Investigating the Application of Flettner Rotors on the Great Lakes

Garrett C. Bond
University of New Orleans

Follow this and additional works at: https://scholarworks.uno.edu/honors_theses
Part of the Ocean Engineering Commons

Recommended Citation
https://scholarworks.uno.edu/honors_theses/140

This Honors Thesis-Unrestricted is protected by copyright and/or related rights. It has been brought to you by ScholarWorks@UNO with permission from the rights-holder(s). You are free to use this Honors Thesis-Unrestricted in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you need to obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself.

This Honors Thesis-Unrestricted has been accepted for inclusion in Senior Honors Theses by an authorized administrator of ScholarWorks@UNO. For more information, please contact scholarworks@uno.edu.
Investigating the Application of Flettner Rotors on the Great Lakes

An Honors Thesis Presented to the

Boysie Bollinger School of Naval Architecture and Marine Engineering

at the University of New Orleans

In Partial Fulfillment

of the Requirements for the Degree of

Bachelor of Science with University Honors

in Naval Architecture and Marine Engineering

By:

Garrett C. Bond

May 2022
# Table of Contents

List of Figures .................................................................................................................. iv  
List of Tables .................................................................................................................... vi  
Abstract .............................................................................................................................. viii  
Introduction and Background .......................................................................................... 1  
  Motivations .................................................................................................................... 1  
  Modern Methods .......................................................................................................... 2  
  Flettner Rotor Technology ............................................................................................. 3  
  The Great Lakes ............................................................................................................ 8  
Theory ............................................................................................................................... 9  
  True and Apparent Wind .............................................................................................. 9  
  Magnus Effect .............................................................................................................. 10  
  Lift and Drag Forces ................................................................................................... 11  
  Spin Ratio .................................................................................................................... 13  
  Aspect Ratio and Thom Disk ....................................................................................... 14  
Method .............................................................................................................................. 15  
  Coefficient Selection .................................................................................................. 15  
  Weather Data .............................................................................................................. 17  
  Route Planning ........................................................................................................... 19  
Hollenbach’s Method ......................................................................................................... 24
Ship Profile .............................................................. 26
Calculations..................................................................... 29
Results............................................................................. 33
Lake Michigan .............................................................. 35
Lake Superior .................................................................. 38
Lake Huron ..................................................................... 40
Lake Erie ........................................................................ 42
Lake Ontario ................................................................... 43
Conclusion ....................................................................... 46
Summary ........................................................................ 46
Future Work .................................................................... 46
References ....................................................................... 48
Appendix .......................................................................... 50
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>UT Wind Challenger</td>
<td>2</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Airbus SeaWing</td>
<td>3</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Flettner Rotor Diagram</td>
<td>4</td>
</tr>
<tr>
<td>Figure 4</td>
<td>The Buckau</td>
<td>4</td>
</tr>
<tr>
<td>Figure 5</td>
<td>The Barbara</td>
<td>5</td>
</tr>
<tr>
<td>Figure 6</td>
<td>E-Ship 1</td>
<td>6</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Viking Grace</td>
<td>7</td>
</tr>
<tr>
<td>Figure 8</td>
<td>SC Connector</td>
<td>7</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Key Great Lake Connections</td>
<td>8</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Maritime Transportation System Shipping Density</td>
<td>9</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Apparent Wind Vector Diagram</td>
<td>10</td>
</tr>
<tr>
<td>Figure 12</td>
<td>The Magnus Effect</td>
<td>11</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Drag Scenarios</td>
<td>13</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Rotor Configurations</td>
<td>15</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Craft CFD Setup</td>
<td>16</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Craft Lift Coefficient Results</td>
<td>17</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Lake Michigan Routes</td>
<td>20</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Lake Superior Routes</td>
<td>21</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Lake Huron Routes</td>
<td>22</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Lake Erie Routes</td>
<td>23</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Lake Ontario Routes</td>
<td>24</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Algoma Equinox</td>
<td>27</td>
</tr>
</tbody>
</table>
Figure 23. Algoma Equinox Rose Plot ................................................................. 34
Figure 24. Lake Michigan Power Contributions .................................................. 37
Figure 25. Lake Superior Power Contributions .................................................... 39
Figure 26. Lake Huron Power Contributions ....................................................... 41
Figure 27. Lake Erie Power Contributions ......................................................... 43
Figure 28. Lake Ontario Power Contributions .................................................... 45
List of Tables

Table 1. Hollenbach's Applicability Limits ................................................................. 25
Table 2. Hollenbach's Required Inputs ........................................................................ 25
Table 3. Algoma Equinox Particulars ......................................................................... 28
Table 4. Specific Energy Ranges for MDO ................................................................. 32
Table 5. Input Values .................................................................................................. 33
Table 6. Lake Michigan - Route 1 .............................................................................. 35
Table 7. Lake Michigan - Route 2 .............................................................................. 35
Table 8. Lake Michigan - Route 3 .............................................................................. 35
Table 9. Lake Michigan - Route 4 .............................................................................. 36
Table 10. Lake Michigan - Route 5 ............................................................................ 36
Table 11. Lake Michigan - Mean Results by Route ................................................... 37
Table 12. Lake Superior - Route 1 .............................................................................. 38
Table 13. Lake Superior - Route 2 .............................................................................. 38
Table 14. Lake Superior - Route 3 .............................................................................. 38
Table 15. Lake Superior - Mean Results by Route ................................................... 39
Table 16. Lake Huron - Route 1 ................................................................................. 40
Table 17. Lake Huron - Route 2 ................................................................................. 40
Table 18. Lake Huron - Route 3 ................................................................................. 40
Table 19. Lake Huron - Mean Results by Route ....................................................... 41
Table 20. Lake Erie - Route 1 .................................................................................... 42
Table 21. Lake Erie - Route 2 .................................................................................... 42
Table 22. Lake Huron - Route 3 ................................................................................. 42
Table 23. Lake Erie - Mean Results by Route ................................................................. 43
Table 24. Lake Ontario - Route 1 .................................................................................. 43
Table 25. Lake Ontario - Route 2 .................................................................................. 44
Table 26. Lake Ontario - Route 3 .................................................................................. 44
Table 27. Lake Ontario - Mean Results by Route .......................................................... 45
Abstract

Climate change is quickly becoming one of the leading problems of the twenty-first century, and the maritime transportation sector is beginning to react. Wind propulsion technologies have been making a comeback led by the Flettner rotor. The deck-mounted, vertical, spinning cylinder generates a lift (or thrust) force perpendicular to the wind by utilizing the Magnus effect. Rotor performance has been studied extensively on many ocean shipping routes, but has not been on the Great Lakes, a region with a large maritime industry. This study aims to assess whether or not Flettner rotors are practical in the region by calculating propulsive power contributions on several popular routes. Results show that a single rotor can provide an average of 2.69% of a ship’s propulsive needs, demonstrating that, while still beneficial, Flettner rotors are less efficient on the smaller bodies of water when compared to ocean based results. However, rotor sails are still capable of providing a significant boost to efficiency in a multi-rotor system deployed on a favorable route.

Keywords: Flettner rotor, Great Lakes, wind propulsion, shipping, climate change


**Introduction and Background**

**Motivations**

The maritime shipping industry faces a major crisis in the form of tightening global pollution standards as they begin to shape the industry in the early part of the 21\textsuperscript{st} century. A report released in 2017 by the International Council on Clean Transportation (ICCT) estimated that international shipping accounted for 3\% of all global greenhouse gas pollution, emitting nearly one-billion tons of CO\textsubscript{2} annually. The report also states that, unchecked by stricter regulations, shipping emissions will increase by between 20\% and 250\% as demand increases, potentially accounting for as much as 17\% of global pollution by 2050 [1].

In response, various regulatory commissions have begun to enact emission standards and set goals for reducing the harmful impact of maritime transportation. In 2018, following the Paris Climate Accord, the International Maritime Organization (IMO) implemented a three-step plan to reduce pollution. Step one is to improve the energy efficiency design requirements for all new IMO registered vessels, the second is to reduce emissions per transport by 40\% by 2030 and 70\% by 2050, and the final step is to reduce total CO\textsubscript{2} emissions by 50\% by 2050 (using 2008 as the reference point) [2]. The European Union’s Maritime Safety Agency (EMSA) has also set goals to take greenhouse gas emissions below 40\% of 1990 levels by 2030 and to move towards carbon neutral by 2050 (this was done in accordance with the European Green Deal) [3]. As these regulations become increasingly normal, ship designers and operators are beginning to explore innovative methods to achieve these targets.
Modern Methods

Two of the most popular ideas are to reduce the overall resistance of the ship or to supplement the typical combustion propulsors with clean energy alternatives. Both techniques aim to reduce the fuel consumption of a vessel and therefore reduce the amount of CO$_2$ released per trip. Methods of resistance reduction include hull lubrication (introducing air bubbles between the hull and water), hydrophobic coatings, and new hull form optimization techniques [4]. Supplemental propulsion concepts have returned to an age-old source of power at sea: the wind.

Today’s attempts to harness the wind for ship propulsion take on several different forms, all quite different from those found on traditional sailing vessels. Rigid, deck-mounted wing sails that model vertical airplane wings have been found to be quite effective in optimal conditions but take up a considerable amount of deck space and have been found to be less effective at higher vessel speeds. An example of a concept vessel outfitted with wing sails can be seen below in Figure 1.

![Figure 1. UT Wind Challenger](image)

Another, more conventional sail being studied is a kite sail. Here, a sail is deployed into the air forward of the vessel (like a kite) and helps pull the ship forward. This can result in enormous
power contributions when the wind is at the ships back but is not favorable in most other wind
directions and must be deployed and recovered for every use. Below is an Airbus concept of an
autonomously deployed and recovered wing sail.

![Airbus SeaWing](image)

_Figure 2. Airbus SeaWing_

One more wind propulsion idea getting modern attention, and the focus of this thesis, is Flettner
Rotor technology, which will be discussed further in the following section [5].

**Flettner Rotor Technology**

Flettner rotors are vertical cylinders mounted to the deck of the vessel that spin around their
longitudinal axis. If a wind perpendicular to the direction of the ship’s motion is introduced, the
spinning rotor produces a lift (or thrust) force in the direction normal to the wind, driving the
vessel forward. Figure 3 below illustrates how the Flettner rotor operates.
Flettner rotors are not a recent development; in fact, the first rotor was patented by German aeronautical engineer Anton Flettner in 1922. Two years later, along with his colleagues Ludwig Prandtl and Albert Bertz, Flettner outfitted two 15-meter tall and 3-meter in diameter steel rotors to a schooner named the Buckau. The rotors were spun at 125 revolutions per minute by two electric motors with power provided by a single diesel generator.

