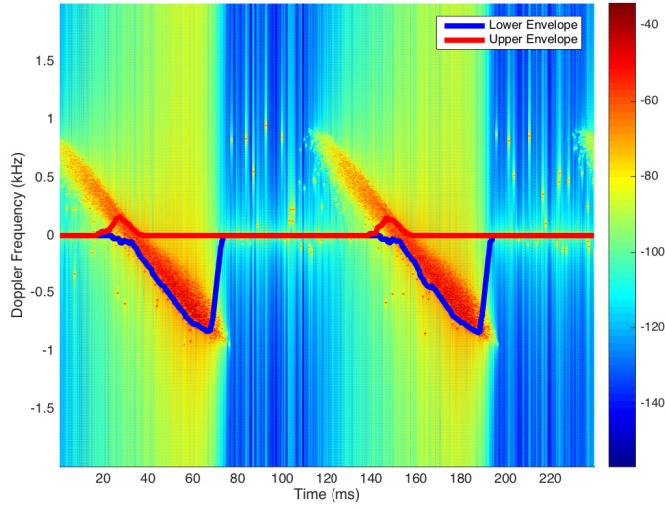


FIGURE 4.2: Example Spectrum with Doppler Profile for one Blade revolution with transmitter at Azimuth of  $45^\circ$  and Elevation of  $35^\circ$

envelope. Those values are then used to characterize that doppler profile and analyze signal trends as different variables change.

### 4.3 Receiver Position

The position of the receiver is determined ultimately by its location on the physical aircraft. The position can be anywhere along the length of the body of the helicopter, but for the purpose of the simulation the receiver position will be varied from directly underneath the rotor to just past the tip.

The rotor blade length from figure 4.3 is 7.5m so the receiver will be varied from 0m to 9m in 1m increments. The resulting signal is analyzed by calculating its doppler envelope. Then the maximum and minimum of that envelope will then be used to determine the overall effect of the position of the receiver.

The figure 4.4 shows maximum doppler vs the position of the receiver. From this graph the maximum doppler frequency changes significantly as the receiver is moved from

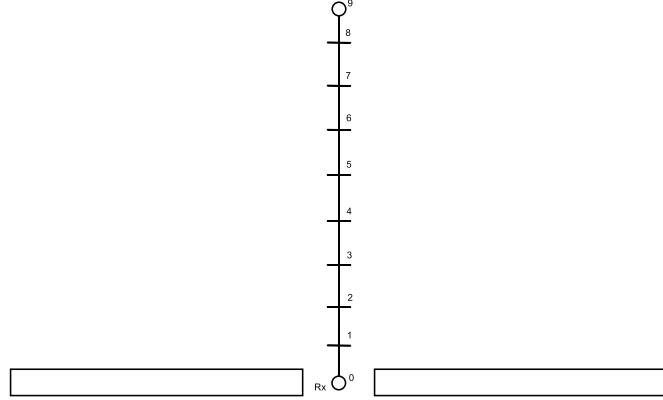
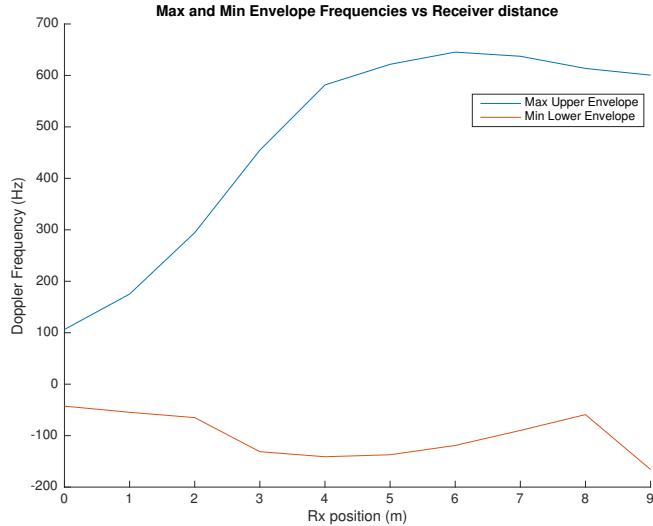


FIGURE 4.3: Position of the Receiver in Relation to the Rotor Blades

FIGURE 4.4: Max and Min Doppler vs. Receiver Position with Tx Azimuth at  $135^\circ$  and Elevation at  $54^\circ$ 

directly under neath the rotor to past the tip of the rotor at 9m. The position of the receiver also effects the resulting signal envelope shapes shown in figures 4.5 and 4.6.

Comparing figure 4.5 with 4.6 the max doppler reflects the information in figure 4.4 and the overall envelope is afected. The envelope in figure 4.5 is much noiser than the envelope of figure 4.6 and the envelope of this second figure has a more distinct shape with less variation between its two peaks. The position of the receiver will also determine whether the azimuth angle can be derived from the resulting signal shown

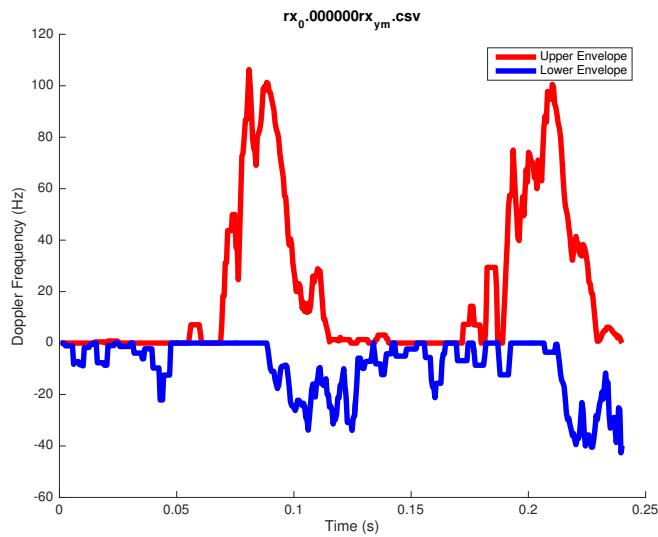


FIGURE 4.5: Doppler Envelope with receiver at 0m

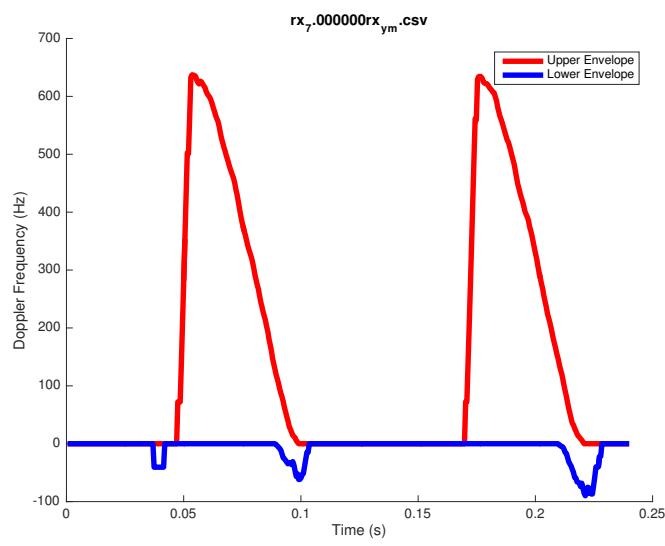


FIGURE 4.6: Doppler Envelope with receiver at 7m

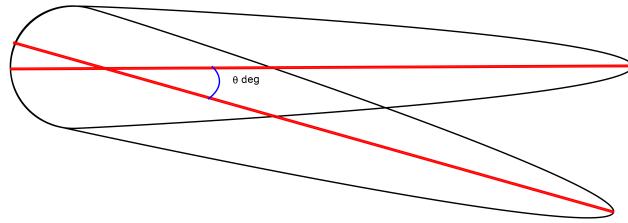


FIGURE 4.7: Rotor Pitch

later in section 4.5. Therefore the receiver will be offset from the center of rotation by 7m for the subsequent simulations.

#### 4.4 Rotor Pitch

The pitch of the rotor blades determines the amount of lift the rotors create. The pitch range is determined by the shape of the airfoil and the angle at which it will stall and no longer produce lift. For the purpose of the simulations, and the selected airfoil, the pitch range will be from  $0^\circ$  to  $15^\circ$ .

The resulting signal is analyzed by calculating its doppler envelope. Then the maximum and minimum of that envelope will then be used to determine the overall effect of pitching the rotor blade.

From figure 4.8 the effect of pitch on the maximum doppler frequency is a linear increase between a pitch from  $0^\circ$  to  $15^\circ$ . Due to the linear increase the pitch of the rotor blade can be accounted for when making measurements on subsequent simulation analysis. One issue does arise in when the transmitter is located underneath the helicopter which is shown in figure 4.9.

As the pitch increases linearly at first then climbs at a much faster rate. This offset when the transmitter is directly underneath is caused by the pitch of the blade reflecting rays into the receiver which would be reflected back at the ground if the pitch was  $0^\circ$ . If the assumption is that the helicopter is hovering when these measurements take place the pitch of the rotor will be much less than the maximum pitch available. This assumption

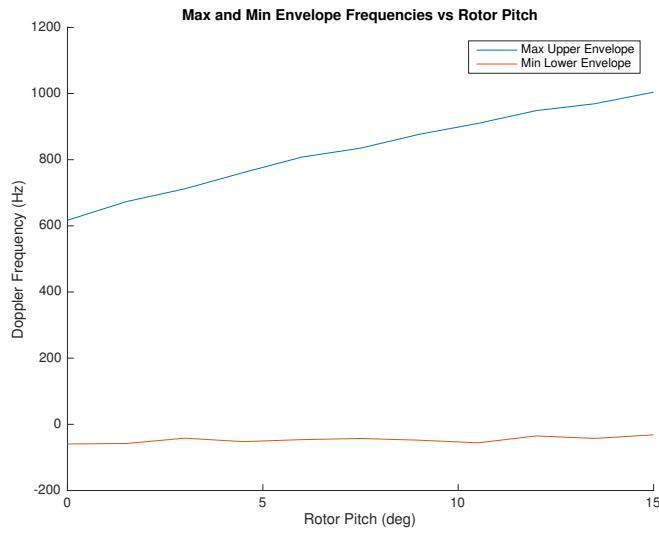


FIGURE 4.8: Max and Min Envelope Frequencies vs Rotor Pitch with Tx Azimuth at  $135^\circ$  and Elevation at  $54^\circ$

