VERTICAL AXIS WIND TURBINES AND BIRDS

Calculating a Theoretical Collision Risk Assuming No Avoiding Action

Programmed and written by, Matthew Wylder, Wind Harvest International
Edited and supervised by Kevin Wolf, VAWT Wind Harvest International. (kwolf@windharvest.com)
DRAFT 5 Aug 2014

Nomenclature

WHI – Wind Harvest International SNH – Scottish Natural Heritage HAWT – Horizontal Axis Wind Turbine VAWT – Vertical Axis Wind Turbine N – Number of Turbines in Farm	D – Diameter of Turbine H– Height of Turbine A_{RW} – Area of Risk Window (see definition in Section E) A_R – Swept Area of Rotors V_{RW} – Volume of Risk Window V_R – Swept Volume of Rotors

A. Overview

The purpose of this document is to compare the bird collision analysis method and equation employed by Scottish Natural Heritage (SNH) for horizontal axis wind turbines (HAWTs) with those that were derived from a similar method for vertical axis wind turbines (VAWTs). WHI repurposed SNH's equations, approximations, assumptions and techniques, which SNH described for determining the frequency of bird strikes for HAWTs. Because VAWTs are three-dimensional while HAWT rotors are essentially two-dimensional, the equations needed to determine the collision rate for birds with VAWTs are significantly different. The VAWT's circular motion is in the same plane as the birds motion, and the HAWTs motion is in a separate plane, the equations needed to determine the collision rate for birds with VAWTs are significantly different.

In each section, SNH's modeling decisions will be presented, followed by the WHI model, making it easier to observe where the methods differ, and where they remain the same. Thus, the organization of this document will be very similar to the "Guidance" paper prepared by SNH [*Windfarms and Birds: Calculating a Theoretical Collision Risk Assuming No Avoiding Action*. 1].

After discussing how the VAWT model is set and run, this paper does a short review of Avoidance and Non-Avoidance of birds around turbines and the "risk window." *This VAWT collision model though does not account for any avoidance of birds, numbers of birds in the area or flight paths around the turbines*. It assumes a bird has entered the VAWT and does a probability-based calculation on the percentage chance it will be struck by the blades or shaft.

B. Model Components and Set Up

The implementation of the assumptions has been streamlined for modification and use by other designers. The bulk of the calculations are executed using Python, but the user will control the variables and execution from within a spreadsheet. WHI chose to use LibreOffice to support the spreadsheet, because its open source nature allows for free access by anyone. Python is also an open source coding language, which comes standard on many computers. It is relatively easy to set this up if you have the correct Windows or Mac operating systems. ***(Note: for short, step-by-step directions for setting up and running the calculator, see Appendix 4.)***

LibreOffice

Installation of LibreOffice is relatively simple. When a computer accesses their download page, the website senses what operating system is used, and prepares the appropriate version of the software for download. If this doesn't work, then there is a list of supported operating systems. The link is:
http://www.libreoffice.org/download/libreoffice-fresh/

Once downloaded, open with the installer and follow the on screen instructions.

The user inputted data is handled and the python code is executed by a macro inside of LibreOffice. In order for these macro controls to work, the user must lower their macro security. To do this, go to LibreOffice

>> Preferences and under the LibreOffice list on the left sidebar select “Security”, and then “Macro Security...” for now, the security must be set to low in order for the controls to work.

Python

While most computers have a version of Python, it is important to make sure that the user has the correct version. The supported operating systems are listed, with the corresponding versions of Python next to it. Python supports reverse compatibility, which means that if the user has a version which is newer than 2.7, then it will work with the given code. Please download and install:
Python 2.7.7: https://www.python.org/download/releases/2.7.7/
Once Python has downloaded, open it with the installer and follow the on screen instructions.

Running the VAWT Bird Collision Equation

Note: Currently this version of the collision model is only available for Mac and Windows users. After downloading OfficeLibre and opening the .ods file, there is a cell at L6 in which the user needs to declare their Mac or Win operating systems. This program may not work in Windows 7 or 8 or Linux. Please let us know if you trouble operating this program in any operating system.