The newly outfitted vessel performed admirably, displaying superior tacking capabilities when compared to conventional sailing vessels. After numerous trials the Buckau was put into
commercial service in 1926 and hauled lumber in the North Sea and Atlantic Ocean for several years. Following the apparent success of the *Buckau*, another vessel, the *Barbara*, was outfitted with three larger rotors and six more vessels were ordered. Unfortunately, with the price of conventional fossil fuels plummeting amid economic turmoil, both vessels (and those ordered) were eventually converted to regular engine-propeller propulsion [6].

![Image of the Barbara](image_url)

*Figure 5. The Barbara*

Feasibility studies of the technology returned as early at the 1980’s, but it took until 2008 for another rotor ship to be conceived. In 2008 Enercon, a German wind sector company, launched *E-Ship 1* outfitted with four Flettner rotors. Each rotor is 25-meters tall and 4-meters in diameter and the company claims that, combined, the rotors cut the fuel cost of the vessel by 30%. In a departure from the normal electric powering of the sails the rotors aboard *E-Ship 1* are powered by steam turbines attached to the exhausts of the main diesel propulsion engines [7].
Even more recently the Finnish company Norsepower has emerged as a pioneer of the Flettner rotor technology. According to their website, Norsepower rotors are modernized versions of the original design constructed of composite materials and autonomously controlled by proprietary software. The software is an important development since it allows the rotors to be turned on and off automatically depending on the favorability of the wind conditions at any given time, boosting overall efficiency. Norsepower sails have been installed on several vessels since the company’s inception in 2012, most notably the tanker Maersk Pelican and ferry Viking Grace (pictured below).
One of the clear disadvantages of Flettner rotors is the substantial addition to the vessel’s air draft. Today’s rotor sails can reach heights of 35-meters above the main deck, which can severely restrict the vessels capabilities to operate in environments with overhead obstructions such as bridges. Luckily, Norsepower has recently developed a solution to this problem. In January 2021, they successfully installed two 35-meter tilting rotor sails aboard the SC Connector and began operation on height restricted routes in the North Sea.
The Great Lakes

The Great Lakes consist of five major freshwater lakes (Lake Michigan, Lake Superior, Lake Huron, Lake Erie, and Lake Ontario) and the connecting waterways between them (namely, the Soo Locks and the Welland Canal). Including the St. Lawrence River, the system represents the largest navigable inland water system in the world, stretching over 2,000 miles into North America.

![Diagram of Great Lakes connections]

**Figure 9. Key Great Lake Connections**

Unsurprisingly, this makes maritime transportation a key industry in the region. Between the 100 U.S. and Canadian ports on the waterways, nearly 200 million tons of cargo is transported every year worth $3.4 billion to the economy. Typical products include iron ore, limestone, coal, sand, salt, cement, grain, and gypsum. Most of the transportation is between inland ports (i.e., iron ore from Michigan’s Upper Peninsula to Buffalo, NY for steel production) but approximately 25% of the transportation are oceangoing vessels traveling to and from overseas ports [8]. Pictured below is a density map of commercial maritime transportation in the region.
It is important to note that the vessels that operate solely on the Great Lakes, often called “Lakers”, have a longer service life than the “Salties” (ocean-going) due to the less corrosive freshwater environment. The typical Great Lakes freighter will be in service for upwards of fifty years, double the average lifespan of their oceangoing counterparts. This is, however, not necessarily good since it means that most of the Lakers in operation today are far less efficient than modern designs and subsequently have a much greater negative impact on the environment. This makes the pursuit of alternative propulsion means an important conversation within the region. Flettner rotors are particularly intriguing solution since they could be easily retrofitted onto nearly any vessel during the winter layup period with no intrusion on the vessel’s operation.

**Theory**

**True and Apparent Wind**

Before the driving principles behind Flettner rotors can be examined and explained further it is important to make a distinction between true wind and apparent wind. True wind is defined as
the wind speed and direction (true wind angle, TWA) environmentally recorded at a certain location. However, since rotors operate on vessels underway, the rotors are subject to the true wind as well as the wind induced by the vessels motion. Apparent wind is simply the vector sum of the true wind and the negative of the ship’s heading vector, as displayed in the figure below. The resulting apparent wind angle (AWA) is measured from the vessel heading.

![Apparent Wind Vector Diagram](image)

*Figure 11. Apparent Wind Vector Diagram*

**Magnus Effect**

Flettner rotor technology relies on a physics phenomenon called the Magnus effect. Named for the German physicist Heinrich Magnus, it explains the observable deflection of spinning objects as they move through a fluid. A more readily observable example of the Magnus effect is the curveball in baseball; by applying spin (topspin, backspin, or sidespin) the pitcher can impart swerve onto the ball by the same lift force that Flettner rotors produce.
The deflection is caused by a pressure-gradient force that can be explained by Bernoulli’s Principle. Recall that, in its simplest form of incompressible flow, Bernoulli’s Equation is:

\[
\frac{1}{2} \rho v^2 + gz + \frac{p}{\rho} = \text{constant}
\]  

(1)

Where \( \rho \) is the density of the fluid, \( v \) is the velocity of the flow, \( g \) is the acceleration due to gravity, \( z \) is elevation, and \( P \) is the pressure. If Equation (1) is used to compare two points on the two-dimensional cylinder shown in Figure 10, the pressure gradient is revealed. As the spinning cylinder pushes the fluid (air in this case) to the top of the cylinder, the flow speed increases while pressure decreases. Subsequently, the air traveling on the bottom slows down and increases in pressure. The result is a force, called the Magnus force, directed from the high-pressure side to the low-pressure side.

**Lift and Drag Forces**

As with any lifting foil, Flettner rotors produces two forces, lift and drag. The lift is the primary force generated by the rotor and is perpendicular to the apparent wind direction experienced by
the vessel. The drag force is perpendicular to the lift force. Equations for the two forces can be found below:

\[ L = \frac{1}{2} \rho A v_a^2 C_L \]  
\[ D = \frac{1}{2} \rho A v_a^2 C_D \]

Here \( A \) is the cross-sectional area exposed to the wind (or height times diameter), \( v_a \) is the apparent wind speed, and \( C_L \) and \( C_D \) are the respective coefficients. These coefficients are usually determined experimentally in a wind tunnel; a more robust discussion of this process and the values used for this study will be completed in following sections.

A common misconception is that the drag force is always working against the lifting force. As the figure below displays, for rotors in certain wind directions, a component of the drag (albeit much smaller) adds to the component of the lift that is pushing the vessel forward. However, like any lifting foil, the rotors are still most efficient when the onflow is perpendicular to the body, shown in case (a) below.
The forward thrust generated by the rotor sail is the sum of the lift and drag components parallel to the ship’s heading.

\[ F_T = L \sin(\alpha) - D \cos(\alpha) \]  

(4)

Substituting in Equations (2) and (3) for the lift \((L)\) and drag \((D)\) gives:

\[ F_T = \frac{1}{2} \rho A v_a^2 (C_L \sin(\alpha) - C_D \cos(\alpha)) \]  

(5)

**Spin Ratio**

The spin ratio is one of the most important parameters to consider in the design of a Flettner rotor system. It is the only variable parameter that can be constantly optimized and controlled (by adjusting the rotational speed) to match the wind conditions to maximize the usable thrust. It refers to the ratio of the tangential velocity at the rotor’s surface and the apparent wind velocity.

\[ SR = \frac{v_{tangential}}{v_{apparent}} \]  

(6)
The spin ratio affects the wake and unsteadiness, often represented by the Strouhal number, downstream of the rotor. The larger the ratio (and a subsequently larger Strouhal number) leads to the formation of smaller eddies and vice versa for smaller ratios. The result of these smaller eddies is reflected in an increased lift and drag coefficient. A further analysis of the spin ratio and its purpose in this study will be conducted in following sections.

**Aspect Ratio and Thom Disk**

The aspect ratio is a simple geometric ratio relating the height of the Flettner rotor to its diameter, or:

\[ AR = \frac{h}{d} \]  

It too influences the produced lift and drag; tall, narrow rotors increase lift and decrease drag, while short, wide rotors decrease lift and increase drag. However, this parameter is not usually considered in studies because rotor dimensions are restricted by other considerations. These include ship stability, air draft restrictions, deck space available, and commercially available rotor sizes.

Nearly all Flettner rotors feature distinctive endplates, also called Thom disks, at the top of the cylinder. The disks serve to help enhance the aerodynamic efficiency of the rotors by modifying the flow at the tip of the sail. The end plate introduces the idea of an effective aspect ratio corresponding to the diameter of the plate and of the cylinder:

\[ AR_{eff} = \frac{d_{disk}}{d} \]  

The figure below displays different rotor sails with varying aspect ratios and effective aspect ratios.
Coefficient Selection
One of the most important decisions to be made in this study is the selection of the lift and drag coefficients that will be used in the calculations. Computational and/or experimental methods to obtain these values are beyond the scope of the project. Instead, these parameters will be sourced from other technical papers dedicated to finding accurate values.

Most particularly, the work of Craft et al [13] will be employed since it applies directly to Flettner rotors. In fact, other route feasibility studies, including Comer et al [10] and Traut et al [9], use the values proposed by Craft.

Craft uses customized three-dimensional, unsteady Computational Fluid Dynamics (CFD) code with a $k-\varepsilon$ eddy-viscosity model to study the flow past a rotating cylinder (a Flettner rotor). The experimental setup includes a uniform inflow (of wind), zero gradient conditions on outflow, and
no-slip boundaries on surfaces. A 3D representation of the model and meshing is show in the figure below.

![3D representation of the model and meshing](image)

**Figure 15. Craft CFD Setup**

Craft examines the lift and drag coefficients over a range of spin ratios (which he denotes as $\Omega$) and also tests bare rotors versus those with Thom disks. His results are plotted against experimental results from the 1920’s obtained by Thom and National Advisory Committee for Aeronautics (NACA). A sample result plot for the lift coefficient is shown below.
From Craft’s results it is possible to determine that, at a spin ratio of five (as proposed by Lele and Rao), the lift coefficient should be set to 12.5 and the drag coefficient to 0.2.

**Weather Data**
To compute the propulsive contribution of a Flettner rotor at a point it is necessary to know the wind speed and direction at a point. This can present a tricky problem when considering the lack of meteorological data collection locations over open water. While many modern ships have their own sensors to gather this information, the data is usually not publicly accessible. The solution in this study, and many like it, is to utilize an emerging meteorological tool, reanalysis databases.

Reanalysis is a data synthesis method in which information from a wide variety of sources is incorporated, adjusted for instrumental bias, and interpolated using predictive models to produce a comprehensive historic weather database. Reanalysis databases are defined by their spatial resolution, expressed in terms of a longitude and latitude, referring to the grid spacing between unique data points.
This study uses the Copernicus Climate Reanalysis Database. Copernicus is a subdivision of the European Center for Medium-Range Weather Forecasts (ECMWF), an intergovernmental research service with thirty-five member states, and serves a key role in the European Union’s Climate Change Service.