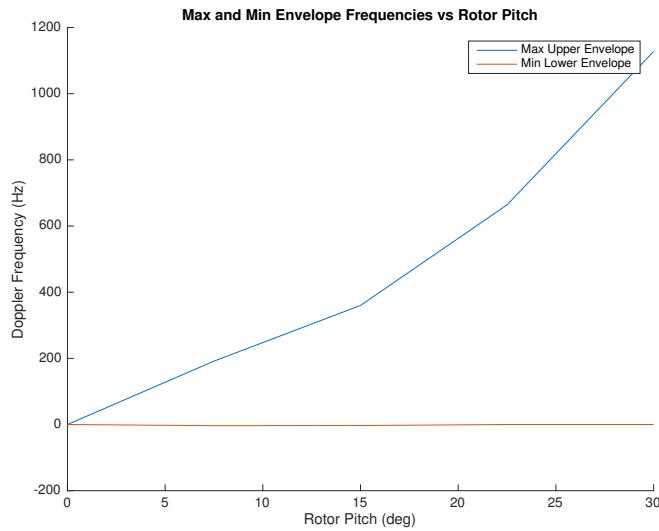


FIGURE 4.9: Max and Min Envelope Frequencies vs Rotor Pitch Transmitter Underneath the Helicopter

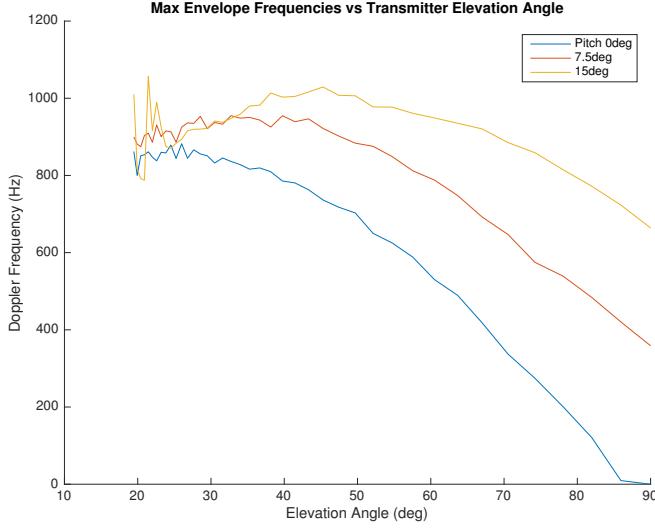


FIGURE 4.10: Max Envelope Frequencies vs Transmitter Elevation Angle

will adjust the maximum pitch to  $15^\circ$  which will allow for the linear assumption to be fulfilled even when the transmitter is directly underneath the helicopter.

In figure 4.10 the max doppler vs elevation angles are initially shifted up in frequency from a pitch of 0 and has changed in slope. This is because the rays hitting the blades from below are reflecting into the receiver at an angle that is closer to the direction of blade travel. The interesting thing about this plot is that the pitch envelopes actually become closer to the  $0^\circ$  envelope as the elevation angle becomes smaller. Figure 4.11 shows the difference plotted over elevation angle.

From figure 4.11 we can see that the difference in frequency changes linearly as the elevation angle decreases, the slope of which is based on the pitch angle. This slope would have to be characterized on the physical craft and can be accounted for due to its predictable nature.

## 4.5 Transmitter Azimuth Angle

The transmitter azimuth angle corresponds to the cardinal direction of the transmitter with respect to the front of the helicopter.

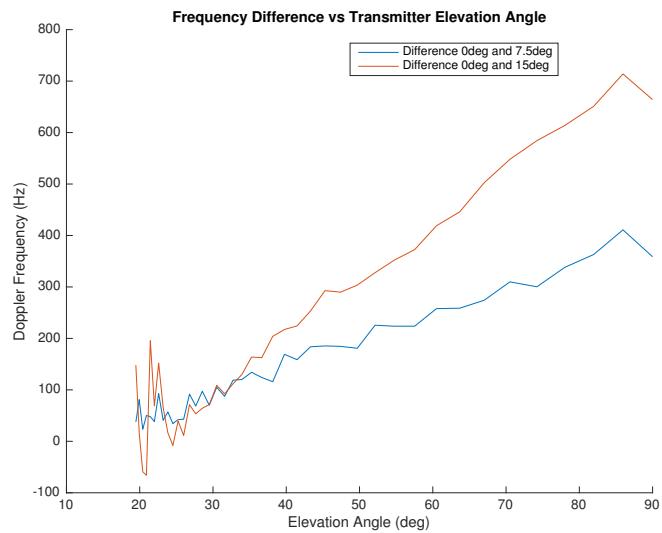


FIGURE 4.11: Frequency Difference vs Transmitter Elevation Angle

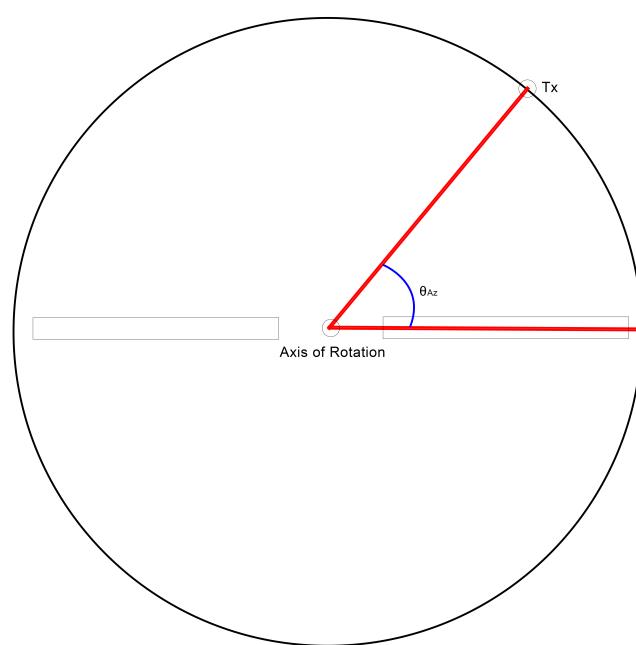


FIGURE 4.12: Transmitter Azimuth Angle Relationship

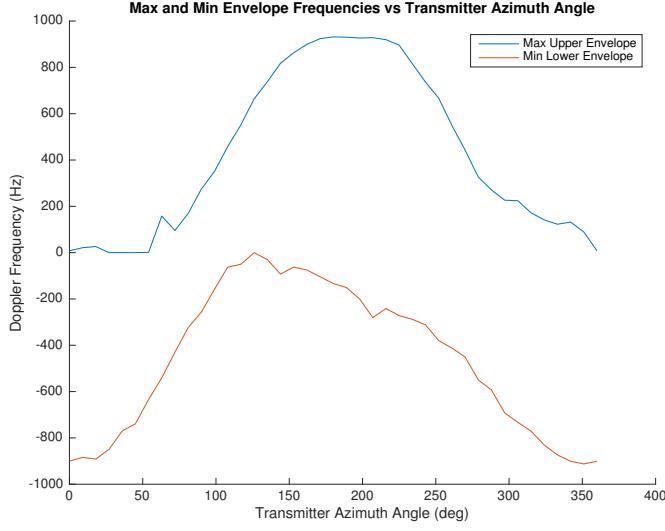


FIGURE 4.13: Max and Min Envelope Frequencies vs Transmitter Azimuth Angle with Tx Elevation at 45°

Where  $\theta_{Az}$  is the azimuth angle. The azimuth angle of the transmitter in the simulation will effect the amount of doppler added to the signal. This is due to the direction of the rotor blade rotation as it approaches both the receiver and the transmitter and how the rays reflect off the blade as it moves. From the model we see the amount of doppler form a sinusoidal pattern as the azimuth angle goes from 0 to  $2\pi$ . This is mirrored in the simulation shown in figure 4.13.

The graph in figure 4.13 shows both the max and the minimum measured doppler as the azimuth angle completes a full rotation. The doppler frequency follows the azimuth angle in a sinusoidal fashion. The sinusoidal pattern is held as the elevation angle changes shown in figures 4.14 and 4.15.

The sinusoidal pattern of min and max doppler frequencies are only present when the receiver is located away from the axis of rotation. This is shown in figure 4.16 which is devoid of any sinusoidal pattern and remains constant as the transmitter azimuth angle is changed.

Therefore all future simulations will set the receiver near the tip of the rotor which would simulate the receiver being in the nose of the helicopter.

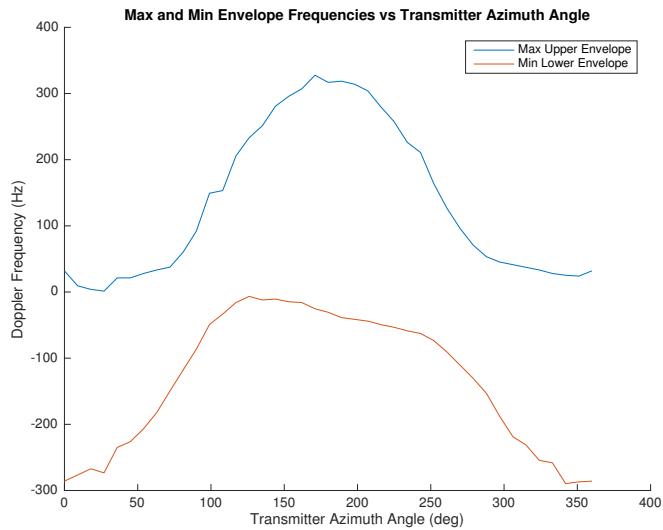


FIGURE 4.14: Max and Min Envelope Frequencies vs Transmitter Azimuth Angle with Tx Elevation at  $76^\circ$

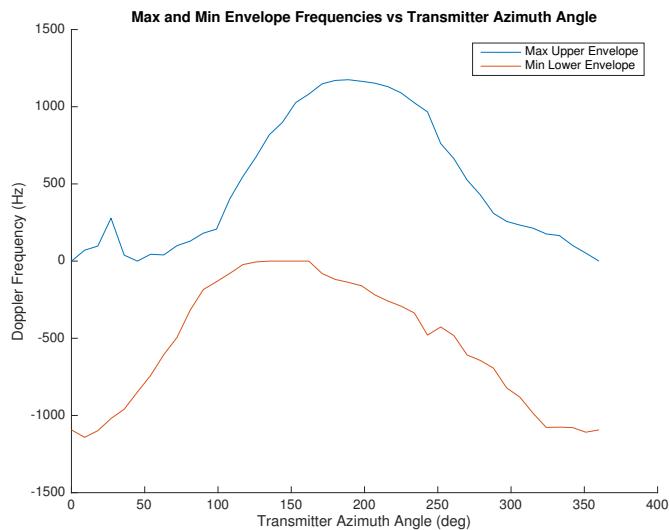


FIGURE 4.15: Max and Min Envelope Frequencies vs Transmitter Azimuth Angle with Tx Elevation at  $26.5^\circ$

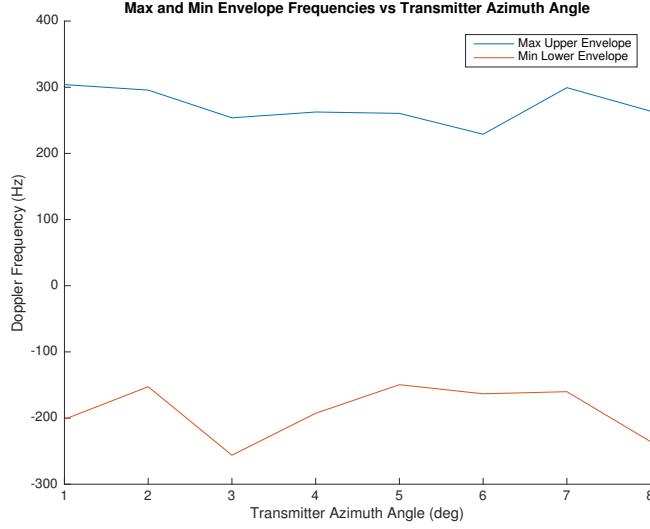


FIGURE 4.16: Max and Min Envelope Frequencies vs Transmitter Azimuth Angle with Rx at Axis of Rotation

The effect of pitching the blade is that it modifies the results of the azimuth sweep when compared to figure 4.13. The resulting min and max envelope frequencies vs azimuth angle for pitch are shown in 4.17. This occurs because the pitch of the blade angles RF reflections more in the direction of travel but not at all points in the azimuth. When the transmitter is located between 90° and 270° pitch causes an increase in doppler but elsewhere decreases the amount of doppler shift. The decrease is caused by the blade surface top reflecting receding doppler away from the receiver at those angles.