The bulk of the work is done by the Python code, however most users will not find it necessary to open the Python file. The “birdstrikefrontend.ods” document (called “Front End” here) is effectively a controller for the Python execution.

The user can enter their information in the provided locations and when ready, click the button that says “Click Here for Bird Strike Test”. At this point they will be prompted to confirm that the information is correct. Then, the front end changes the status to “Process Running” and hands off the work to Python.

For the majority of the time, the user will leave the Front End in this state, as Python executes its calculations. For an update, the user can click the “Update Progress Bar” button to determine how far along the code is. This button also prepares a time estimate for the expected completion. It does this linearly, which means it determines the average rate of completion per minute and extrapolates out. Additionally, if your computer is running other processes, they will take processing power from Python, slowing the process. Therefore, running other programs during the initial runtime may increase the expected completion time considerably. For better accuracy leave the computer running for several minutes with no other processes.

The calculator will not work if any of the enclosing files have a space in the description. If need be, move the files into your home directory (usually named after the user). As soon as the Python begins, it will create several things. First is a file called test.txt in the home directory, which is where any error reports will be printed, in the event of a failure. Save this file for evaluation. Second, is a folder named diagnostics, where Python will maintain a file which tracks progress in a folder specific to this set of trials. This second folder contains text files which describe the outputs for each stage of the process. Also, if the user tells the front-end, by setting the “Output” value to True in the “Advanced Options” section, there will be a folder labeled “Points”, where the centers of the objects are recorded as ordered pairs for each run.

When all the tests have been completed, Python will place the results into a .csv file which is readable by either excel or LibreOffice. Ensure that the encoding for reading the file is set to UTF-8, also known as Unicode. This file organizes the percentage of strikes by the bird size, blade/bird speed and test conditions. All values are prepared as percentages.

If at any point the user wants to cancel the test, there is a button labeled “Cancel” in the front-end document, which will end the Python process. If the user attempts to start a new process, they will be prompted to end the previous test, or else a new test will not be started. Note that logging out will also cancel Python.

C. Avoidance

The basic principle of avoidance, which is outlined in the SNH paper [1], is that in reality many birds will notice wind turbines ahead of time and avoid them, or else take last second evasive maneuvers if they notice an oncoming blade. The effect of turbine avoidance can be quantified as an “avoidance factor,” but this depends on the site, species and climate [1], and so should be determined by observation. For this reason, both SNH's and WHI's analyses assume that no avoiding action is taken by incoming birds.

D. No-Avoidance Collision Risk

The technique can be summarized by multiplying the number of birds flying through the rotors with the percentage chance that they will be struck. The result will be an estimate for the number of bird strikes per year:

*Strikes per Year = (Bird Flying Into Risk Area) * (%Birds that fly through turbines)*(%Likelihood of strike)*

The method for determining the number of birds which fly through the rotors is the same as the method used by SNH. While the flight patterns of species vary significantly, it is useful to classify their habits to estimate the risk for a bird flying into a turbine. Depending on the bird species, their flight path can be quantified in two ways:

> *Case 1: Regular Flights* – Flight path with predictable directionality and frequency (i.e. migrating geese traveling across a farm along their way from roost to feeding ground, or divers moving from nest sites to the coast.)
>
> C*ase 2: Sharing Airspace* – Roaming birds, like a raptor, which may have a regular territory, with some distribution of time spent at different heights.

E. Birds Flying Through Rotors

Both cases above follow a general formula. Determine, a large-scale region, which can be considered a portion of the bird's regular flight paths. Use this to estimate the number of birds that will even be in the area during the course of a year. This is when alternate routes, flight elevation and species density determine a rough estimate for the number of birds that will be at risk over a certain time span (usually 1 year). The following calculations are shown graphically in Appendix 1

Case 1: Regular Flights

When the flight direction is relatively uniform, the wind farm can be viewed from the oncoming path of the birds as a 'risk window'. The "risk window" is defined as the width and height of the bird observation area perpendicular to of the farm perpendicular to the flight path of the bird, times the height of the wind farm.[1] So; A_{RW} = Width of Farm x Height of Rotors. Both WHI and SNH calculate this in the same way although WHI uses the distance from the lowest bottom of a blade, to the highest top of a blade.