The “ERA5-Land Monthly Averaged Database” is utilized for this study. The “Land” designation implies that the database does not carry data for oceanic environments, however, it does for lake environments (including the Great Lakes). The database includes monthly averages of fifty scientific variables at a spatial resolution of 0.1° x 0.1°. For the purposes of this study the database is trimmed to only include two of the variables, the 10-m u-component of wind and the 10-m v-component of wind, from which it will be possible to calculate the required wind magnitude and heading. The database is also given latitude and longitude limits so that only an area enclosing the Great Lakes is considered. The time domain is restricted to only include months between March and November to encapsulate the Great Lakes shipping season (December to February represents the winter layup period due to icy conditions). Furthermore, the time domain is restricted to only include the last five full years of data (2017-2021) to ensure that calculations are relative to the regions recent weather conditions.

These parameters are submitted to Copernicus and a custom data file in a specialized meteorological file format, NETCDF4, is generated. It should be noted that, although this is a heavily modified data file, the code written for this report is capable of handling unmodified files with minimal adjustments (the file modification is done to increase computational speed by reducing input file size).
**Route Planning**

To predict the propulsive efficiency of the rotors it is necessary to develop routes on which to calculate the generated power. This is accomplished by employing the Python plotting module Plotly and supplying a list of latitude and longitude points. The lists of latitude and longitude inputs for each route can be found in the lake scripts located in the Appendix. These routes are developed as a balance between following the marine traffic density (see Figure 10) and the spatial resolution of the reanalysis database, with resolution limiting the number and location of the points to tenth of a degree increments. The following figures display the key shipping routes modeled for each of the five Great Lakes. It should be noted that many of the routes are overlapping as they share points because of their common destination.
On Lake Michigan, Route 1 represents a passage from the Port of Chicago to the Straits of Mackinac. Route 3 presents a common variation of Route 1, avoiding the potentially dangerous Manitou Passage. Routes 2 and 4 map Marinette and Milwaukee, Wisconsin, respectively, to the Straits of Mackinac. Finally, Route 5 connects the Port of Chicago to Milwaukee.

Figure 17. Lake Michigan Routes
Three routes are generated for Lake Superior, with Routes 1 and 2 connecting the primarily iron ore ports of Duluth, Minnesota and Marquette, Michigan to the Soo Locks. Meanwhile, Route 3 maps the grain rich Thunder Bay, Ontario to the Locks.

*Figure 18. Lake Superior Routes*
Routes 1 and 2 on Lake Huron pick up at points where previous routes left off. Route 1 stretches from the Straits of Mackinac (the termination point of the Lake Michigan routes) to the St. Clair River, while Route 2 connects the Soo Canal (Lake Superior endpoint) to the same point. Route 3 originates in the grain port of Goderich and also ends at the mouth of the St. Clair River.
The first two Lake Erie routes pick up at the exit of the Detroit River (connected to the St. Clair River) and map to Cleveland, Ohio and the Welland Canal (which is near Buffalo, New York). These represent important endpoints because they are common transit hubs for iron ore to be delivered to steel mills in Ohio and Pennsylvania. The final route, Route 3, connects Toledo, Ohio to the Welland Canal as well.
The final set of routes all connect to the St. Lawrence River for eventual transit to the Atlantic Ocean. Route 1 starts at the exit of the Welland Canal, Route 2 at the steel city of Hamilton, Ontario, and Route 3 at Toronto, Ontario.

**Hollenbach’s Method**
Around thirty years ago, Dr. Uwe Hollenbach developed a method for calculating estimates for ship resistance and propulsion of single and twin-screw displacement vessels by creating and implementing regression formulas of test data from the Vienna Model Basin and comparing them against data from the Hamburg Ship Model Basin. The regression formulas used were developed based on the experimental data of 433 vessels with varying drafts.

*Figure 21. Lake Ontario Routes*
Like all resistance methods, Dr. Hollenbach proposed a set of limits on the applicability of his method; these are summarized below.

*Table 1. Hollenbach's Applicability Limits*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Unit</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>ship speed</td>
<td>$v$</td>
<td>m/s</td>
<td>displacement type vessel</td>
</tr>
<tr>
<td>length between perpendiculars</td>
<td>$L_{pp}$</td>
<td>m</td>
<td>42.0 to 205.0</td>
</tr>
<tr>
<td>length-beam ratio</td>
<td>$L_{pp}/T$</td>
<td>-</td>
<td>4.71 to 7.11</td>
</tr>
<tr>
<td>beam-draft ratio</td>
<td>$B/T$</td>
<td>-</td>
<td>1.99 to 4.00</td>
</tr>
<tr>
<td>propeller diameter-draft ratio</td>
<td>$D/T_A$</td>
<td>-</td>
<td>0.43 to 0.84</td>
</tr>
<tr>
<td>block coefficient</td>
<td>$C_B$</td>
<td>-</td>
<td>0.60 to 0.83</td>
</tr>
<tr>
<td>length-displacement ratio</td>
<td>$L_{pp}/\sqrt{\Delta}$</td>
<td>-</td>
<td>4.49 to 6.01</td>
</tr>
</tbody>
</table>

It should also be noted here that Hollenbach’s method has been extended to include wider ranges then were initially proposed.

An advantage of utilizing Dr. Hollenbach’s method is that it only requires a few basic measurements of the hull form, meaning that a resistance and propulsion estimate can be performed with relatively little vessel information. The required inputs for the Hollenbach method are summarized in the table below.

*Table 2. Hollenbach's Required Inputs*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>length between perpendiculars</td>
<td>$L_{pp}$</td>
</tr>
<tr>
<td>Parameter</td>
<td>Symbol</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>length in waterline</td>
<td>$L_{WL}$</td>
</tr>
<tr>
<td>length over wetted surface</td>
<td>$L_{OS}$</td>
</tr>
<tr>
<td>molded beam</td>
<td>$B$</td>
</tr>
<tr>
<td>molded draft at aft perpendicular</td>
<td>$T_A$</td>
</tr>
<tr>
<td>molded draft at forward perpendicular</td>
<td>$T_F$</td>
</tr>
<tr>
<td>propeller diameter</td>
<td>$D$</td>
</tr>
<tr>
<td>block coefficient (based on $L_{pp}$)</td>
<td>$C_B$</td>
</tr>
<tr>
<td>transverse vertical area above waterline</td>
<td>$A_V$</td>
</tr>
</tbody>
</table>

The optional parameters of $S_{BK}$ (bilge keel surface area) and $d_{TH}$ (transverse thruster diameter) will also be implemented to better capture the appendage resistance.

This study only utilizes the resistance portion of Hollenbach’s method as the goal is to achieve an estimate of the effective power required for the vessel (propulsion characteristics are unnecessary).

A comprehensive detailing of Hollenbach’s method is beyond the scope of this study, but one can be found in the References. The Python script implementing the method (hollenbach.py) can also be found in the Appendix and reads input from a specified ship profile module.

**Ship Profile**

With the routes mapped the next step is to consider a vessel on which to base the calculations.

Most Great Lakes ships share common design characteristics given their unique operating environment. Given the relatively short travel distances, cargo capacity is prioritized over speed, leading to large block coefficients. Geometrically, the vessels are restricted by the connecting
seaways. For vessels using the St. Lawrence Seaway to access the Atlantic Ocean, length is restricted to 230 meters and beam to 24 meters. Vessels based solely within the Great Lakes system can stretch up to lengths in excess of 300 meters. Drafts are restricted to 6.4 meters by Lake St. Clair, which connects the St. Clair River to the Detroit River (and therefore connects Lake Huron to Lake Erie). Another distinguishing characteristic is an increased length-to-beam ratio (as compared to ocean bulk carriers). This is in part thanks to reduced longitudinal stresses caused by the Lake’s smaller wave heights and periods.

One of the newest freighters is the Chinese designed and built Algoma Equinox, which entered service in late 2013. Operated by Algoma Central, a Canadian shipping company, the new design claims to be 45% more efficient than previous grain carrying vessels in service in the region. The vessel is considered gearless, so there is no concern that introduction of Flettner rotors would interfere with the ship’s ability to perform its mission.

Figure 22. Algoma Equinox

Table 1 below summarizes the input parameters for the Algoma Equinox ship profile primarily used in this study.
Table 3. Algoma Equinox Particulars

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length between perpendiculars, $L_{pp}$</td>
<td>222.06 m</td>
</tr>
<tr>
<td>Length on waterline, $L_{WS}$</td>
<td>225.55 m</td>
</tr>
<tr>
<td>Length over wetted surface, $L_{OS}$</td>
<td>225.55 m</td>
</tr>
<tr>
<td>Beam, $B$</td>
<td>23.77 m</td>
</tr>
<tr>
<td>Draft, $T$</td>
<td>10.31 m</td>
</tr>
<tr>
<td>Displacement, $V$</td>
<td>47,952 m$^3$</td>
</tr>
<tr>
<td>Wetted surface area of bilge keels, $S_{BK}$</td>
<td>50.0 m$^2$</td>
</tr>
<tr>
<td>Transverse area above waterline, $A_{VS}$</td>
<td>574.9 m$^2$</td>
</tr>
<tr>
<td>Diameter of bow thruster tunnel, $d_{TH}$</td>
<td>1.20 m</td>
</tr>
<tr>
<td>Ship service speed, $v$</td>
<td>13.7 kn</td>
</tr>
</tbody>
</table>

The transverse area above the waterline is an important parameter in this study since it must be amended to include the added lateral surface area of the rotors. This is done by the following equation:

$$A_{VST} = A_{VS} + (D \times H \times N)$$  \hspace{1cm} (9)

Where $D$ is the diameter of the rotor, $H$ is the height, and $N$ is the number of rotors installed. The increased surface area will increase the environmental resistance calculated in Hollenbach’s method.

Values for bilge keel surface area and bow thruster diameter were not available for the Algoma Equinox specifically, so values were sourced from similarly sized vessels. Hence this study uses “vessel profiles” rather than modeling an exact vessel.
It should also be mentioned that the Algoma Equinox ship profile does test the applicability limits of Hollenbach’s method. Though, as mentioned previously, the method has been extended beyond the original limits and the method is designed for bulk carrier vessels such as this, so little risk is assumed.