The effect of azimuth angle on the maximum measured doppler will be characterized in section 5.2

## 4.6 Transmitter Elevation Angle

The transmitter elevation angle is the angle between the helicopter rotor and the position of the transmitter shown in figure 4.18.

Where  $\theta_{El}$  is the elevation angle.

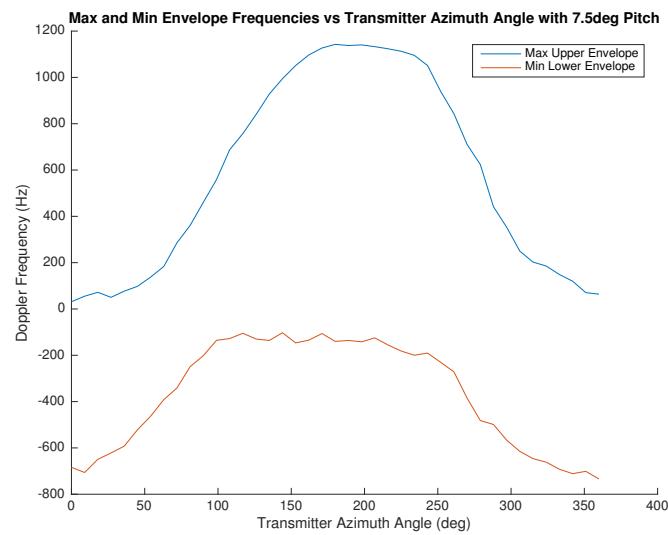


FIGURE 4.17: Max and Min Envelope Frequencies vs Transmitter Elevation Angle with 7.5deg Pitch

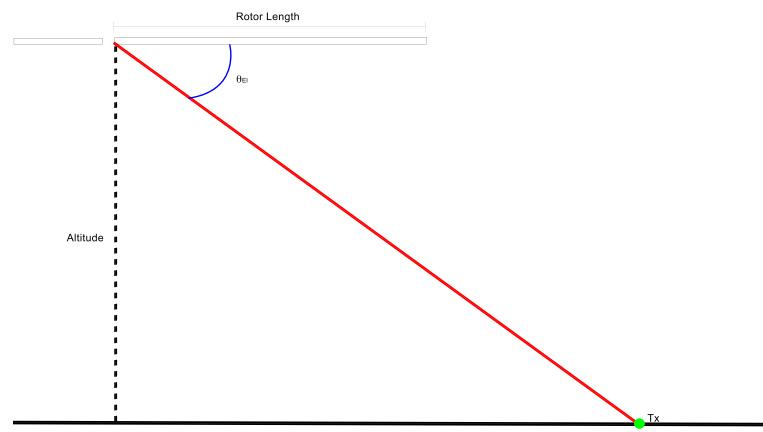


FIGURE 4.18: Transmitter Elevation Angle

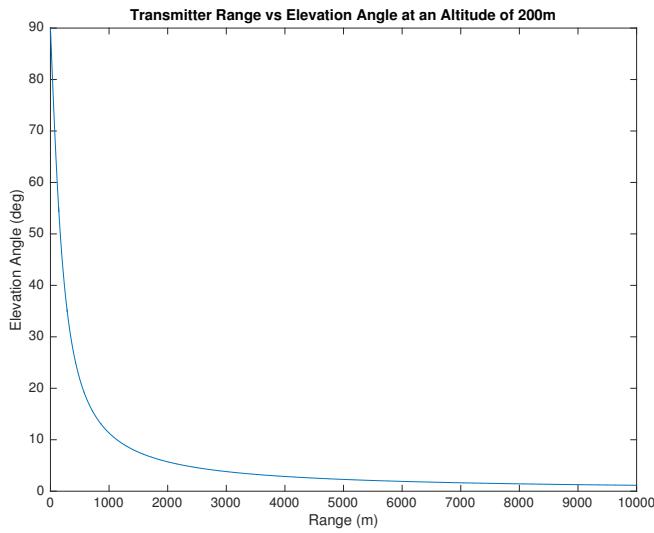


FIGURE 4.19: Transmitter Range vs Elevation Angle at an Altitude of 200m

The smaller elevation angles mean that the transmitter is farther away from the helicopter. The elevation angle in relation to the distance of the transmitter is shown by figure 4.19.

The elevation angle of the transmitter in the simulation will effect the amount of doppler added to the signal. This is due to the rays having a vector component in the direction of the travel of the rotor. When the transmitter is directly underneath the rotor and receiver there is no vector component in the direction of rotor travel so the reflected signal is not doppler shifted. From the model we see the amount of doppler increase as the elevation angle decreases. This is mirrored in the simulation shown in figure 4.20.

The graph in figure 4.20 shows the maximum measured doppler as the elevation angle increases.

Looking back at the geometric models for transmitter azimuth angles of  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$  and their simulated counterparts shown in figure 4.21.

The effect of elevation angle on the maximum measured doppler will be characterized in section 5.3.

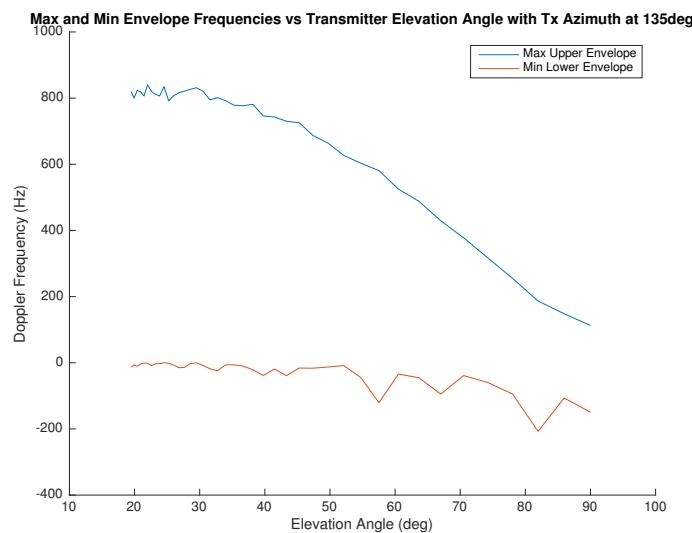


FIGURE 4.20: Max and Min Envelope Frequencies vs Transmitter Elevation Angle with Tx Azimuth of 135deg

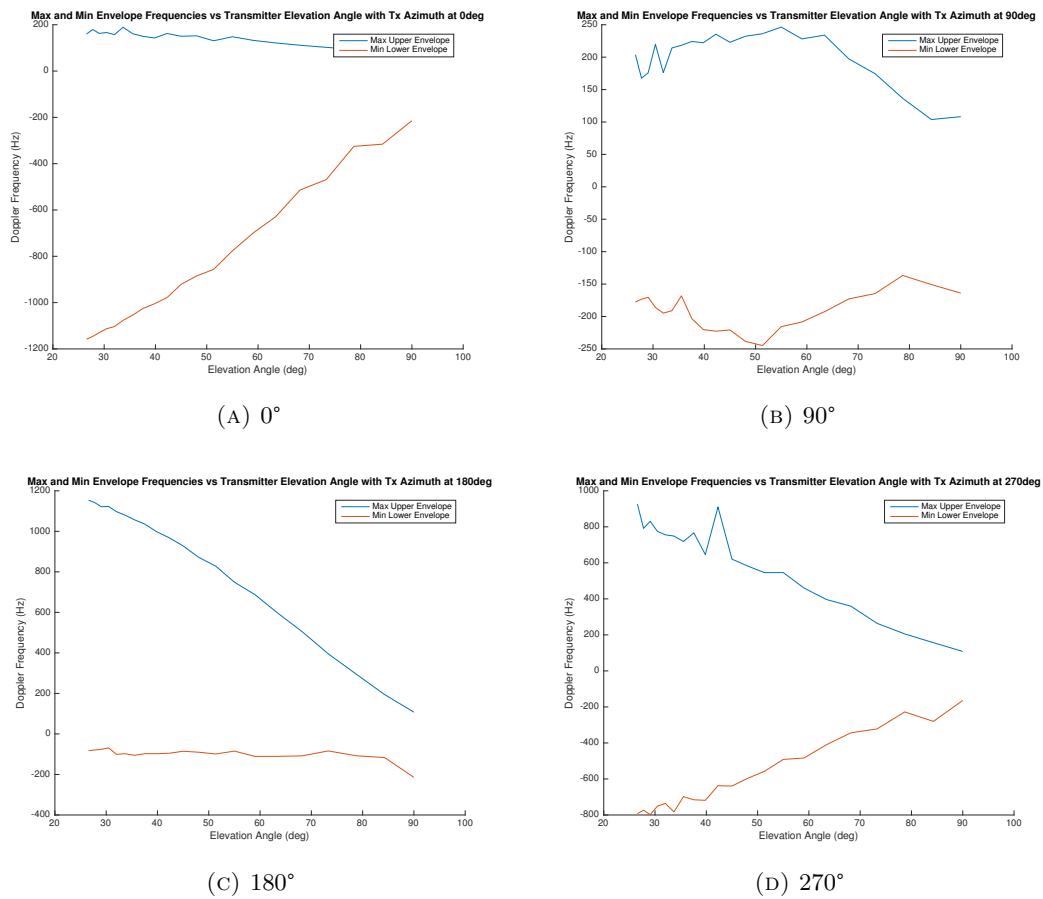


FIGURE 4.21: Doppler Profile vs. Transmitter Elevation angle for various Azimuth angles.