Then, the area presented by the turbines covers the possible paths where birds would pass in between or be struck by the moving blades. SNH assumes that all turbines are facing the bird path, This is because they assume that the decreased risk in passing through the turbines due to their smaller visible section, is offset by the increase risk of a strike due to longer transit time through the strike zone of the rotor.

This is not a factor for VAWTs for case 1, since each individual VAWT presents the same profile from all directions. For both methods, you determine the percentage of birds, which pass through turbines as the amount of paths that are blocked by turbines divided by the total space which is investigated.

A_{RW}=*Width of Farm x Height of Rotors*	-For Both VAWTs and HAWTs
$A_R = N \times (\pi \times R^2)$	-For HAWTs
$A_R = N \times D \times H$	-For VAWTs
% Birds that fly through turbines = A_R/A_{RW}	-For Both VAWTs and HAWTs

Note that overlapping areas are counted multiple times, as this flight path results in higher risk for a strike.

Case 2: Sharing Airspace

In the case of a bird that shares the airspace with the turbines, the percentage of birds which come within the risk volume is determined by dividing volumes. The method is extremely similar to that from case one, however now areas are replaced by volumes. Therefore:

V_{RW} *= Width of Farm x Height of Rotors x Depth of Farm*	-For Both VAWTs and HAWTs
V_R *= N x πR^2 x Average Blade Chord Size over length of blade*	-For HAWTs
V_R *= N x piR^2 x Blade Height*	-For VAWTs
% Birds that fly through turbines = V_R/V_{RW}	-For Both VAWTs and HAWTs

F. Probability that Bird Flying Through Rotor Will Be Struck

When determining whether a bird that passes through a turbine's rotor will be struck, it is a matter of investigating all possible bird paths through the blades. For a HAWT, the blade speed is always taken to be perpendicular to the bird's flight path, and given a constant rate of rotation. This results in a continuous, predictable path, with only one instance at which the bird may intersect with the blades.

However, for VAWTs, the blade path moves circularly through three dimension, and so at some times the blade may be moving away from or towards the bird. This means that the blades relative paths vary with the path of the bird, across the width of the bird.

The SNH method uses the geometry of the bird, and blade to create a purely mathematical probability of a strike. By approximating the blade speed as constant, and defining the probability as a function of the distance of the bird from the center of the rotor, they are able to determine the probability by integrating along the blade. The definition of the probability investigates possible bird paths relative to the blade, and determines the likelihood of a strike. [1]

For a VAWT, the blade motion is more complex, so the same equations will not apply. Rather than derive a mathematical solution, WHI opted for the approach of computationally simulating all possible flight paths. By varying the position from which the bird approaches, it is possible to investigate the entire range of possibilities.

Both methods approximate the bird as a cross shape, with a vertical line representing a body, and a horizontal line at the halfway point representing the wings. SNH uses a 1-D line to represent the blade path, but since the strike can occur from a variety of angles in a VAWT, WHI uses a 2-D cross, like the bird, with the length and thickness of the airfoil included.

Any intersection of the lines, which represent the bird with the lines that represent the blade is a strike. The SNH method investigates a mathematical definition that directly determines the strike probability, but WHI uses a finite test, with a high resolution. The effect is less mathematically eloquent, and requires longer computation time; but provides a relatively accurate estimate for strike rates.

A graphical representation of the calculation technique is included in Appendix 2. This shows the result of a test for a set of birds, with each blade in a different color. The WHI test creates a series of ordered pairs that represent a finite list of locations that the blade will pass, as well as a series of ordered pairs for each bird for each unique path through the blades.

As the VAWT blades rotates, the distance between the centers of blades and birds is calculated. If it falls within a certain range, then there is a collision test for the blade and bird. The test for a collision and assumptions are called a "cross product test", which is a well-understood mathematical test. It relies on determining if the points of line A, are on either side of line B, and vice versa. The full details of the algorithm are outlined in Appendix 3.

To summarize, the SNH paper uses a direct mathematical calculation, while WHI uses a series of direct tests to get an estimate for the percentage of birds that will be struck if they fly through VAWT rotors. Both methods have inherent strengths and weaknesses; the SNH method is more accurate and fast, but not directly transferrable to a VAWT, while the WHI method takes longer, it is easily understood and prepared. The WHI model does add numerical inaccuracies, but these are small relative to the inaccuracies that are introduced by the following assumptions and approximations.