Calculations

This section summarizes the calculations completed in the spinnthingcalcs_2.py script that can be found in the Appendix. The first set of calculations finds the ship heading and distance traveled between two points along the selected route. The heading is found in two steps, first by calculating values $x$ and $y$:

\[
    x = \cos(lat_{n+1}) \times \sin(lon_{n+1} - lon_n) \tag{10}
\]

\[
    y = \cos(lat_n) \times \sin(lon_{n+1}) - \sin(lat_n) \times \cos(lat_{n+1}) \times \cos(lon_{n+1} - lon_n) \tag{11}
\]

From here the bearing, $\beta$, is:

\[
    \beta = \arctan\left(\frac{x}{y}\right) \tag{12}
\]

To model the estimated fuel and emissions savings over the two points it is necessary to know the distance between the points. A linear distance formula will not suffice for this problem due to the curvature of the Earth, therefore the Haversine formula is employed.

\[
    d = 2R \arcsin\left(\sin^2\left(\frac{lat_{n+1} - lat_n}{2}\right) + \cos(lat_n) \cos(lat_{n+1}) \sin^2\left(\frac{lon_{n+1} - lon_n}{2}\right)\right) \tag{13}
\]
Where $R$ is the radius of the Earth, taken as 6,371 km. The Haversine formula can present an error of up to 0.5% as the Earth is not perfectly spherical, but these errors only arise over distances much larger than those considered in this study.

It is now possible to estimate the travel time by assuming the ship maintains a constant service speed as simply:

$$t = \frac{d}{v_s} \quad (14)$$

With this setup complete the script can now begin to consider the wind data. As discussed in the Theory section, the apparent wind magnitude and direction must be calculated. This is done component-wise as:

$$u = -v_s \sin(\beta) + u_{10} \quad (15)$$

$$v = -v_s \cos(\beta) + v_{10} \quad (16)$$

Simple vector math gives:

$$w_{true} = \sqrt{u^2 + v^2} \quad (17)$$

$$\kappa = \arctan \left( \frac{u}{v} \right) \quad (18)$$

Finally, the angle of attack, $\alpha$, of the true wind relative to the ships heading is:

$$\alpha = \text{abs}(\beta - \kappa) \quad (19)$$

From here, Equations (2), (3), and (4) of the Theory section are used to calculate the lift ($L$), drag ($D$), and forward thrust ($F_T$), respectively.

The power generated by the rotor(s) is then:
\[ P_{\text{gen}} = F_T \times v_s \times N \]  

(20)

To model the power consumed in spinning the rotor(s) the method developed by Comer et al is employed. This method takes the power to be that required to overcome the frictional resistance of the air. First the Reynolds number of the flow is computed:

\[ Re = \frac{\rho \times SR \times w_{\text{true}} \times \pi D}{\mu} \]  

(21)

Second, a frictional coefficient is calculated:

\[ C_f = \frac{0.455}{(\log(Re))^{2.58}} - \frac{1700}{Re} \]  

(22)

The frictional force is then:

\[ F_f = \frac{1}{2} \times C_f \times \rho \times A \times (SR \times w_{\text{true}})^2 \]  

(23)

Here, \( A \), remains the surface area of the rotor. Then the consumed power is:

\[ P_{\text{con}} = F_f \times (SR \times w_{\text{true}}) \times N \]  

(24)

The net power is simply:

\[ P_{\text{net}} = (P_{\text{gen}} - P_{\text{con}}) \times N \times \eta_p \]  

(25)

Where \( \eta_p \) is the propulsive efficiency, and is derived as:

\[ \eta_{\text{Prop}} = \eta_o \times \eta_R \times \eta_H \times \eta_s \]  

(26)

Values for \( \eta_o \), \( \eta_s \), and \( \eta_R \) are taken as 0.65, 0.98, and 1.009 from MAN Energy Solutions Basic Principles of Ship Propulsion [14]. From the same reference, a thrust deduction factor of 0.2 and
a wake fraction of 0.32 (ranged values for single-screw vessels) are used to determine the hull efficiency as:

\[ \eta_H = \frac{1 - t}{1 - w} \]  

(27)

Yielding a hull efficiency of 1.18 and, by plugging the efficiencies into Equation (26), a propulsive efficiency of 0.75.

A safety feature is built into the Python script; in cases where the net power is negative, net power is set to zero and the rotor is assumed to be turned off.

A propulsive contribution percentage is also calculated using the effective power estimate provided by Hollenbach’s method.

\[ P_{sav} = \frac{P_{net}}{P_E} \times 100\% \]  

(28)

The last portion of the calculation script attempts to model the mass of fuel and CO₂ savings. Since each engine will burn fuel at different rates, a more generalized approach is utilized. Specific energy values for Marine Diesel Oil (MDO), the most common fuel used in maritime shipping, are used, and summarized in the table below.

**Table 4. Specific Energy Ranges for MDO**

<table>
<thead>
<tr>
<th>Parameter:</th>
<th>Specific Energy (g/kW-hr):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel, (E_f)</td>
<td>205 – 211</td>
</tr>
<tr>
<td>CO₂, (E_{CO2})</td>
<td>580 – 630</td>
</tr>
</tbody>
</table>
Median values of 208 g/kW-hr and 610 g/kW-hr are selected for calculations [15]. This implies the mass savings of each as:

\[
fuel_{sav} = E_f \times P_{net} \times t
\]  
(29)

\[
CO2_{sav} = E_{CO2} \times P_{net} \times t
\]  
(30)

Results

The results presented in this section model a single Flettner rotor installed upon the Algoma Equinox ship profile. The key input values utilized for the calculations are presented in the table below.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Number of Rotors & 1 \\
\hline
Rotor Diameter & 5 m \\
\hline
Rotor Height & 30 m \\
\hline
Spin Ratio & 5 \\
\hline
Lift Coefficient & 12.50 \\
\hline
Drag Coefficient & 0.20 \\
\hline
Propulsive Efficiency & 0.75 \\
\hline
Density of Air & 1.23 kg/m$^3$ \\
\hline
Dynamic Viscosity of Air & 0.20 kg/m·s \\
\hline
Ship Speed & 13.7 knots \\
\hline
Effective Power & 3512.5 kW \\
\hline
\end{tabular}
\caption{Input Values}
\end{table}
The effective power is generated from the resistance estimate performed by the hollenbach.py script and does not need to be directly entered. The other parameters are defined in the RunFile.py script in order to make the code more adaptable. Configurations with multiple rotors (or rotors of different sizes) can easily be tested by operating the code from this central script.

A 30-meter by 5-meter rotor is selected as it is the most commercially popular size of those offered by Norsepower and the only one to have existing tiltable technology, which may be required for some of the defined routes.

The first result generated by the code is a wind rose plot for the given ship profile. The rose plot is a useful tool for visualizing expected rotor power contributions given a variety of true wind speeds and headings. In the figure below the vessel is assumed to be traveling at a heading of 0° and true wind values are displayed.

![Propulsive Power Generated by a Flettner Rotor](image)

**Figure 23. Algoma Equinox Rose Plot**
The rose plot will be unchanged for each lake as it only depends on the defined ship profile.

Results in the following sections will be presented for each lake and relevant trends will be discussed.

**Lake Michigan**

First, monthly results are presented by route in the table below.

### Table 6. Lake Michigan - Route 1

<table>
<thead>
<tr>
<th>Month</th>
<th>$P_{\text{gen}}$ [kW]</th>
<th>$P_{\text{con}}$ [kW]</th>
<th>$P_{\text{net}}$ [kW]</th>
<th>$P_{\text{sav}}$ [%]</th>
<th>Fuel$_{\text{sav}}$ [kg]</th>
<th>CO2$_{\text{sav}}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>164.22</td>
<td>3.43</td>
<td>120.59</td>
<td>3.43</td>
<td>460.61</td>
<td>1350.82</td>
</tr>
<tr>
<td>Apr.</td>
<td>134.55</td>
<td>3.75</td>
<td>98.10</td>
<td>2.79</td>
<td>337.94</td>
<td>991.06</td>
</tr>
<tr>
<td>May.</td>
<td>78.42</td>
<td>3.09</td>
<td>56.50</td>
<td>1.61</td>
<td>223.67</td>
<td>655.95</td>
</tr>
<tr>
<td>Jun.</td>
<td>56.92</td>
<td>1.93</td>
<td>41.24</td>
<td>1.17</td>
<td>141.45</td>
<td>414.83</td>
</tr>
<tr>
<td>Jul.</td>
<td>100.97</td>
<td>2.66</td>
<td>73.74</td>
<td>2.10</td>
<td>309.63</td>
<td>908.04</td>
</tr>
<tr>
<td>Aug.</td>
<td>83.42</td>
<td>1.97</td>
<td>61.08</td>
<td>1.74</td>
<td>243.83</td>
<td>715.08</td>
</tr>
<tr>
<td>Sep.</td>
<td>64.69</td>
<td>1.80</td>
<td>47.17</td>
<td>1.34</td>
<td>145.94</td>
<td>428.01</td>
</tr>
<tr>
<td>Oct.</td>
<td>164.53</td>
<td>1.88</td>
<td>121.99</td>
<td>3.47</td>
<td>519.09</td>
<td>1522.34</td>
</tr>
<tr>
<td>Nov.</td>
<td>362.77</td>
<td>1.67</td>
<td>270.83</td>
<td>7.71</td>
<td>1195.87</td>
<td>3507.11</td>
</tr>
</tbody>
</table>

### Table 7. Lake Michigan - Route 2

<table>
<thead>
<tr>
<th>Month</th>
<th>$P_{\text{gen}}$ [kW]</th>
<th>$P_{\text{con}}$ [kW]</th>
<th>$P_{\text{net}}$ [kW]</th>
<th>$P_{\text{sav}}$ [%]</th>
<th>Fuel$_{\text{sav}}$ [kg]</th>
<th>CO2$_{\text{sav}}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>133.62</td>
<td>2.40</td>
<td>98.41</td>
<td>2.80</td>
<td>98.89</td>
<td>290.02</td>
</tr>
<tr>
<td>Apr.</td>
<td>172.20</td>
<td>3.53</td>
<td>126.51</td>
<td>3.60</td>
<td>104.37</td>
<td>306.07</td>
</tr>
<tr>
<td>May.</td>
<td>80.59</td>
<td>3.55</td>
<td>57.78</td>
<td>1.64</td>
<td>64.23</td>
<td>188.38</td>
</tr>
<tr>
<td>Jun.</td>
<td>127.70</td>
<td>2.33</td>
<td>94.03</td>
<td>2.68</td>
<td>79.38</td>
<td>232.80</td>
</tr>
<tr>
<td>Jul.</td>
<td>68.96</td>
<td>1.90</td>
<td>50.29</td>
<td>1.43</td>
<td>37.90</td>
<td>111.14</td>
</tr>
<tr>
<td>Aug.</td>
<td>81.09</td>
<td>1.75</td>
<td>59.51</td>
<td>1.69</td>
<td>40.15</td>
<td>117.74</td>
</tr>
<tr>
<td>Sep.</td>
<td>131.71</td>
<td>1.67</td>
<td>97.53</td>
<td>2.78</td>
<td>64.30</td>
<td>188.59</td>
</tr>
<tr>
<td>Oct.</td>
<td>97.40</td>
<td>1.62</td>
<td>71.84</td>
<td>2.04</td>
<td>78.95</td>
<td>231.53</td>
</tr>
<tr>
<td>Nov.</td>
<td>162.84</td>
<td>0.80</td>
<td>121.53</td>
<td>3.46</td>
<td>162.96</td>
<td>477.93</td>
</tr>
</tbody>
</table>