# Chapter 5

## Results

### 5.1 Introduction

The information gathered from the previous section is used to estimate the azimuth and elevation angles of the transmitter. First the azimuth angle will be estimated which will provide the information necessary for calculating the elevation angle. The results of the estimation techniques are then analyzed to determine their accuracy compared to the actual.

### 5.2 Transmitter Azimuth Angle Estimation

The azimuth angle estimation technique is based on the data collected by rotating the helicopter platform in a clockwise manner, during which at a set angle a doppler envelope is calculated and subsequent maximum and minimum values are collected. After a complete rotation the doppler maximum and minimum values produce a sinusoidal pattern as shown in figure 4.13 without the assumption of knowing the starting azimuth angle. With this data we need to find an unambiguous angle that will set the coordinate space and provide an estimate of current azimuth angle. To do this the absolute value of the minimum envelope data is superimposed onto the maximum envelope data as shown in figure 5.1.

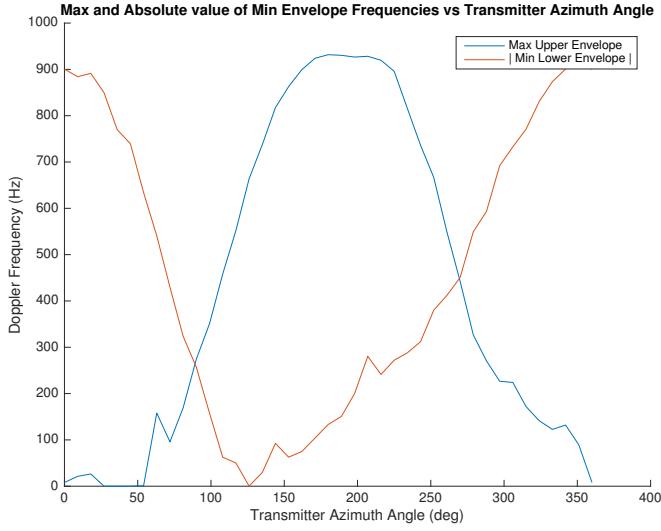


FIGURE 5.1: Max and Absolute value of Min Envelope Frequencies vs Transmitter Azimuth Angle at an Elevation Angle of  $45^\circ$

From figure 5.1 there are two points at which the maximums and the absolute value of the minimums cross. By computing the difference 5.1

$$\text{EnvelopeDifference} = \text{MaxUpperEnvelope} - |\text{MinLowerEnvelope}| \quad (5.1)$$

we produce an almost exact sinusoid centered around zero, shown in figure 5.2.

While figure 5.2 provides information about the angle of the transmitter it does not contain enough to locate an unambiguous azimuth angle in itself, therefore by computing the correct doppler to use from the *MaxUpperEnvelope* and the  $|\text{MinLowerEnvelope}|$ . This is done by evaluating the difference at each angle and if the difference is negative the  $|\text{MinLowerEnvelope}|$  is used and if positive the *MaxUpperEnvelope*.

```

1 %corrected Doppler frequency for processing
2 for i = 1:length(upper)
3     if(difference(i) < 0)
4         correct_fd(i) = abs(lower(i));
5     else
6         correct_fd(i) = upper(i);

```

```

7      end
8  end

```

The correct Doppler is shown in figure 5.3 which displays the amount of doppler that is currently acting on the signal for that specific azimuth angle. The data gathered

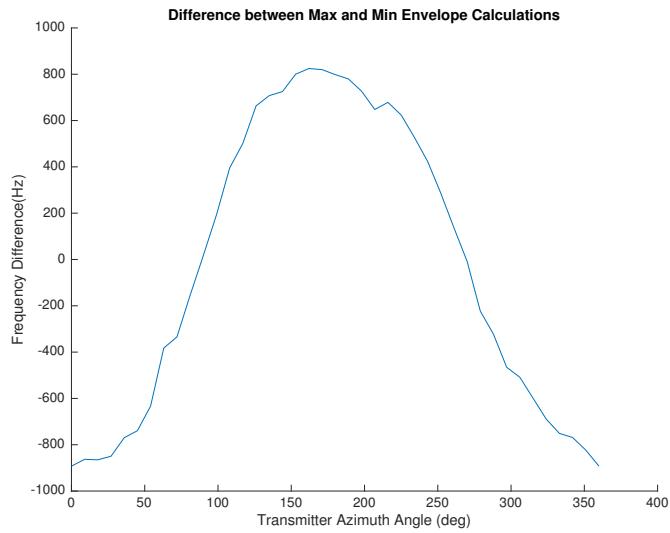


FIGURE 5.2: Difference between Max and Min Envelope Calculations vs Transmitter Azimuth Angle at an Elevation Angle of 45°

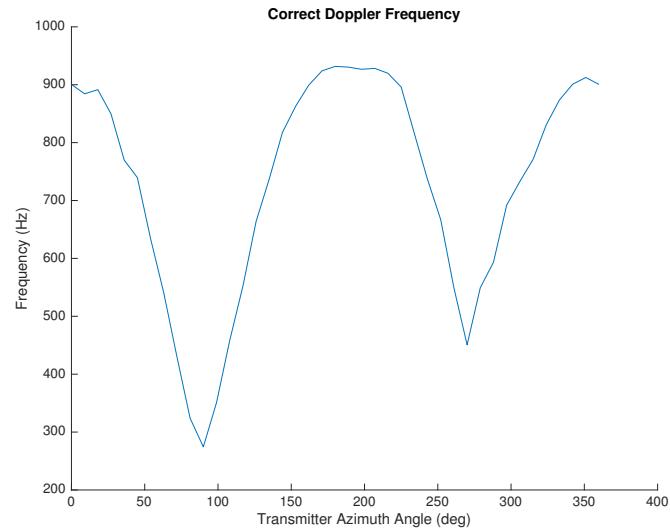


FIGURE 5.3: Correct Doppler Frequencies vs Transmitter Azimuth Angle at an Elevation Angle of 45°

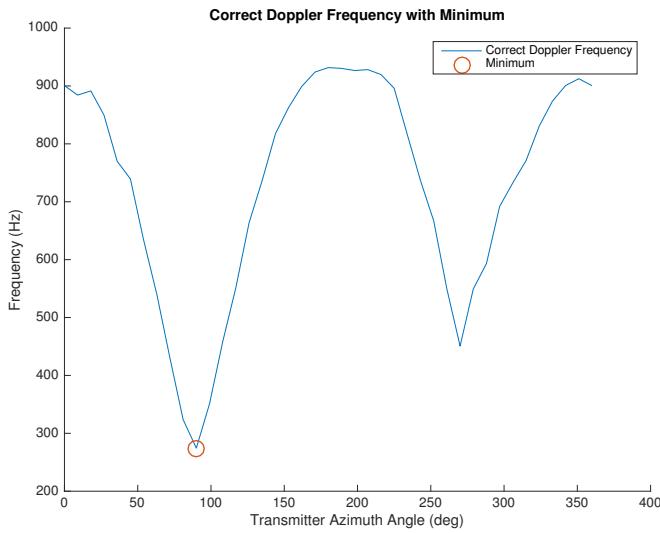


FIGURE 5.4: Correct Doppler Frequencies with Minimum vs Transmitter Azimuth Angle at an Elevation Angle of 45°

contains a global minimum at 90° and a second local minimum at 270°. The 90° can be found easily by taking the minimum of the correct Doppler signal

```

1 %Minimum of corrected Doppler
2 [Min,I_min] = min(correct_fd);
```

But in order to find the local minimum at 270° we need to use a different method. To do this we can use the peak detection function within Matlab to locate both the 90° and 270° point. By taking the inverse of the correct Doppler and tuning the peak detection function with the maximum and minimum values to determine accurate prominence.

```

1 %peak detection with prominence calculation
2 min_peak_prominence = (1/(Max - Min))*.5;
3 [pk,lc] = findpeaks(1./correct_fd, 'NPeaks', 2, 'MinPeakProminence', ...
    min_peak_prominence);
```

From figure 5.5 we see the two peaks denoted by the red x's, one at 90° and the other at 270°. Now we have a minimum value that should denote the 90° location and the

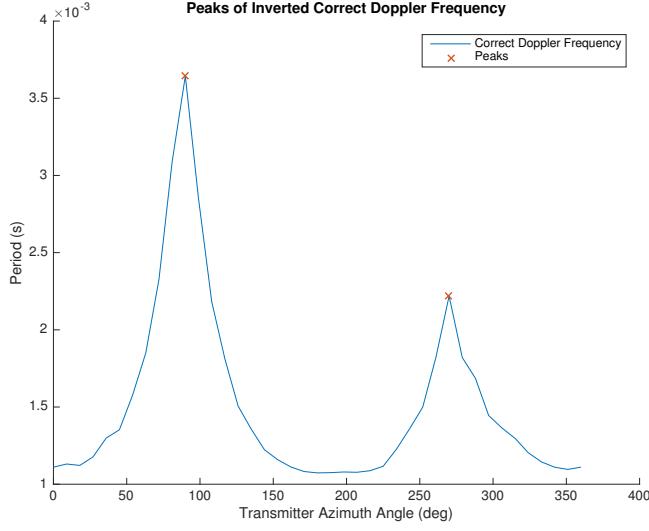


FIGURE 5.5: Peaks of inverted Correct Doppler vs Transmitter Azimuth Angle at an Elevation Angle of  $45^\circ$

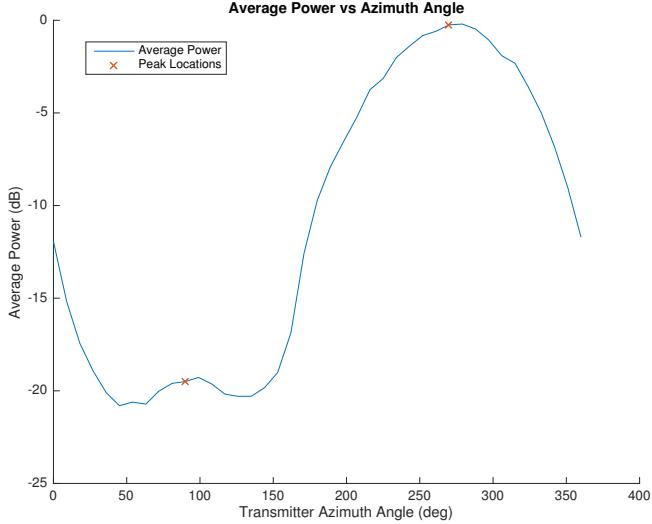


FIGURE 5.6: Average Power vs Transmitter Azimuth Angle at an Elevation Angle of  $45^\circ$

two peak locations that denote the  $90^\circ$   $270^\circ$  locations. Because it is possible that the minimum could find the wrong peak for the  $90^\circ$  location we can compare the peak locations on the average power data, that was used to find the successive average during data collection, to determine which location is actually  $90^\circ$ .