G. Assumptions and Approximations

Both papers make a series of assumptions in order to simplify the problem into one that can be solved with relatively simple math. The first assumption is that there is no avoiding behavior, because this can be determined from future work and be inserted as a factor to the theoretical value. The second is that near misses are not counted as strikes, while in actuality, the wake of the blade is likely to disturb the bird's flight path, or potentially cause them harm. Also, bird and blade speeds are taken to be constant, which will not always be the case, as wind speed has a direct effect on the speed of the bird and of the blades of turbines with variable speed generators.

Also, the SNH paper assumes that birds, which approach the blades at an oblique angle face approximately the same strike percentage as those which approach dead on. SNH assumes that the decrease in risk due to a smaller risk window, is offset by the increase in risk due to a longer time to transverse the blades. This doesn't apply to birds that approach the same geometric configuration regardless of their approach angle; however it may be possible for WHI to apply this approximation to birds, which may pass through the rotors at a vertical angle (as in diving birds).

Finally, the SNH paper approximates the blade to be a one dimensional line while WHI uses both the blade thickness and its chord size. However, the current VAWT analysis doesn't account for bird strikes against the arms that support the blades and the cables that support the arms. These obstacles will likely increase the percentage of strikes. The current analysis only looks at a purely two dimensional path of the bird coming towards a set of blades and a central shaft upwards through the plane, with no other surfaces which it can strike.

Due to the above limitations of their model, SNH suggests that the results may be inaccurate to a tune of around 10%. WHI's VAWT analysis is susceptible to the same amount of error in addition to the numerical error of doing a finite calculation. In addition, there will be some numerical error in the results matrix where the trends in probability of collision are not completely accurate. This error results because this program is not a direct mathematical calculation but uses a finite approximation and without a regression analysis based on test resolution.

Note on Results

H. Model Dependencies

The program designed by WHI can be used for a range of bird and turbine sizes, so the following list of attributes effects the outcome of the testing. Since some components, such as blade rotation rate and bird speed, are not set permanently, a range of values will be tested. The model currently only considers the motion of the blades, and the center hub, and there are future additions listed at the end. There is also a range of units for most of the variables that can be selected.

Supported Dependents		**Supported Units**			
Bird Width		Feet	Inches	Meters	
Bird Length					
Blade Diameter		Feet	Inches	Meters	
Center Hub Diameter					
Chord Length					
Blade Thickness		Feet	Inches	Meters	% of Chord Length
Blade Thickness Location					
Low/High and Interval	Bird Speed	Meters per Second	Feet per Second		
	Blade Speed	RPM	Meters per Second	Feet per Second	

The user must also declare the number of blades in the VAWT, which are presumed to be equally spaced from one another, and whether or not the user wants to output the data that is generated in the code. The connection of these dependents is outlined in a pseudo-code in Appendix 3, that includes the process of computation, without the explicit Python language.

Each of these inputs has a different effect on the strike potential: a larger bird, a tighter blade path, larger center hub, thicker blades, higher blade speed or lower bird speed all result in a higher chance of bird strike, and vice versa.

Future calculations will include dependencies that take into account other turbine features, including the addition of the arms that connect the blades to the hub, and supporting wires that strengthen the blades against gravity and other forces. Since these components vary along the vertical axis, the test will have a higher number of computations.

Future Dependencies:

The actual number collisions will be affected by the following additional VAWT properties that are currently not accounted for in this collision model.

- Location and angle of reinforcing cables
- Number and thickness of arms which support blades
- Cross sections along the total height of blades. Presently this model only accounts for one cross section.
- Thickness of bird and number of tests in vertical direction

Feature Expansion:

This collision model would be benefit from the following additional features.

- Include bird flight characteristics and wind farm size into calculations
- Allow for inputs of more bird sizes
- Outsource computation to faster computers

The full Python code is available in a separate document. The Python "pseudo code" is available in Appendix 3.