### Table 8. Lake Michigan - Route 3

<table>
<thead>
<tr>
<th>Month</th>
<th>$P_{\text{gen}}$ [kW]</th>
<th>$P_{\text{con}}$ [kW]</th>
<th>$P_{\text{net}}$ [kW]</th>
<th>$P_{\text{sav}}$ [%]</th>
<th>Fuel$_{\text{sav}}$ [kg]</th>
<th>CO2$_{\text{sav}}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>158.83</td>
<td>3.38</td>
<td>116.59</td>
<td>3.32</td>
<td>467.70</td>
<td>1371.63</td>
</tr>
</tbody>
</table>
### Table 9. Lake Michigan - Route 4

<table>
<thead>
<tr>
<th>Month</th>
<th>P_{gen} [kW]</th>
<th>P_{con} [kW]</th>
<th>P_{net} [kW]</th>
<th>P_{sav} [%]</th>
<th>Fuel_{sav} [kg]</th>
<th>CO2_{sav} [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>178.47</td>
<td>3.19</td>
<td>131.46</td>
<td>3.74</td>
<td>374.31</td>
<td>1097.75</td>
</tr>
<tr>
<td>Apr.</td>
<td>148.00</td>
<td>3.69</td>
<td>108.23</td>
<td>3.08</td>
<td>280.27</td>
<td>821.96</td>
</tr>
<tr>
<td>May.</td>
<td>82.38</td>
<td>3.20</td>
<td>59.38</td>
<td>1.69</td>
<td>159.28</td>
<td>467.11</td>
</tr>
<tr>
<td>Jun.</td>
<td>71.10</td>
<td>1.96</td>
<td>51.85</td>
<td>1.48</td>
<td>131.03</td>
<td>384.26</td>
</tr>
<tr>
<td>Jul.</td>
<td>77.55</td>
<td>2.37</td>
<td>56.38</td>
<td>1.60</td>
<td>178.04</td>
<td>522.15</td>
</tr>
<tr>
<td>Aug.</td>
<td>78.78</td>
<td>1.85</td>
<td>57.70</td>
<td>1.64</td>
<td>170.66</td>
<td>500.48</td>
</tr>
<tr>
<td>Sep.</td>
<td>78.92</td>
<td>1.72</td>
<td>57.90</td>
<td>1.65</td>
<td>129.13</td>
<td>378.70</td>
</tr>
<tr>
<td>Oct.</td>
<td>132.86</td>
<td>1.71</td>
<td>98.36</td>
<td>2.80</td>
<td>289.98</td>
<td>850.44</td>
</tr>
<tr>
<td>Nov.</td>
<td>284.58</td>
<td>1.23</td>
<td>212.52</td>
<td>6.05</td>
<td>660.40</td>
<td>1936.76</td>
</tr>
</tbody>
</table>

### Table 10. Lake Michigan - Route 5

<table>
<thead>
<tr>
<th>Month</th>
<th>P_{gen} [kW]</th>
<th>P_{con} [kW]</th>
<th>P_{net} [kW]</th>
<th>P_{sav} [%]</th>
<th>Fuel_{sav} [kg]</th>
<th>CO2_{sav} [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>41.71</td>
<td>3.62</td>
<td>28.57</td>
<td>0.81</td>
<td>27.72</td>
<td>81.29</td>
</tr>
<tr>
<td>Apr.</td>
<td>51.82</td>
<td>3.72</td>
<td>36.08</td>
<td>1.03</td>
<td>34.28</td>
<td>100.52</td>
</tr>
<tr>
<td>May.</td>
<td>107.95</td>
<td>2.83</td>
<td>78.84</td>
<td>2.24</td>
<td>102.79</td>
<td>301.44</td>
</tr>
<tr>
<td>Jun.</td>
<td>65.69</td>
<td>2.31</td>
<td>47.54</td>
<td>1.35</td>
<td>46.07</td>
<td>135.11</td>
</tr>
<tr>
<td>Jul.</td>
<td>41.39</td>
<td>3.26</td>
<td>28.60</td>
<td>0.81</td>
<td>35.96</td>
<td>105.46</td>
</tr>
<tr>
<td>Aug.</td>
<td>51.10</td>
<td>2.34</td>
<td>36.57</td>
<td>1.04</td>
<td>34.30</td>
<td>100.58</td>
</tr>
<tr>
<td>Sep.</td>
<td>51.75</td>
<td>2.29</td>
<td>37.09</td>
<td>1.06</td>
<td>33.78</td>
<td>99.05</td>
</tr>
<tr>
<td>Oct.</td>
<td>222.09</td>
<td>2.93</td>
<td>164.37</td>
<td>4.68</td>
<td>191.84</td>
<td>562.61</td>
</tr>
<tr>
<td>Nov.</td>
<td>488.11</td>
<td>3.82</td>
<td>363.22</td>
<td>10.34</td>
<td>426.11</td>
<td>1249.65</td>
</tr>
</tbody>
</table>

The propulsive power contribution results are summarized graphically in the figure shown below.
Figure 24. Lake Michigan Power Contributions

The plot helps to display a clear (and expected) result. Power contributions are highest in windy seasons (Spring and Fall), particularly in October and November, and noticeably lower in the summer months, where contributions drop to just 2% of the required power.

The final table for this lake displays the average power contributions for all months of the mapped routes.

Table 11. Lake Michigan - Mean Results by Route

<table>
<thead>
<tr>
<th>Route</th>
<th>P\text{\scriptsize gen} [kW]</th>
<th>P\text{\scriptsize con} [kW]</th>
<th>P\text{\scriptsize net} [kW]</th>
<th>P\text{\scriptsize sav} [%]</th>
<th>Fuel\text{\scriptsize sav} [kg]</th>
<th>CO2\text{\scriptsize sav} [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>134.50</td>
<td>2.46</td>
<td>99.03</td>
<td>2.82</td>
<td>397.56</td>
<td>1165.92</td>
</tr>
<tr>
<td>2</td>
<td>117.34</td>
<td>2.17</td>
<td>86.38</td>
<td>2.46</td>
<td>81.24</td>
<td>238.24</td>
</tr>
<tr>
<td>3</td>
<td>144.92</td>
<td>2.46</td>
<td>106.84</td>
<td>3.04</td>
<td>434.19</td>
<td>1273.34</td>
</tr>
<tr>
<td>4</td>
<td>125.85</td>
<td>2.32</td>
<td>92.64</td>
<td>2.64</td>
<td>263.68</td>
<td>773.29</td>
</tr>
<tr>
<td>5</td>
<td>124.62</td>
<td>3.01</td>
<td>91.21</td>
<td>2.60</td>
<td>103.65</td>
<td>303.97</td>
</tr>
</tbody>
</table>
One thing to note here is that Route 3, an alteration of Route 1 as mentioned previously, displays higher savings (3.04% vs 2.82%) suggesting that avoiding the Manitou Passage is not only safer but also beneficial to rotor performance.

Lake Superior

Table 12. Lake Superior - Route 1

<table>
<thead>
<tr>
<th>Month</th>
<th>$P_{gen}$ [kW]</th>
<th>$P_{con}$ [kW]</th>
<th>$P_{net}$ [kW]</th>
<th>$P_{sav}$ [%]</th>
<th>$Fuel_{sav}$ [kg]</th>
<th>$CO2_{sav}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>119.30</td>
<td>2.11</td>
<td>87.89</td>
<td>2.50</td>
<td>152.11</td>
<td>446.09</td>
</tr>
<tr>
<td>Apr.</td>
<td>190.65</td>
<td>3.14</td>
<td>140.63</td>
<td>4.00</td>
<td>235.84</td>
<td>691.64</td>
</tr>
<tr>
<td>May.</td>
<td>67.23</td>
<td>3.50</td>
<td>47.79</td>
<td>1.36</td>
<td>74.38</td>
<td>218.13</td>
</tr>
<tr>
<td>Jun.</td>
<td>117.49</td>
<td>3.11</td>
<td>85.78</td>
<td>2.44</td>
<td>147.48</td>
<td>432.51</td>
</tr>
<tr>
<td>Jul.</td>
<td>100.56</td>
<td>1.97</td>
<td>73.94</td>
<td>2.10</td>
<td>137.82</td>
<td>404.18</td>
</tr>
<tr>
<td>Aug.</td>
<td>103.94</td>
<td>2.23</td>
<td>76.28</td>
<td>2.17</td>
<td>141.86</td>
<td>416.02</td>
</tr>
<tr>
<td>Sep.</td>
<td>176.53</td>
<td>2.34</td>
<td>130.64</td>
<td>3.72</td>
<td>247.43</td>
<td>725.64</td>
</tr>
<tr>
<td>Oct.</td>
<td>91.09</td>
<td>1.86</td>
<td>66.92</td>
<td>1.90</td>
<td>132.96</td>
<td>389.94</td>
</tr>
<tr>
<td>Nov.</td>
<td>129.95</td>
<td>0.83</td>
<td>96.85</td>
<td>2.76</td>
<td>198.26</td>
<td>581.42</td>
</tr>
</tbody>
</table>

Table 13. Lake Superior - Route 2

<table>
<thead>
<tr>
<th>Month</th>
<th>$P_{gen}$ [kW]</th>
<th>$P_{con}$ [kW]</th>
<th>$P_{net}$ [kW]</th>
<th>$P_{sav}$ [%]</th>
<th>$Fuel_{sav}$ [kg]</th>
<th>$CO2_{sav}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>167.16</td>
<td>1.97</td>
<td>123.90</td>
<td>3.53</td>
<td>89.32</td>
<td>261.96</td>
</tr>
<tr>
<td>Apr.</td>
<td>195.02</td>
<td>2.72</td>
<td>144.22</td>
<td>4.10</td>
<td>99.25</td>
<td>291.07</td>
</tr>
<tr>
<td>May.</td>
<td>43.86</td>
<td>2.88</td>
<td>30.74</td>
<td>0.87</td>
<td>35.54</td>
<td>66.62</td>
</tr>
<tr>
<td>Jun.</td>
<td>93.31</td>
<td>2.58</td>
<td>68.05</td>
<td>1.94</td>
<td>52.73</td>
<td>154.64</td>
</tr>
<tr>
<td>Jul.</td>
<td>52.71</td>
<td>2.12</td>
<td>37.95</td>
<td>1.08</td>
<td>35.54</td>
<td>104.23</td>
</tr>
<tr>
<td>Aug.</td>
<td>62.68</td>
<td>2.22</td>
<td>45.35</td>
<td>1.29</td>
<td>41.40</td>
<td>121.40</td>
</tr>
<tr>
<td>Sep.</td>
<td>164.54</td>
<td>2.32</td>
<td>121.67</td>
<td>3.46</td>
<td>98.39</td>
<td>288.55</td>
</tr>
<tr>
<td>Oct.</td>
<td>85.92</td>
<td>2.15</td>
<td>62.82</td>
<td>1.79</td>
<td>52.58</td>
<td>154.20</td>
</tr>
<tr>
<td>Nov.</td>
<td>169.03</td>
<td>1.16</td>
<td>125.90</td>
<td>3.58</td>
<td>108.92</td>
<td>319.43</td>
</tr>
</tbody>
</table>