Altitude	Elevation Angle	Azimuth Error in (deg)	Percent Error
200	76°	0	0
200	45°	0	0
200	26.5°	0	0
500	78.7°	0	0
500	51.34°	0	0
500	35.5°	0	0

TABLE 5.1: Azimuth Error and Azimuth Percent Error of Different Elevation Angles During Azimuth Estimation

Altitude	Elevation Angle	Percent Peak Difference	Power Difference in dB	Peak Frequency Difference Hz
200	76°	-16.8675%	-14.0914	18.67
200	45°	-39.1124%	-19.7476	176.267
200	26.5°	-69.7368%	-13.3443	424.0
500	78.7°	-21.6667%	-11.4508	17.33
500	51.34°	-38.1875%	-22.4610	162.93
500	35.5°	-53.4303%	-18.6043	313.6

TABLE 5.2: Percent Peak, Power, and Frequency Difference of Different Elevation Angles During Azimuth Estimation

Figure 5.6 shows the average power at each angle sample with the angles 90° and 270°denoted. From the plot we can see that the average power at 90° is much smaller than the the power at 270°, so this inequality will be used to resolve ambiguity in the azimuth angle.

From there the Minimum method and the peak method can be compared to see if they agree on the 90° location. If this is not the case one will be chosen and the helicopter can fly in that direction and measure based on the physical movement of the helicopter or if the resulting envelope got larger or smaller because of the change in elevation angle.

From table 5.1 we can see that there is no error in the estimation of the 90° point over a variety of Altitude and elevation angles that were tested. But because of the algorithm could potentially pick the 270°we will analyze the difference in peak heights and the power difference to evaluate the estimation further.

From table 5.2 we can see that at the larger elevation angles the peak percent difference are significant to determine which is correctly the 90°value, but from the Peak frequency

difference at the large elevation angles the difference is in the tens of Hz so in practical applications it is possible to confuse the estimation by just taking the minimum frequency so other values need to play a role such as the power.

### 5.3 Transmitter Elevation Angle Estimation

Using the information gathered from the azimuth estimation procedure we can compute the elevation angle of the transmitter. In the process of gathering azimuth information the maximum and minimum envelopes follow a sinusoidal pattern that contains a maximum Doppler information at approximately  $180^\circ$  and  $0^\circ$  azimuth angles. Using that maximum value along with the correct Doppler and the assumption that the maximum will occur near the tip of the rotor blade we can approximate the respected reflection radii off the blade as the azimuth angle changes. This is done through

$$R_{Estimate} = BladeLength \left( \frac{f_{DopplerCorrect}(\theta_{Az})}{Max(f_{DopplerCorrect})} \right) \quad (5.2)$$

where  $R_{Estimate}$  is the estimated reflection radius at a particular azimuth,  $BladeLength$  is the length of a single blade or radius of the rotor, and  $f_{DopplerCorrect}$  is the correct doppler frequency for each azimuth angle  $\theta_{Az}$ .

Figure 5.7 shows the estimated reflection radius for each azimuth angle.

Now we find the maximum amount of Doppler that could be induced at that reflection radius location through

$$f_{DopplerREstimateMax} = 2 \frac{v_r f_c}{c} \quad (5.3)$$

where  $f_{DopplerREstimateMax}$  is the maximum amount of doppler that could happen at that location if the signal was reflected parallel to the direction of motion.  $f_c$  is the center frequency of the signal,  $c$  is the speed of light and  $v_r$  is the radial velocity based of the *RPM* defined by

$$v_r = \frac{2\pi}{60} RPM R_{Estimate} \quad (5.4)$$

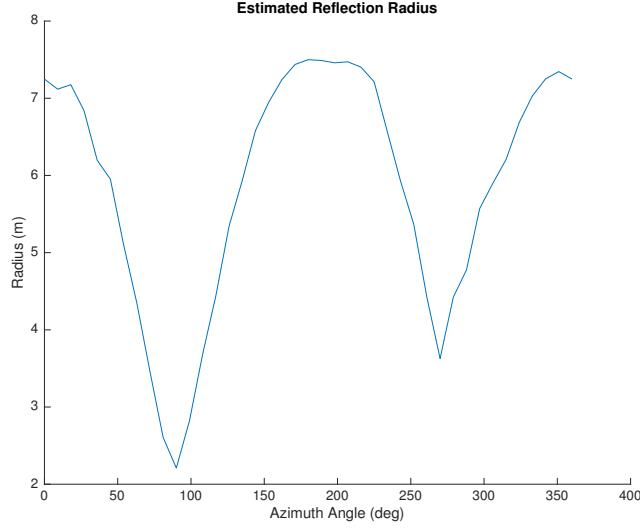


FIGURE 5.7: Estimated Reflection Radius at an Elevation Angle of  $45^\circ$

Then to find the elevation angle we look back at equation (-) where we then solve for the elevation angle

$$\theta_{El} = \arccos \left( \frac{f_{DopplerCorrect}(\theta_{Az})}{f_{DopplerREstimateMax}} \right) \quad (5.5)$$

where  $\theta_{El}$  is the elevation angle of the transmitter. If the elevation angle is all the information needed equation 5.5 distills down to

$$\theta_{El} = \arccos \left( \frac{\text{Max}(f_{DopplerCorrect})}{f_{DopplerMax}} \right) \quad (5.6)$$

where  $f_{DopplerMax}$  is defined in equation 5.11.

Figure 5.8 shows the estimation of the elevation angle  $\theta_{El}$  with respect to the azimuth angle, what it shows is that the elevation angle remains constant with respect to azimuth angle. which means that the estimation in the reflection radius is correctly compensating for the azimuth angle. The percent error off the actual elevation angle is shown in figure 5.9 which is very small and consistent over the azimuth range.

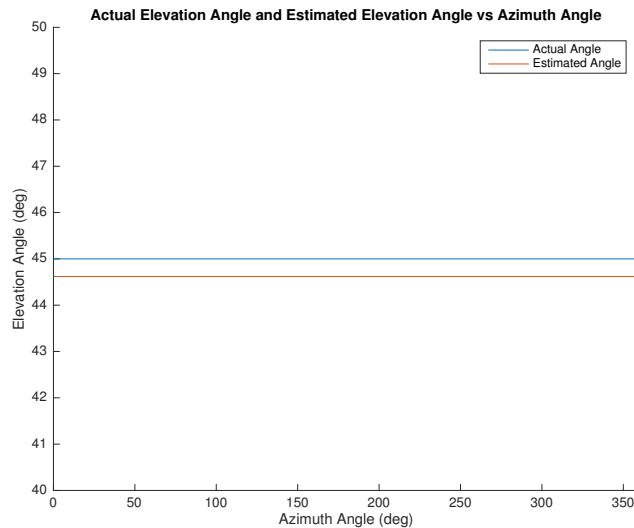


FIGURE 5.8: Actual Elevation Angle and Estimated Elevation Angle vs Azimuth Angle

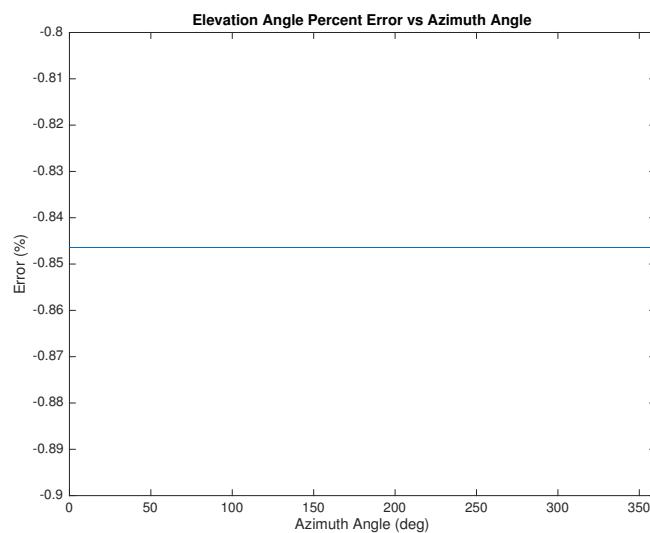


FIGURE 5.9: Elevation Angle Percent Error vs Transmitter Azimuth Angle

Altitude	Elevation Angle	Average Elevation Error in (deg)	Average Percent Error
200	76°	-0.4632°	-0.6097%
200	45°	-0.3809°	-0.8464%
200	26.5°	-0.4072°	-1.5329%
500	78.7°	0.0914°	0.1162%
500	51.34°	-0.6236°	-1.2146%
500	35.5°	-0.6803°	-1.9143%

TABLE 5.3: Average Elevation Error and Average Elevation Percent Error of Different Elevation Angles During Azimuth Estimation

From table 5.3 elevation estimation error is extremely small for the given actual elevation angles. But there is a trend that is evident in the data that as the elevation angle decreases the amount of error is increasing.