I. Sources

[1]Scotland. Scottish Natural Heritage. *Windfarms and Birds: Calculating a Theoretical Collision Risk Assuming No Avoiding Action*. National Renewable Energy Labratory, 1 Jan. 2000. Web. 14 May 2014. <http://nrelpubs.nrel.gov/Webtop/ws/avianlt/www/web_data/Record;jsessionid=08C037846844D69ABF06974083C71974?m=1320>.

[2]Prosser, Patrick. *Geometric Algorithms*. *Algorithms and Data Structure*. Glasgow University/ School of Computing Science, 30 Mar. 2000. Web. 3 July 2014. <http://www.dcs.gla.ac.uk/~pat/52233/slides/Geometry1x1.pdf>.

Appendix 1

Figures 1 and 2 – Images from SNH Paper to which VAWT Imagery has been inserted. Figure 1 shows Case 1 and Figure 2 shows Case 2 *Note:* The lines represent flight paths for birds.

Appendix 2 – Not-To-Scale Illustration of WHI Algorithm for VAWTs

Notes:

1. Each of the three blades is represented by the blue, red or green, and the black represents both the arms and the central hub. The dark black arms is the initial location of the blades.

2. Rather than simulate motion of the bird and blades, the blades move relative to the bird, so the hub appears to be moving towards the bird as the blades rotate. This allows for a single calculation of bird position, and many calculations of blade position. Each bird is counted as a strike if any part of it intersects with any part of the turbine.

3. Since this is only a crude representation, we see that each bird is struck, and that the possible blade locations are not fully explored. The test is complete only when all of the blades are completely beyond the locations of the birds. Additionally, the blade is represented as rotating more slowly than the bird is flying, and the birds and blades are relatively large. More detailed modeling is a possibility.

Appendix 3 – Pseudo-Code

1. Collect the Inputs for calculation dependencies and Adjust Units to SI, RPM and %chord.
2. Create a list of ratios of bird speed to rotation rate which is used for calculations: Degrees/Meter
3. Connect each set of test conditions (bird speed/rotation rate) to a value of Degrees/Meter
4. Each cycle is the amount of rotation to return to the initial condition: chosen 3
5. Number of birds per row and degree step size is declared by the user
6. Create a lists of ordered pairs for both blade and bird positions, all motion is represented from the birds reference frame. So, effectively the birds remain still, while the blades spin and move toward the bird. This is why the motion can be described as a ratio of rotational to linear displacement.
7. **Bird Positions:**
    - **a)** Equally space birds in a row placed just outside of the initial blade position, initially as center points
    - **b)** After each degree step (0.25 degrees) place next bird row the same distance outside of the turbine

      Explanation: If for each degree step the blades move 'x' meters closer to a row, the following row is placed x meters behind, this ensures that bird rows are uniformly distributed within the blade path, without a higher concentration of birds in any given location, and that the rows are in sets of unique conditions.
    - **c)** After each cycle, the bird rows will face the same conditions as the row in the previous cycle

      Explanation: The first row in the first cycle will have the same strike pattern as the first row in the last cycle, multiple cycles serves to increase the accuracy of the tests.
    - **d)** Depending on the size of the bird, the full representation of the wings and body is calculated.
8. **Blade Positions:**
    - **a)** Ensure that the position of the blades are calculated for enough rotations such that the last row of birds has fully cleared the path.
    - **b)** Create a set of ordered pairs for the blades end points, and the center of the hub.
9. There is now an option to output a list of the ordered pairs of the centers to a file labeled after the conditions of the test. If this is done, the file is located under Diagnostics/TestOn[TheDate+Time]/Points in several files. Each labeled either, birds, hub or blade number.
10. Go through each bird and each blade position, if the bird falls within the hub, or if the lines which represent the bird and blade overlap, then the bird is counted as a hit. Also, there is a line traced between the points on each blade to the same corresponding points, on the previous blade.
    - **a)** Specifics: First the distance between the centers is calculated, if the distance is less than the minimum acceptable distance then an overlap test of all the lines is calculated. The minimum acceptable distance is the largest of the following: bird width, bird length, chord length, hub diameter.
    - **b)** The test of an overlap utilizes a mathematical property called the cross product, where the sign of the output follows the right hand rule. Given segment AB and CD it determines if points C and D are on either side of segment AB and if points A and B are on either side of segment CD. If this is true, then the segments overlap. In the special case of parallel segments, there is a further test to determine overlap. [2]
11. Take the number of hits and divide it by the total number of birds, this is the chance of a hit for a given bird size moving at a given Degrees/Meter.