Table 14. Lake Superior - Route 3

<table>
<thead>
<tr>
<th>Month</th>
<th>$P_{gen}$ [kW]</th>
<th>$P_{con}$ [kW]</th>
<th>$P_{net}$ [kW]</th>
<th>$P_{sav}$ [%]</th>
<th>$Fuel_{sav}$ [kg]</th>
<th>$CO2_{sav}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>71.58</td>
<td>1.66</td>
<td>52.44</td>
<td>1.49</td>
<td>71.45</td>
<td>209.54</td>
</tr>
<tr>
<td>Apr.</td>
<td>195.73</td>
<td>2.43</td>
<td>144.98</td>
<td>4.13</td>
<td>210.70</td>
<td>617.93</td>
</tr>
<tr>
<td>May.</td>
<td>41.90</td>
<td>2.95</td>
<td>29.21</td>
<td>0.83</td>
<td>43.22</td>
<td>126.75</td>
</tr>
</tbody>
</table>
Interestingly, Lake Superior, famous for its rough and windy conditions, displays lower savings values than other the other Great Lakes, though the values are more consistent month to month. This can be explained, however, by the almost strictly West to East nature of the routes (compared to other lakes which map more North-South routes). This obviously effects the encountered wind heading and thus impacts rotor performance.

**Table 15. Lake Superior - Mean Results by Route**

<table>
<thead>
<tr>
<th>Route</th>
<th>$P_{gen}$ [kW]</th>
<th>$P_{con}$ [kW]</th>
<th>$P_{net}$ [kW]</th>
<th>$P_{sav}$ [%]</th>
<th>Fuel$_{sav}$ [kg]</th>
<th>CO2$_{sav}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>121.86</td>
<td>2.34</td>
<td>89.64</td>
<td>2.55</td>
<td>163.13</td>
<td>478.40</td>
</tr>
</tbody>
</table>
Lake Huron

<table>
<thead>
<tr>
<th>Month</th>
<th>$P_{\text{gen}}$ [kW]</th>
<th>$P_{\text{con}}$ [kW]</th>
<th>$P_{\text{net}}$ [kW]</th>
<th>$P_{\text{sav}}$ [%]</th>
<th>Fuel$_{\text{sav}}$ [kg]</th>
<th>CO2$_{\text{sav}}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>109.59</td>
<td>1.56</td>
<td>81.03</td>
<td>2.31</td>
<td>245.38</td>
<td>719.61</td>
</tr>
<tr>
<td>Apr.</td>
<td>43.72</td>
<td>1.96</td>
<td>31.32</td>
<td>0.89</td>
<td>93.52</td>
<td>274.26</td>
</tr>
<tr>
<td>May.</td>
<td>43.88</td>
<td>2.63</td>
<td>30.94</td>
<td>0.88</td>
<td>83.22</td>
<td>244.07</td>
</tr>
<tr>
<td>Jun.</td>
<td>146.92</td>
<td>3.31</td>
<td>107.71</td>
<td>3.07</td>
<td>287.57</td>
<td>843.36</td>
</tr>
<tr>
<td>Jul.</td>
<td>119.11</td>
<td>2.39</td>
<td>87.53</td>
<td>2.49</td>
<td>220.68</td>
<td>647.18</td>
</tr>
<tr>
<td>Aug.</td>
<td>164.15</td>
<td>3.11</td>
<td>120.78</td>
<td>3.44</td>
<td>313.41</td>
<td>919.13</td>
</tr>
<tr>
<td>Sep.</td>
<td>191.49</td>
<td>3.69</td>
<td>140.85</td>
<td>4.01</td>
<td>338.14</td>
<td>991.65</td>
</tr>
<tr>
<td>Nov.</td>
<td>645.53</td>
<td>3.28</td>
<td>481.69</td>
<td>13.71</td>
<td>1398.39</td>
<td>4101.06</td>
</tr>
</tbody>
</table>

Table 16. Lake Huron - Route 1

<table>
<thead>
<tr>
<th>Month</th>
<th>$P_{\text{gen}}$ [kW]</th>
<th>$P_{\text{con}}$ [kW]</th>
<th>$P_{\text{net}}$ [kW]</th>
<th>$P_{\text{sav}}$ [%]</th>
<th>Fuel$_{\text{sav}}$ [kg]</th>
<th>CO2$_{\text{sav}}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>87.61</td>
<td>1.60</td>
<td>64.50</td>
<td>1.84</td>
<td>208.76</td>
<td>612.23</td>
</tr>
<tr>
<td>Apr.</td>
<td>64.95</td>
<td>2.07</td>
<td>47.16</td>
<td>1.34</td>
<td>106.89</td>
<td>313.47</td>
</tr>
<tr>
<td>May.</td>
<td>51.75</td>
<td>2.66</td>
<td>36.82</td>
<td>1.05</td>
<td>83.45</td>
<td>244.74</td>
</tr>
<tr>
<td>Jun.</td>
<td>115.06</td>
<td>3.06</td>
<td>84.00</td>
<td>2.39</td>
<td>240.46</td>
<td>705.21</td>
</tr>
<tr>
<td>Jul.</td>
<td>78.49</td>
<td>2.23</td>
<td>57.20</td>
<td>1.63</td>
<td>148.94</td>
<td>436.81</td>
</tr>
<tr>
<td>Aug.</td>
<td>126.69</td>
<td>2.80</td>
<td>92.92</td>
<td>2.64</td>
<td>249.77</td>
<td>732.50</td>
</tr>
<tr>
<td>Sep.</td>
<td>170.17</td>
<td>3.31</td>
<td>125.15</td>
<td>3.56</td>
<td>292.04</td>
<td>856.46</td>
</tr>
<tr>
<td>Oct.</td>
<td>250.28</td>
<td>3.24</td>
<td>185.29</td>
<td>5.27</td>
<td>556.88</td>
<td>1633.14</td>
</tr>
<tr>
<td>Nov.</td>
<td>515.67</td>
<td>2.73</td>
<td>384.70</td>
<td>10.95</td>
<td>1251.91</td>
<td>3671.47</td>
</tr>
</tbody>
</table>

Table 17. Lake Huron - Route 2

<table>
<thead>
<tr>
<th>Month</th>
<th>$P_{\text{gen}}$ [kW]</th>
<th>$P_{\text{con}}$ [kW]</th>
<th>$P_{\text{net}}$ [kW]</th>
<th>$P_{\text{sav}}$ [%]</th>
<th>Fuel$_{\text{sav}}$ [kg]</th>
<th>CO2$_{\text{sav}}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>288.65</td>
<td>3.00</td>
<td>214.23</td>
<td>6.10</td>
<td>138.00</td>
<td>404.73</td>
</tr>
<tr>
<td>Apr.</td>
<td>132.22</td>
<td>2.81</td>
<td>97.06</td>
<td>2.76</td>
<td>62.52</td>
<td>183.36</td>
</tr>
<tr>
<td>May.</td>
<td>55.51</td>
<td>2.67</td>
<td>39.63</td>
<td>1.13</td>
<td>25.53</td>
<td>74.86</td>
</tr>
<tr>
<td>Jun.</td>
<td>89.15</td>
<td>3.95</td>
<td>63.90</td>
<td>1.82</td>
<td>41.15</td>
<td>120.69</td>
</tr>
<tr>
<td>Jul.</td>
<td>151.05</td>
<td>2.96</td>
<td>111.06</td>
<td>3.16</td>
<td>71.55</td>
<td>209.82</td>
</tr>
<tr>
<td>Aug.</td>
<td>91.31</td>
<td>3.73</td>
<td>65.69</td>
<td>1.87</td>
<td>42.31</td>
<td>124.10</td>
</tr>
</tbody>
</table>

Table 18. Lake Huron - Route 3
**Figure 26. Lake Huron Power Contributions**

Route 1 of Lake Huron (Soo Locks to St. Clair River) presents the highest power saving seen in this report, with the single rotor supplying 13.71% of the ships necessary power in the month of November. This is significant because this is one of the busiest routes in all of the Great Lakes.

**Table 19. Lake Huron - Mean Results by Route**

<table>
<thead>
<tr>
<th>Route</th>
<th>$P_{gen}$ [kW]</th>
<th>$P_{con}$ [kW]</th>
<th>$P_{net}$ [kW]</th>
<th>$P_{sav}$ [%]</th>
<th>$Fuel_{sav}$ [kg]</th>
<th>$CO2_{sav}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>195.00</td>
<td>2.83</td>
<td>144.12</td>
<td>4.10</td>
<td>398.48</td>
<td>1168.62</td>
</tr>
<tr>
<td>2</td>
<td>162.30</td>
<td>2.63</td>
<td>119.75</td>
<td>3.41</td>
<td>348.79</td>
<td>1022.89</td>
</tr>
<tr>
<td>3</td>
<td>160.88</td>
<td>4.04</td>
<td>117.63</td>
<td>3.35</td>
<td>75.77</td>
<td>222.21</td>
</tr>
</tbody>
</table>
### Table 20. Lake Erie - Route 1