This error is caused by a limitation on the estimation is that the doppler shift is limited by the maximum amount doppler that the blade rotation can produce shown in equation 5.4 by replacing  $v_r$  with  $v_{rMax}$  which is calculated by

$$v_{rMax} = \frac{2\pi}{60} RPM Blade Length \quad (5.7)$$

then using that to find the maximum available doppler through

$$f_{DopplerMax} = 2 \frac{v_{rMax} f_c}{c} \quad (5.8)$$

This limitation is shown by varying the elevation angle at the azimuth angle that provides the maximum Doppler shift, which is 180° . The maximum and minimum envelope data is shown in figure 5.10

Using the estimation method described above, figure 5.11 shows the elevation angle estimated versus the actual angle. The estimation method is accurate till about 18° or 20° where it becomes noisy. This is due to the inability of the envelope estimation to resolve the correct maximum doppler for that range. By going back to figure (anglerangerelation) we can see that as the transmitter moves farther and farther away the elevation

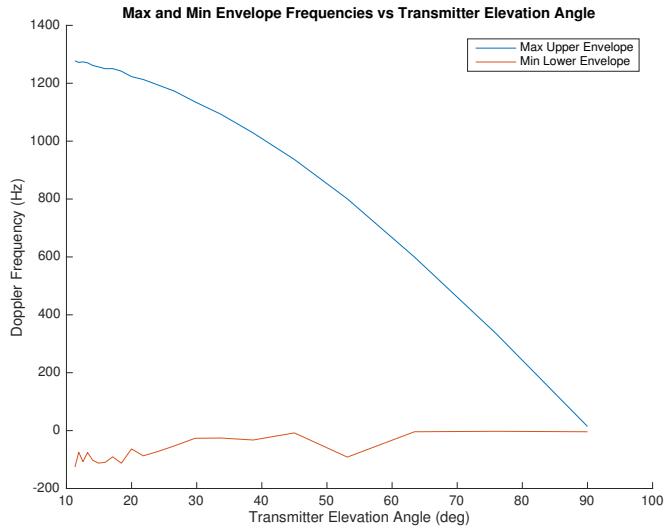


FIGURE 5.10: Max and Min Envelope Frequencies vs Elevation Angle with Transmitter Azimuth Angle of 180°

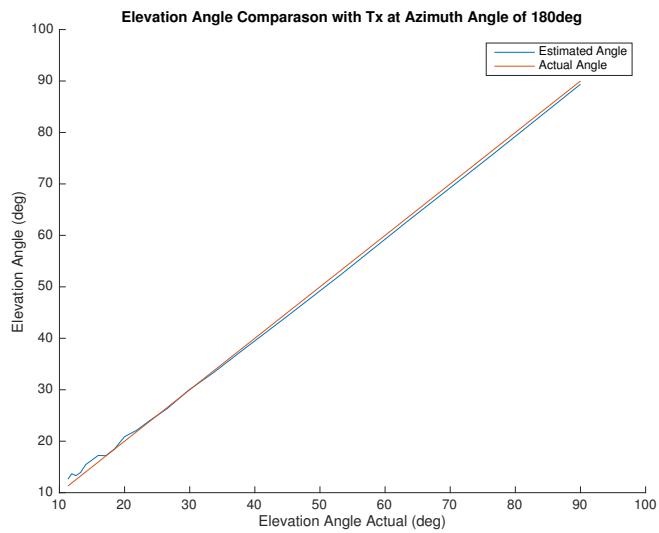


FIGURE 5.11: Elevation Estimate and Actual Angle with Transmitter Azimuth Angle of 180°

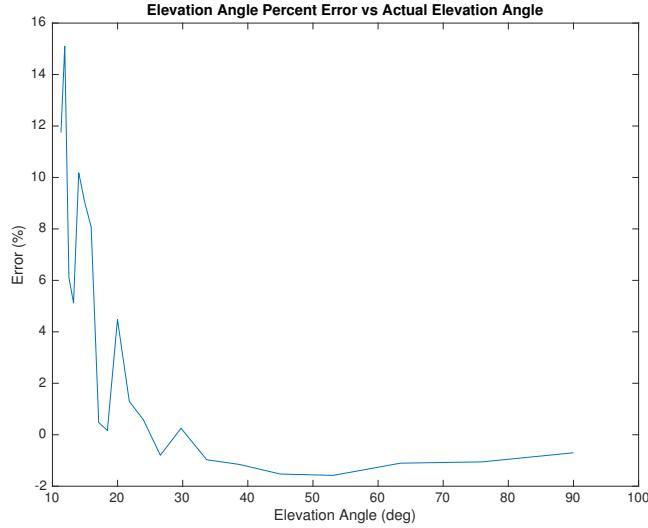


FIGURE 5.12: Elevation Angle Percent Error with Transmitter Azimuth Angle of 180°

angle is still decreasing but at a much smaller and smaller rate. So the difference between elevation angles closer to zero are more difficult to resolve.

Figure 5.12 shows the percent error with respect to the elevation angle and it starts to become unusable around 18° or 20° where the percent error starts to shoot past acceptable ranges.

If we now consider a pitched blade the above method for estimating Azimuth and elevation becomes inaccurate due to the increased doppler between 90° and 270° and the decreased doppler on the other side. To correct this error we need to correct both the difference calculation and the correct doppler data accordingly. The uncorrected data is shown in figure 5.13 and 5.14 with percent azimuth error of 20% and percent elevation error of 35%.

Comparing the envelope difference of no pitch and pitched blade, in figures 5.2 and 5.13 respectively, we can see that the difference calculation of the pitch is a shifted up version of the one with no pitch. Therefore we can first correct this by re-centering the pitch difference according to

$$A_{shift} = \frac{Max(MaxUpperEnvelope) - |Min(MinLowerEnvelope)|}{2} \quad (5.9)$$

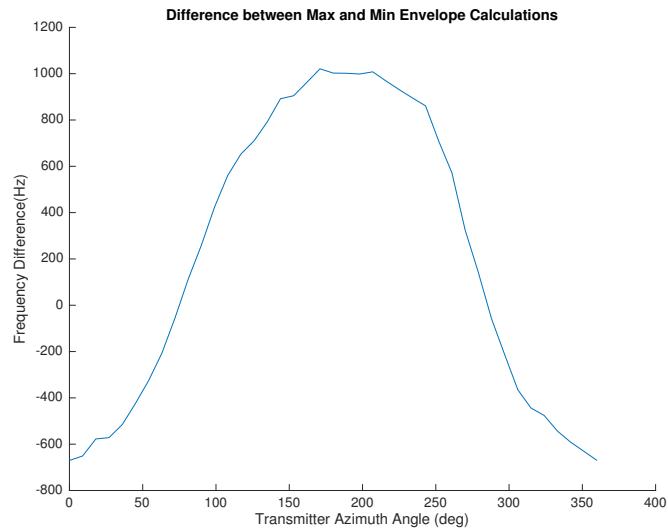


FIGURE 5.13: Uncorrected Envelope Difference vs. Transmitter Azimuth and Elevation angle of 45°with Blade pitch of 7.5°

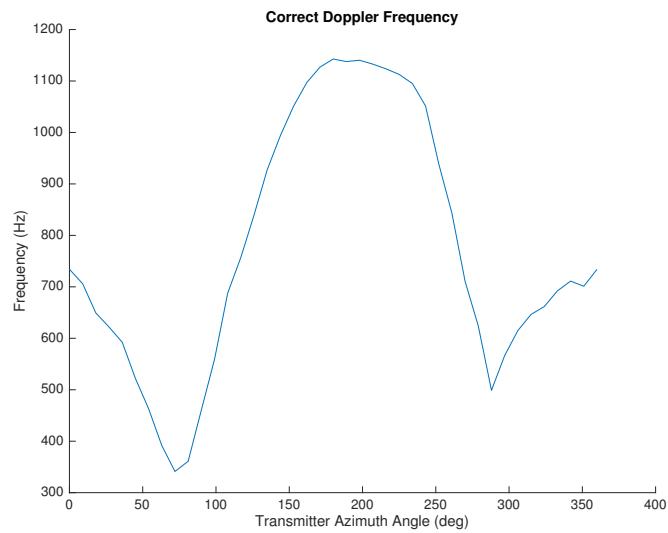


FIGURE 5.14: Uncorrected Doppler vs. Transmitter Azimuth and Elevation angle of 45°with Blade pitch of 7.5°

where  $A_{shift}$  is the amount that the difference calculation will be shifted down

$$EnvelopeDifference = EnvelopeDifference - A_{shift} \quad (5.10)$$

Then with this knowledge  $f_{DopplerCorrect}$  is recalculated and then is adjusted to the correct frequencies with

$$f_{DopplerCorrect} = f_{DopplerCorrect} + A_{shift}\cos(\theta_{Az}) \quad (5.11)$$

Since  $\theta_{Az}$  is unknown during the calculation of this equation the correct starting phase can be determined by matching it to the  $EnvelopeDifference$  with the a priori information that the cosign should be one period in length. These corrections result in the percent azimuth error of 0% and 7% elevation angle error. The elevation angle percent error is within a useable range but could be decreased with further tuning to the preliminary pitch adjustment algorithm. When the pitch corrections are applied to a transmitter elevation angle of 26.56° the correction achieves better performance than the unpitched calculation with percent azimuth error of 0% and percent elevation error of 0.114%. This is still a preliminary correction method that requires more in depth analysis but from the initial results shows merit. The increased accuracy as the elevation angle decreases relates back to figure 4.10 where as the elevation angle decreases the effect of pitch on the signal converges back to unpitched results.

# **Chapter 6**

# **Conclusion and Future Work**

## **6.1 Summary**

Over the course of this thesis a three dimensional ray tracer application was built in C++ to produce a RBM signal according to a variety of physical parameters. The physical parameters associated with a helicopter blade and the positioning of transmitter and receiver hardware were swept independently to build knowledge on how they affect the resulting signal. That data was then processed using time frequency analysis to produce a doppler envelope. An envelope of each parameter step was used to determine an overall trend. Then with those trends an algorithm was derived to find the azimuth and elevation angle of the transmitter with relationship to the helicopter platform. The algorithm accurately finds the azimuth angle and elevation using the data collected in the simulation environment with almost no error in the azimuth estimation and less than 2% error in elevation angle when blade pitch in 0°.

## **6.2 Future Work**

The first thing that needs to be done in continuing this research is to perform a real world experiment to officially evaluate the ray tracing application. To do this appropriate hardware needs to be defined in the form of a software defined radio for both the receiver

and transmitter along with a software workflow such as GNU Radio. Next a small scale environment would need to be set up, including the above hardware along with some form of rotating metal blade or thin plate rotating at a sufficient rate to produce a doppler spread similar to those defined in the simulation. After that data is collected and analyzed a full scale experiment would need to be conducted including a real helicopter platform and an environment with large expanses of flat ground.

Once the real world experiments have been conducted, and are shown to be similar to the signal produced by the simulation, the estimation techniques defined in this thesis can be evaluated on the real signal to determine if they are sufficient in practice.

After the physical validation the ray tracing application can be modified to include more physical phenomenon. The other improvement would be to introduce more objects into the scene model which would require modification to the tracing algorithm to include ray casting distributions in their particular direction.

Another improvement on the simulation would be to increase the speed of computation. This could be done by implementing multi-threading or, since raytracing is heavily parallelizable, a GPU based architecture. The multi-threading option would require minor code modifications but realizing the simulation on a GPU would require a full rewrite of the software to take advantage of the GPU's abilities.

The current estimation does not take into account the physical pitch value when correcting for pitch and might not cover all the cases that pitch can effect. Therefore further investigation is needed into how the pitch affects the received signal for improved azimuth and elevation estimation.

The estimation methods described in this thesis rely only on the maximum and minimum values of the doppler envelopes calculated but their shape could also be of particular interest as well. When varying the parameters of azimuth and elevation the shape of the envelope changes in a distinctive manner, and by using a training algorithm the position of the transmitter might be able to be tracked in time by evaluating the change in shape. where currently if we want to locate the transmitter in a successive fashion a full revolution of the helicopter platform is needed to resolve position ambiguity.