12. Pair the Degrees/Meter results with their corresponding pair of bird speed and rotation speed, then results are written to an output file. In the Diagnostics folder located next to the BirdStrikeFrontEnd.ods file.

13. After all results are prepared, they are compiled into a .csv file in the same folder as BirdStrikeFronEnd.ods file.

The exact code for both the Libre Office Macro (Visual Basic) and the Python execution are available upon request.

Appendix 4

Quick Directions for Using the VAWT Bird Strike Calculator
DRAFT - M. Wylder
5 August 2014

1. Install LibreOffice – Go to http://www.libreoffice.org/download/libreoffice-fresh/ and check above the "Download Version" button to make sure that LibreOffice has correctly identified your operating system. Then select the "Download Version" button.

2. Open the downloaded file once and follow the installation directions.

3. Update Python – All computers come with some Python, make sure that yours is up to date. Go to https://www.python.org/downloads/ and make sure that the title of the page says "Download the latest version for [Your OS]" to make sure that Python has correctly identified your operating system. Then, select "Download Python 2.7.8" to get the update.

4. Open the file once downloaded and follow the installation directions.

5. Now you have LibreOffice and an up to date version of Python. Examine the Bird Strike Calculator files provided by WHI, there should be the following:
   - ***BirdStrikeFrontEnd.ods*** – This is the LibreOffice spreadsheet where you will enter the information for the trials. It is called the "Front End" because it doesn't execute any calculations, it is simply where you hand data off to Python. The data which is handed off will be stored in a file in the same folder named "Variables.csv" which will appear the first time that you run a test.
   - ***ExecutePython.py*** – This is the Python file which does all of the math and logical operations to determine an estimate for bird strike percentages. When BirdStrikeFrontEnd.ods creates the "Variables.csv" file Python takes those values and prepares a folder named "Diagnostics" where each trial has a results folder called "TestOn[Date/Time]." In this folder there are files containing the results and ordered pairs for some key points for each described condition. NOTE: Completed trials are compiled in a file named after the bird species with a number on the end. This can be opened with any spreadsheet reader.
   - ***BirdStrike.cmd.txt*** – Mac users: ignore, LibreOffice has enough control to initiate Python. Windows users: This file is responsible for calling the Python. LibreOffice is less effective at controlling a Windows system, so this file is called to start the process. Many email providers don't allow a .cmd file to be attached, and so it adds a .txt ending.
     NOTE: To prepare the file for use, rename it by either double clicking the name, or opening the "info" option and changing the name to: "BirdStrike.cmd"

6. Python will not run properly if any of the enclosing folders have spaces in the name. This means that if the file is in a folder like: "C:\Matt\Desktop\WHI\Bird Strike" it will not work, however "C:\Matt\Desktop\WHI\BirdStrike" would be just fine.

7. Open BirdStrikeFrontEnd.ods with LibreOffice, and fill in the details of the test which you would like to run. The user must specify whether they want to include a list of ordered pairs to describe test positions, or if they want to run the test with slight variation to show the variation caused by numerical error. Also, the test resolution must be specified, a higher test resolution will result in longer run-times but a more accurate result. The following describe the resolutions:
   - Low – 100 birds, 1 deg step; Good for a quick test(10-15 mins) reasonable errors
   - Med – 250 birds, 0.5 deg step; Longer test(45-75 mins) reduced errors
   - High – 500 birds, 0.25 deg step; Very Long Test(2 hrs+) negligible numerical errors

   If the user is logged out at any time then the test will be cancled and cannot be resumed

8. Begin the test by selecting the “Click Here for Bird Strike Test” and verifying that the inputs are ready. Once the test is running, it is possible to test the progress after some time by selecting the “Update Progress Bar” button at the bottom of the page. And the test can be canceled with the “Cancel” button. Remember that running other applications while doing a test will increase the run-time.