<table>
<thead>
<tr>
<th>Month</th>
<th>P&lt;sub&gt;gen&lt;/sub&gt; [kW]</th>
<th>P&lt;sub&gt;con&lt;/sub&gt; [kW]</th>
<th>P&lt;sub&gt;net&lt;/sub&gt; [kW]</th>
<th>P&lt;sub&gt;sav&lt;/sub&gt; [%]</th>
<th>Fuel&lt;sub&gt;sav&lt;/sub&gt; [kg]</th>
<th>CO2&lt;sub&gt;sav&lt;/sub&gt; [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>116.77</td>
<td>1.78</td>
<td>86.24</td>
<td>2.45</td>
<td>136.46</td>
<td>400.20</td>
</tr>
<tr>
<td>Apr.</td>
<td>74.43</td>
<td>1.95</td>
<td>54.36</td>
<td>1.55</td>
<td>89.27</td>
<td>261.80</td>
</tr>
<tr>
<td>May.</td>
<td>47.74</td>
<td>2.64</td>
<td>33.82</td>
<td>0.96</td>
<td>42.20</td>
<td>123.76</td>
</tr>
<tr>
<td>Jun.</td>
<td>141.17</td>
<td>1.90</td>
<td>104.45</td>
<td>2.97</td>
<td>115.80</td>
<td>339.60</td>
</tr>
<tr>
<td>Jul.</td>
<td>72.53</td>
<td>1.93</td>
<td>52.95</td>
<td>1.51</td>
<td>72.90</td>
<td>213.79</td>
</tr>
<tr>
<td>Aug.</td>
<td>125.36</td>
<td>1.96</td>
<td>92.55</td>
<td>2.63</td>
<td>106.39</td>
<td>312.01</td>
</tr>
<tr>
<td>Sep.</td>
<td>101.99</td>
<td>2.75</td>
<td>74.43</td>
<td>2.12</td>
<td>101.50</td>
<td>297.67</td>
</tr>
<tr>
<td>Oct.</td>
<td>130.70</td>
<td>1.57</td>
<td>96.85</td>
<td>2.76</td>
<td>113.60</td>
<td>333.16</td>
</tr>
<tr>
<td>Nov.</td>
<td>141.08</td>
<td>0.82</td>
<td>105.19</td>
<td>2.99</td>
<td>136.99</td>
<td>401.75</td>
</tr>
</tbody>
</table>

### Table 21. Lake Erie - Route 2

<table>
<thead>
<tr>
<th>Month</th>
<th>P&lt;sub&gt;gen&lt;/sub&gt; [kW]</th>
<th>P&lt;sub&gt;con&lt;/sub&gt; [kW]</th>
<th>P&lt;sub&gt;net&lt;/sub&gt; [kW]</th>
<th>P&lt;sub&gt;sav&lt;/sub&gt; [%]</th>
<th>Fuel&lt;sub&gt;sav&lt;/sub&gt; [kg]</th>
<th>CO2&lt;sub&gt;sav&lt;/sub&gt; [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>120.21</td>
<td>1.80</td>
<td>88.81</td>
<td>2.53</td>
<td>140.74</td>
<td>412.73</td>
</tr>
<tr>
<td>Apr.</td>
<td>74.32</td>
<td>1.91</td>
<td>54.31</td>
<td>1.55</td>
<td>89.56</td>
<td>262.64</td>
</tr>
<tr>
<td>May.</td>
<td>47.65</td>
<td>2.59</td>
<td>33.79</td>
<td>0.96</td>
<td>42.28</td>
<td>124.00</td>
</tr>
<tr>
<td>Jun.</td>
<td>119.38</td>
<td>1.62</td>
<td>88.32</td>
<td>2.51</td>
<td>92.31</td>
<td>270.73</td>
</tr>
<tr>
<td>Jul.</td>
<td>62.65</td>
<td>1.82</td>
<td>45.62</td>
<td>1.30</td>
<td>62.33</td>
<td>182.79</td>
</tr>
<tr>
<td>Aug.</td>
<td>104.15</td>
<td>1.73</td>
<td>76.82</td>
<td>2.19</td>
<td>83.43</td>
<td>244.68</td>
</tr>
<tr>
<td>Sep.</td>
<td>104.36</td>
<td>2.66</td>
<td>76.27</td>
<td>2.17</td>
<td>104.36</td>
<td>306.07</td>
</tr>
<tr>
<td>Oct.</td>
<td>96.81</td>
<td>1.29</td>
<td>71.64</td>
<td>2.04</td>
<td>76.75</td>
<td>225.09</td>
</tr>
<tr>
<td>Nov.</td>
<td>80.09</td>
<td>0.43</td>
<td>59.74</td>
<td>1.70</td>
<td>70.70</td>
<td>207.33</td>
</tr>
</tbody>
</table>

### Table 22. Lake Erie - Route 3

<table>
<thead>
<tr>
<th>Month</th>
<th>P&lt;sub&gt;gen&lt;/sub&gt; [kW]</th>
<th>P&lt;sub&gt;con&lt;/sub&gt; [kW]</th>
<th>P&lt;sub&gt;net&lt;/sub&gt; [kW]</th>
<th>P&lt;sub&gt;sav&lt;/sub&gt; [%]</th>
<th>Fuel&lt;sub&gt;sav&lt;/sub&gt; [kg]</th>
<th>CO2&lt;sub&gt;sav&lt;/sub&gt; [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>70.12</td>
<td>1.99</td>
<td>51.10</td>
<td>1.45</td>
<td>27.12</td>
<td>79.54</td>
</tr>
<tr>
<td>Apr.</td>
<td>61.96</td>
<td>2.29</td>
<td>44.75</td>
<td>1.27</td>
<td>31.29</td>
<td>91.77</td>
</tr>
<tr>
<td>May.</td>
<td>39.66</td>
<td>3.29</td>
<td>27.27</td>
<td>0.78</td>
<td>13.39</td>
<td>39.27</td>
</tr>
<tr>
<td>Jun.</td>
<td>213.56</td>
<td>2.86</td>
<td>158.02</td>
<td>4.50</td>
<td>99.07</td>
<td>290.55</td>
</tr>
<tr>
<td>Jul.</td>
<td>82.42</td>
<td>2.47</td>
<td>59.96</td>
<td>1.71</td>
<td>40.73</td>
<td>119.46</td>
</tr>
<tr>
<td>Aug.</td>
<td>183.33</td>
<td>2.77</td>
<td>135.41</td>
<td>3.85</td>
<td>87.04</td>
<td>255.26</td>
</tr>
<tr>
<td>Sep.</td>
<td>78.43</td>
<td>3.28</td>
<td>56.36</td>
<td>1.60</td>
<td>32.10</td>
<td>94.14</td>
</tr>
<tr>
<td>Oct.</td>
<td>249.86</td>
<td>2.33</td>
<td>185.64</td>
<td>5.28</td>
<td>130.86</td>
<td>383.77</td>
</tr>
<tr>
<td>Nov.</td>
<td>323.88</td>
<td>1.64</td>
<td>241.68</td>
<td>6.88</td>
<td>183.30</td>
<td>537.58</td>
</tr>
</tbody>
</table>
Figure 27. Lake Erie Power Contributions

Table 23. Lake Erie - Mean Results by Route

<table>
<thead>
<tr>
<th>Route</th>
<th>$P_{\text{gen}}$ [kW]</th>
<th>$P_{\text{con}}$ [kW]</th>
<th>$P_{\text{net}}$ [kW]</th>
<th>$P_{\text{sav}}$ [%]</th>
<th>Fuel$_{\text{sav}}$ [kg]</th>
<th>CO2$_{\text{sav}}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>105.75</td>
<td>1.92</td>
<td>77.87</td>
<td>2.22</td>
<td>101.68</td>
<td>298.19</td>
</tr>
<tr>
<td>2</td>
<td>89.96</td>
<td>1.76</td>
<td>66.15</td>
<td>1.88</td>
<td>84.72</td>
<td>248.45</td>
</tr>
<tr>
<td>3</td>
<td>144.80</td>
<td>2.55</td>
<td>106.69</td>
<td>3.04</td>
<td>71.66</td>
<td>210.15</td>
</tr>
</tbody>
</table>

Lake Ontario

Table 24. Lake Ontario - Route 1

<table>
<thead>
<tr>
<th>Month</th>
<th>$P_{\text{gen}}$ [kW]</th>
<th>$P_{\text{con}}$ [kW]</th>
<th>$P_{\text{net}}$ [kW]</th>
<th>$P_{\text{sav}}$ [%]</th>
<th>Fuel$_{\text{sav}}$ [kg]</th>
<th>CO2$_{\text{sav}}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>223.14</td>
<td>1.88</td>
<td>165.94</td>
<td>4.72</td>
<td>176.29</td>
<td>516.99</td>
</tr>
<tr>
<td>Apr.</td>
<td>151.79</td>
<td>1.78</td>
<td>112.51</td>
<td>3.20</td>
<td>119.86</td>
<td>351.50</td>
</tr>
<tr>
<td>May.</td>
<td>61.90</td>
<td>1.91</td>
<td>45.00</td>
<td>1.28</td>
<td>49.02</td>
<td>143.77</td>
</tr>
<tr>
<td>Jun.</td>
<td>85.99</td>
<td>1.17</td>
<td>63.62</td>
<td>1.81</td>
<td>72.17</td>
<td>211.66</td>
</tr>
<tr>
<td>Jul.</td>
<td>61.89</td>
<td>1.55</td>
<td>45.26</td>
<td>1.29</td>
<td>49.41</td>
<td>144.90</td>
</tr>
<tr>
<td>Aug.</td>
<td>55.63</td>
<td>1.43</td>
<td>40.65</td>
<td>1.16</td>
<td>46.04</td>
<td>135.01</td>
</tr>
<tr>
<td>Sep.</td>
<td>52.76</td>
<td>2.05</td>
<td>38.03</td>
<td>1.08</td>
<td>41.74</td>
<td>122.42</td>
</tr>
<tr>
<td>Month</td>
<td>$P_{\text{gen}}$ [kW]</td>
<td>$P_{\text{con}}$ [kW]</td>
<td>$P_{\text{net}}$ [kW]</td>
<td>$P_{\text{sav}}$ [%]</td>
<td>$\text{Fuel}_{\text{sav}}$ [kg]</td>
<td>$\text{CO}<em>2</em>{\text{sav}}$ [kg]</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Oct.</td>
<td>64.89</td>
<td>1.59</td>
<td>47.47</td>
<td>1.35</td>
<td>52.99</td>
<td>155.39</td>
</tr>
<tr>
<td>Nov.</td>
<td>114.50</td>
<td>0.66</td>
<td>85.37</td>
<td>2.43</td>
<td>91.02</td>
<td>266.93</td>
</tr>
</tbody>
</table>