## Appendix A

# Ray Tracing Software

The ray tracing software developed during the course of the thesis is written in C++ in an XCode environment, but a generic implementation is provided to run on other operating systems. This code is loosely based on the work of Eric Degroot's developed in (Eric's thesis) and implemented some modified base classes developed by him. It is a specific tool for analyzing the affects of rotor blade modulation on a signal tone, it does not provide generic 3D ray tracing in order to optimize the performance for this task. The software is provided without a GUI implementation and is run on different parameters by modifying the main.c file. This is due to the amount of parameters involved but could be easily extended to provide command line arguments in future iterations.

To use the software for performing RBM simulations there are several key parameters that are needed. First a file containing the points that define the 2D section of the airfoil shape, one default file is included in the software package for ease of use and as a template for other airfoil shapes.

the other key parameters that define the simulation are shown in table [A.1](#)

The output file also needs to be defined with a unique name that changes, this is important because the program will just write over the old file if you do not have a unique

Name	type	Default Value	units	Description
rx_x	double	0	m	receiver position in x
rx_y	double	7	m	receiver position in y
rx_z	double	2	m	receiver position in z below rotor
Bandwidth	double	2000	Hz	Bandwidth of the receiver
rx_fc	double	1000000000	Hz	center frequency of the receiver
tx_x	double	100	m	transmitter position in x
tx_y	double	100	m	transmitter position in y
tx_fc	double	1000000000	Hz	transmitter center frequency
tx_power	double	4000		transmitter power
num_blades	int	2	none	number of blades
RPM	double	250	RPM	Blade revolutions per minuet
altitude	double	200	m	rotor altitude
blade_length	double	7.5	m	length of the one rotor blade
pitch	double	0	radians	pitch of the rotor use $-\theta_p$
num_ribs	double	10	none	number of ribs that makes up one blade
files	double	10	none	number of files to produce

TABLE A.1: Simulation parameters

naming scheme. this could be just an increasing number or related to one of the variable sweeps.

To run the simulation on a Mac load into XCode the project file and configure the main file to reflect the type of simulation desired, then hit the play button.

On Linux machines the code must first be built using make with the main file configured before hand. Then run the Raytracer3D executable.

1	2	3	4	5	6	7	8	9	10
rx_x	rx_y	rx_z	Bandwidth	rx_fc	tx_x	tx_y	tx_fc	tx_power	num_ribs
11	12	13	14	15	16	17	18		
num _blades	blade _length	pitch	RPM	altitude	tx_radius	elevation _angle	azimuth _angle		

TABLE A.2: Output file header with simulation parameter order

The output file contains more than just the signal data the first few data entries are data pertaining to the simulation parameters. it is important to separate this data from the signal data before processing the signal. The template for this data is shown in table A.2.

this is all provided in a single string of comma separated values called a .CSV file. MATLAB has a built in function to read CSV data.

```
1 data = csvread('file_path/file_name')
```

after all the data is read into MATLAB the data can be split into two vectors, one pertaining to the physical simulation parameters and the other to the received signal.

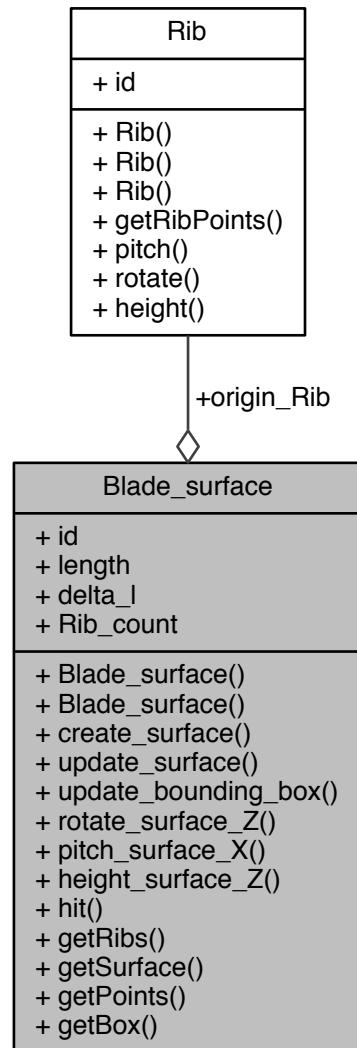
There are several templates for simulation execution in the main.c file that are commented out to help get started, depending on the execution environment the generation of a single file can take anywhere from a few minuets to several days. Under the defaults provided a single file can be generated in approximately 15min.

## **Appendix B**

## **Class Documentation**

## B.1 Blade\_surface Class Reference

Collaboration diagram for Blade\_surface:



## Public Member Functions

- **Blade\_surface** (const int id, const double length, const int Rib\_count, [Rib &origin\\_Rib](#))
- void **create\_surface** ()
- void **update\_surface** ()
- void **update\_bounding\_box** ()
- void **rotate\_surface\_Z** (double angle)
- void **pitch\_surface\_X** (const double angle)
- void **height\_surface\_Z** (const double height)
- bool **hit** (const [Ray3D &ray](#), double &hitDistance, [Vector3D &hitNormal](#), [Point3D &hitPoint](#))
- std::vector< [Rib](#) > **getRibs** ()
- std::vector< [Triangle](#) > **getSurface** ()
- std::vector< [Point3D](#) > **getPoints** ()
- [Bounding\\_box](#) **getBox** ()

## Public Attributes

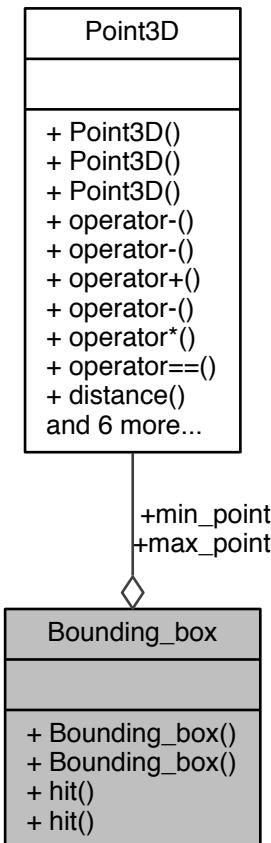
- int **id**
- double **length**
- double **delta\_l**
- int **Rib\_count**
- [Rib](#) **origin\_Rib**

The documentation for this class was generated from the following files:

- Raytracer3D/Raytracer3D/Blade\_surface.hpp
- Raytracer3D/Raytracer3D/Blade\_surface.cpp

## B.2 Bounding\_box Class Reference

Collaboration diagram for Bounding\_box:



### Public Member Functions

- `Bounding_box (std::vector< Point3D > &points)`
- `bool hit (const Ray3D &ray, Point3D &hit_point) const`
- `bool hit (const Ray3D &ray) const`

## Public Attributes

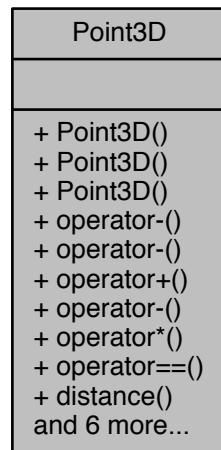
- `Point3D min_point`
- `Point3D max_point`

The documentation for this class was generated from the following files:

- Raytracer3D/Raytracer3D/Bounding\_box.hpp
- Raytracer3D/Raytracer3D/Bounding\_box.cpp

### B.3 Point3D Class Reference

Collaboration diagram for Point3D:



#### Public Member Functions

- **Point3D** (const double x, const double y, const double z)
- **Point3D** (const double y, const double z)
- **Point3D operator-** () const
- **Vector3D operator-** (const [Point3D](#) &p) const
- **Point3D operator+** (const [Vector3D](#) &v) const
- **Point3D operator-** (const [Vector3D](#) &v) const
- **Point3D operator\*** (const double a) const
- bool **operator==** (const [Point3D](#) &p) const
- double **distance** (const [Point3D](#) &p) const
- double **x** () const
- double **y** () const
- double **z** () const

- void **rotate\_Z** (const double angle)
- void **rotate\_X** (const double angle)
- void **translate\_Z** (const double height)

## Friends

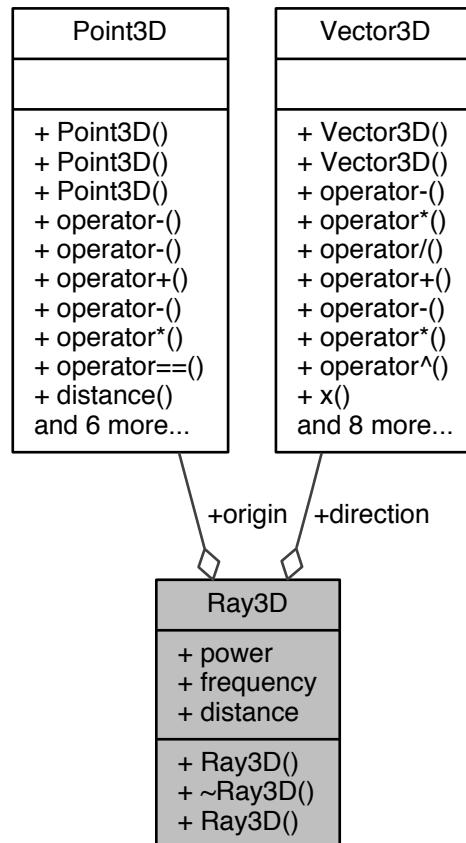
- void **swap** ([Point3D](#) &a, [Point3D](#) &b)

The documentation for this class was generated from the following files:

- Raytracer3D/Raytracer3D/Point3d.hpp
- Raytracer3D/Raytracer3D/Point3d.cpp

## B.4 Ray3D Class Reference

Collaboration diagram for Ray3D:



### Public Member Functions

- **Ray3D** (const [Point3D](#) &origin, const [Vector3D](#) &direction, const double power=1.0, const double frequency=0.0, const double distance=0)

## Public Attributes

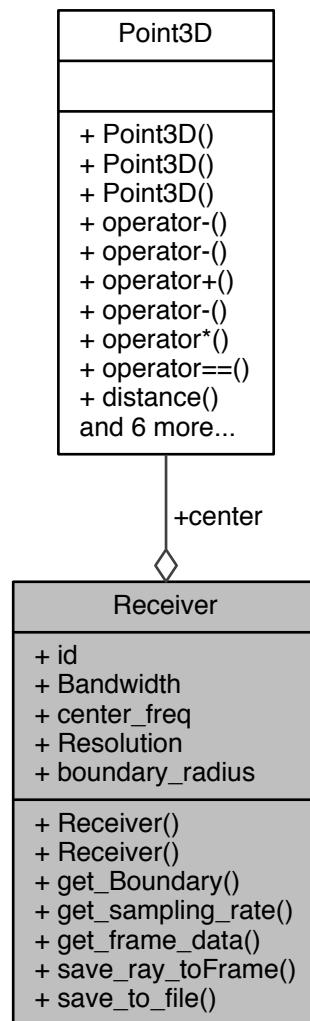
- `Point3D origin`
- `Vector3D direction`
- `double power`
- `double frequency`
- `double distance`

The documentation for this class was generated from the following files:

- Raytracer3D/Raytracer3D/Ray3d.hpp
- Raytracer3D/Raytracer3D/Ray3d.cpp

## B.5 Receiver Class Reference

Collaboration diagram for Receiver:



### Public Member Functions

- **Receiver** (const int id, const double Bandwidth, const double center\_freq, const [Point3D](#) &center, const double boundary\_radius, const std::string &savefile\_name,

- ```
const std::string &dopplerfile_name)
```
- **Sphere get\_Boundary ()**
  - **double get\_sampling\_rate ()**
  - **std::vector< Ray3D > get\_frame\_data ()**
  - **void save\_ray\_toFrame (Ray3D &ray)**
  - **void save\_to\_file ()**

## Public Attributes

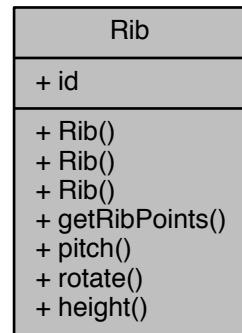
- **int id**
- **double Bandwidth**
- **double center\_freq**
- **double Resolution**
- **double boundary\_radius**
- **Point3D center**

The documentation for this class was generated from the following files:

- Raytracer3D/Raytracer3D/Receiver.hpp
- Raytracer3D/Raytracer3D/Receiver.cpp

## B.6 Rib Class Reference

Collaboration diagram for Rib:



### Public Member Functions

- **Rib** (int id, [Rib](#) &x, const double delta\_l)
- **Rib** (int id, const std::string &filename)
- std::vector< [Point3D](#) > **getRibPoints** ()
- void **pitch** (double angle)
- void **rotate** (double angle)
- void **height** (const double height)

### Public Attributes

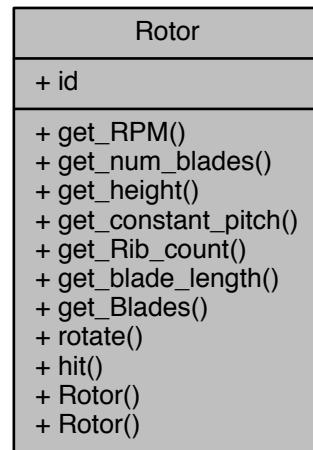
- int **id**

The documentation for this class was generated from the following files:

- Raytracer3D/Raytracer3D/Rib.hpp
- Raytracer3D/Raytracer3D/Rib.cpp

## B.7 Rotor Class Reference

Collaboration diagram for Rotor:



### Public Member Functions

- double **get\_RPM** ()
- int **get\_num\_blades** ()
- double **get\_height** ()
- double **get\_constant\_pitch** ()
- double **get\_Rib\_count** ()
- double **get\_blade\_length** ()
- std::vector< [Blade\\_surface](#) > **get\_Blades** ()
- void **rotate** (const double angle)
- bool **hit** (const [Ray3D](#) &ray, double &hitDistance, [Vector3D](#) &hitNormal, [Point3D](#) &hitPoint)
- **Rotor** (const int id, const int num\_blades, const double RPM, const double height, const double constant\_pitch, const double blade\_length, const int Rib\_count)

## Public Attributes

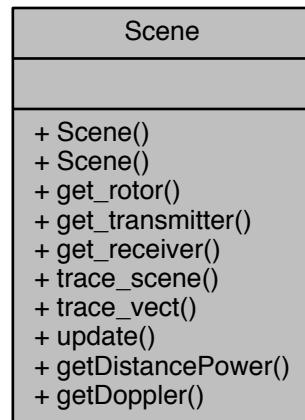
- int **id**

The documentation for this class was generated from the following files:

- Raytracer3D/Raytracer3D/Rotor.hpp
- Raytracer3D/Raytracer3D/Rotor.cpp

## B.8 Scene Class Reference

Collaboration diagram for Scene:



### Public Member Functions

- **Scene** (double rx\_x, double rx\_y, double rx\_z, double Bandwidth, double rx\_fc, double tx\_x, double tx\_y, double tx\_fc, double tx\_power, int num\_blades, double RPM, double altitude, double pitch, double blade\_length, int num\_ribs, const std::string &filename)
- **Rotor** **get\_rotor** ()
- **Transmitter** **get\_transmitter** ()
- **Receiver** **get\_receiver** ()
- void **trace\_scene** (int num\_rays)
- void **trace\_vect** ([Ray3D](#) &test\_ray, double &hitDistance, [Vector3D](#) &hitNormal, [Point3D](#) &hitPoint)
- void **update** (double angle)
- double **getDistancePower** (const double frequency, const double power, const double distance) const

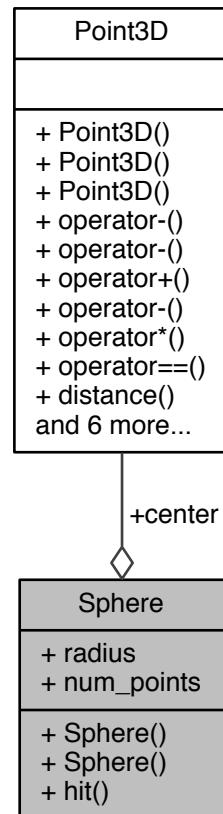
- double getDoppler (Ray3D &test\_ray, Vector3D &hitNormal, Point3D &hitPoint, double RPM) const

The documentation for this class was generated from the following files:

- Raytracer3D/Raytracer3D/Scene.hpp
- Raytracer3D/Raytracer3D/Scene.cpp

## B.9 Sphere Class Reference

Collaboration diagram for Sphere:



### Public Member Functions

- **Sphere** (const double radius, const `Point3D` &center)
- bool **hit** (const `Ray3D` &ray, double &hitDistance, `Vector3D` &hitNormal, `Point3D` &hitPoint) const

## Public Attributes

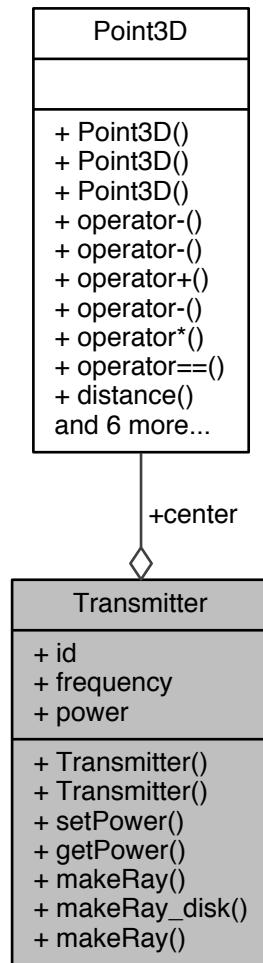
- double **radius**
- [Point3D](#) **center**
- int **num\_points**

The documentation for this class was generated from the following files:

- Raytracer3D/Raytracer3D/Sphere.hpp
- Raytracer3D/Raytracer3D/Sphere.cpp

## B.10 Transmitter Class Reference

Collaboration diagram for Transmitter:



### Public Member Functions

- **Transmitter** (const int id, const double frequency, const double power, const [Point3D](#) &center, const double l)
- void **setPower** (const double power)

- double **getPower** () const
- Ray3D **makeRay** ()
- Ray3D **makeRay\_disk** (const double height)
- Ray3D **makeRay** (const Vector3D &rayDirection)

## Public Attributes

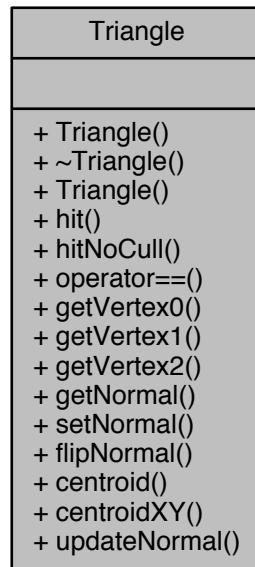
- int **id**
- double **frequency**
- Point3D **center**
- double **power**

The documentation for this class was generated from the following files:

- Raytracer3D/Raytracer3D/Transmitter.hpp
- Raytracer3D/Raytracer3D/Transmitter.cpp

## B.11 Triangle Class Reference

Collaboration diagram for Triangle:



### Public Member Functions

- **Triangle** (`Point3D &v0, Point3D &v1, Point3D &v2`)
- bool **hit** (const `Ray3D` &ray, double &hitDistance, `Vector3D` &hitNormal, `Point3D` &hitPoint) const
- bool **hitNoCull** (const `Ray3D` &ray, double &hitDistance, `Vector3D` &hitNormal, `Point3D` &hitPoint) const
- bool **operator==** (`Triangle` &Tri)
- `Point3D` **getVertex0** ()
- `Point3D` **getVertex1** ()
- `Point3D` **getVertex2** ()
- `Vector3D` **getNormal** () const

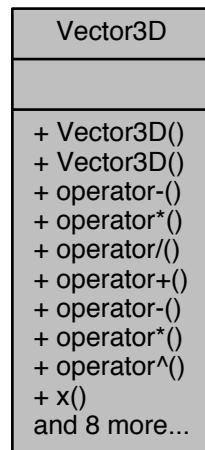
- void **setNormal** (const [Vector3D](#) &normal)
- void **flipNormal** ()
- [Point3D](#) **centroid** () const
- [Point3D](#) **centroidXY** () const
- void **updateNormal** ()

The documentation for this class was generated from the following files:

- Raytracer3D/Raytracer3D/Triangle.hpp
- Raytracer3D/Raytracer3D/Triangle.cpp

## B.12 Vector3D Class Reference

Collaboration diagram for Vector3D:



### Public Member Functions

- **Vector3D** (const double x, const double y, const double z)
- **Vector3D operator-** (void) const
- **Vector3D operator\*** (const double a) const
- **Vector3D operator/** (const double a) const
- **Vector3D operator+** (const [Vector3D](#) &v) const
- **Vector3D operator-** (const [Vector3D](#) &v) const
- double **operator\*** (const [Vector3D](#) &b) const
- **Vector3D operator^** (const [Vector3D](#) &v) const
- double **x** () const
- double **y** () const
- double **z** () const
- double **dotProduct** (const [Vector3D](#) &v) const

- `Vector3D crossProduct (const Vector3D &v) const`
- `double length () const`
- `Vector3D normalized () const`
- `Vector3D rotatedAboutZ (const double angle) const`
- `bool isNormal () const`

The documentation for this class was generated from the following files:

- Raytracer3D/Raytracer3D/Vector3d.hpp
- Raytracer3D/Raytracer3D/Vector3d.cpp