### Table 25. Lake Ontario - Route 2

<table>
<thead>
<tr>
<th>Month</th>
<th>$P_{\text{gen}}$ [kW]</th>
<th>$P_{\text{con}}$ [kW]</th>
<th>$P_{\text{net}}$ [kW]</th>
<th>$P_{\text{sav}}$ [%]</th>
<th>$\text{Fuel}_{\text{sav}}$ [kg]</th>
<th>$\text{CO}<em>2</em>{\text{sav}}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>183.19</td>
<td>1.76</td>
<td>136.07</td>
<td>3.87</td>
<td>167.77</td>
<td>492.03</td>
</tr>
<tr>
<td>Apr.</td>
<td>130.08</td>
<td>1.76</td>
<td>96.23</td>
<td>2.74</td>
<td>114.51</td>
<td>335.81</td>
</tr>
<tr>
<td>May.</td>
<td>54.28</td>
<td>2.02</td>
<td>39.19</td>
<td>1.12</td>
<td>46.64</td>
<td>136.78</td>
</tr>
<tr>
<td>Jun.</td>
<td>89.04</td>
<td>1.32</td>
<td>65.79</td>
<td>1.87</td>
<td>73.62</td>
<td>215.91</td>
</tr>
<tr>
<td>Jul.</td>
<td>48.40</td>
<td>1.65</td>
<td>35.06</td>
<td>1.00</td>
<td>44.88</td>
<td>131.63</td>
</tr>
<tr>
<td>Aug.</td>
<td>60.93</td>
<td>1.56</td>
<td>44.53</td>
<td>1.27</td>
<td>46.80</td>
<td>137.24</td>
</tr>
<tr>
<td>Sep.</td>
<td>58.90</td>
<td>2.16</td>
<td>42.55</td>
<td>1.21</td>
<td>44.47</td>
<td>130.43</td>
</tr>
<tr>
<td>Oct.</td>
<td>73.12</td>
<td>1.68</td>
<td>53.58</td>
<td>1.52</td>
<td>55.11</td>
<td>161.62</td>
</tr>
<tr>
<td>Nov.</td>
<td>104.98</td>
<td>0.69</td>
<td>78.22</td>
<td>2.23</td>
<td>86.44</td>
<td>253.50</td>
</tr>
</tbody>
</table>

### Table 26. Lake Ontario - Route 3

<table>
<thead>
<tr>
<th>Month</th>
<th>$P_{\text{gen}}$ [kW]</th>
<th>$P_{\text{con}}$ [kW]</th>
<th>$P_{\text{net}}$ [kW]</th>
<th>$P_{\text{sav}}$ [%]</th>
<th>$\text{Fuel}_{\text{sav}}$ [kg]</th>
<th>$\text{CO}<em>2</em>{\text{sav}}$ [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar.</td>
<td>185.43</td>
<td>1.65</td>
<td>137.84</td>
<td>3.92</td>
<td>146.36</td>
<td>429.24</td>
</tr>
<tr>
<td>Apr.</td>
<td>117.61</td>
<td>1.72</td>
<td>86.91</td>
<td>2.47</td>
<td>100.87</td>
<td>295.82</td>
</tr>
<tr>
<td>May.</td>
<td>68.83</td>
<td>2.03</td>
<td>50.10</td>
<td>1.43</td>
<td>55.74</td>
<td>163.48</td>
</tr>
<tr>
<td>Jun.</td>
<td>119.07</td>
<td>1.26</td>
<td>88.36</td>
<td>2.51</td>
<td>78.92</td>
<td>231.45</td>
</tr>
<tr>
<td>Jul.</td>
<td>92.93</td>
<td>1.50</td>
<td>68.57</td>
<td>1.95</td>
<td>66.81</td>
<td>195.94</td>
</tr>
<tr>
<td>Aug.</td>
<td>100.28</td>
<td>1.47</td>
<td>74.11</td>
<td>2.11</td>
<td>63.84</td>
<td>187.23</td>
</tr>
<tr>
<td>Sep.</td>
<td>79.98</td>
<td>2.14</td>
<td>58.38</td>
<td>1.66</td>
<td>46.80</td>
<td>137.26</td>
</tr>
<tr>
<td>Oct.</td>
<td>75.85</td>
<td>1.71</td>
<td>55.60</td>
<td>1.58</td>
<td>51.39</td>
<td>150.71</td>
</tr>
<tr>
<td>Nov.</td>
<td>104.24</td>
<td>0.55</td>
<td>77.76</td>
<td>2.21</td>
<td>93.38</td>
<td>273.86</td>
</tr>
</tbody>
</table>
In a departure from the trends of all other results, Lake Ontario features higher savings in the Spring as compared to the Fall. This is likely due to the fact that Lake Ontario is the Easternmost lake and experiences different prevailing wind conditions than the other lakes.

**Figure 28. Lake Ontario Power Contributions**

**Table 27. Lake Ontario - Mean Results by Route**

<table>
<thead>
<tr>
<th>Route</th>
<th>P(_{\text{gen}}) [kW]</th>
<th>P(_{\text{con}}) [kW]</th>
<th>P(_{\text{net}}) [kW]</th>
<th>P(_{\text{sav}}) [%]</th>
<th>Fuel(_{\text{sav}}) [kg]</th>
<th>CO2(_{\text{sav}}) [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>96.94</td>
<td>1.56</td>
<td>71.54</td>
<td>2.04</td>
<td>77.61</td>
<td>227.62</td>
</tr>
<tr>
<td>2</td>
<td>89.21</td>
<td>1.62</td>
<td>65.69</td>
<td>1.87</td>
<td>75.58</td>
<td>221.66</td>
</tr>
<tr>
<td>3</td>
<td>104.91</td>
<td>1.56</td>
<td>77.52</td>
<td>2.21</td>
<td>78.24</td>
<td>229.44</td>
</tr>
</tbody>
</table>
Conclusion

Summary
The results presented in the above section shows that, for all lakes and routes modeled, the average savings produced by a single Flettner rotor is 2.69%. This suggests that, while still effective, Flettner rotors are less efficient on the Great Lakes when compared to ocean routes. This is emphasized by the results of other similar studies. Traut et al found that a single rotor produced 193 kW to 373 kW, a range higher than the results above (but a larger, 35-meter-tall rotor is used in that study). Traut reported propulsive contributions of 5% per rotor on a bulk carrier on a route over 9,000 km long, double that found above. Elsewhere Comer et al found fuel savings per rotor of 3.33% while Lu and Ringsberg reported 6.5% savings over a route between Brazil and the Netherlands. One thing that can be highlighted from all case studies is the influence of the route can not be overstated. This is unsurprising since the route determines the wind heading experienced by the rotor.

The results of this study are not unexpected. The Great Lakes are far smaller bodies of water then the oceans used in other studies and so smaller wind values were expected (and therefore a lesser rotor performance). This does not mean that Flettner rotors are not a good investment in the region. A ship operating primarily between the Soo Locks and the Welland Canal could see 20% fuel savings in a four-rotor system.

Future Work
Future work on the subject should focus on obtaining more accurate wind data. Instead of relying on a reanalysis database, obtaining information directly from ship-board transponders would provide the ability to preform a real-time analysis of rotor performance. Additionally, it could provide a more accurate route then can be obtained by the means utilized in this report. Specific
vessels could also be modeled by obtaining true resistance and powering values, providing a stronger base on which to evaluate results. Finally, preforming a stability and seakeeping analysis on vessels with rotors installed, particularly in non-symmetric arrangements, could provide valuable insight into potentially harmful side effects of adding rotors.
References


Appendix

1. Algoma Equinox Ship Profile (AlgomaEquinox.py)

```python
## Ship profile input for Algoma Equinox
## gcbond, 022822
##
## Setup ------------
##
## # Importing
## import numpy as np
##
## # Base file name
## iobase = 'Algoma Equinox'
##
## # Ship Data -------------
## Lpp = 222.06
## # m, length between perpendiculars
## Lws = 225.55
## # m, length over waterline
## Los = 225.55
## # m, length over wetted surface
## B = 23.77
## # m, beam
## T = 10.31
## # m, draft
## Taf = 10.31
## # m, draft at aft perp
## Twd = 10.31
## # m, draft at fwd perp
## Vdisp = 47952
## # m^3, displaced volume
## CB = Vdisp / (Lpp*B*T)
## # block coefficient
## SBK = 58.0
## # m^2, wetted surface area of bilge keels
## SAPP = SBK # + other appendages
## # m^2, area of appendages
## equifac = 1 + .4 # + other appendage factors
## # appendage equivalency factor
## AWS = 74.9
## # m^2, transverse area above waterline
## ABT = 0
## # m^2, bulbous bow cross section area at FP
## dTH = 1.58
## # m, diameter of bow thruster tunnel
## D = 6.693
## # m, propeller diameter
## vkn = np.array([13.7])
## # kn, ship service speed
##
## # Water Data --------------
## nueS = 1.1386e-6
## # m^2/s, freshwater kinematic at 15deg C
## rhoS = 999.1026
## # kg/m^3, freshwater density at 15deg C
## rhoair = 1.225
## # kg/m^3, density of air
## CDA = 0.8
## # air drag coefficient
## grav = 9.807
## # m/s^2, acceleration due to gravity
##
## # END ---------------------
```
2. Lake Michigan Input File (LakeMichigan.py)

```python
# Lake Michigan Input File (LakeMichigan.py)

##
## Lake Michigan
## gc=pond, 630322
##

name = 'Lake Michigan'
zoom = 7.35

## Importing
import numpy as np

## Routes
lake lat = []
lake lon = []

# empty bins for storing routes
lat c = 44.6
lon c = -86.7

# centering points for map

# Route 1 - Chicago to Mackinac Strait
lat1 = np.array([41.7, 42.8, 42.3, 42.6, 42.9, 43.2, 43.5, 43.8, 44.1, \ 
                 44.4, 44.7, 44.9, 45.6, 45.2, 45.5, 45.8, 45.8, 45.8])
lon1 = np.array([-87.5, -87.4, -87.3, -87.2, -87.1, -87.0, -86.9, -86.8, \ 
                 -86.7, -86.6, -86.4, -86.2, -86.0, -85.7, -85.4, -85.2, \ 
                 -84.9, -84.7])

lake lat.append(lat1)
lake lon.append(lon1)

# Route 2 - Marinette to Mackinac Strait
lat2 = np.array([45.1, 45.2, 45.3, 45.4, 45.5, 45.5, 45.6, 45.7, \ 
                 45.8, 45.8, 45.9, 45.9, 45.8, 45.8, 45.8])
lon2 = np.array([-87.6, -87.5, -87.3, -87.2, -87.1, -86.9, -86.7, -86.5, \ 
                 -86.3, -86.1, -85.9, -85.7, -85.5, -85.2, -84.9, -84.7])

lake lat.append(lat2)
lake lon.append(lon2)

# Route 3 - Chicago to Mackinac Strait (ALTERNATE ROUTE)
lat3 = np.array([41.7, 42.8, 42.3, 42.6, 42.9, 43.2, 43.5, 43.8, 44.1, \ 
                 44.4, 44.7, 44.9, 45.1, 45.3, 45.5, 45.7, 45.9, 45.9, 45.8, \ 
                 45.8, 45.8])
lon3 = np.array([-87.5, -87.4, -87.3, -87.2, -87.1, -87.0, -86.9, -86.8, \ 
                 -86.7, -86.6, -86.4, -86.3, -86.2, -86.1, -86.0, -85.9, \ 
                 -85.7, -85.5, -85.2, -84.9, -84.7])

lake lat.append(lat3)
lake lon.append(lon3)

# Route 4 - Milwaukee to Mackinac Strait
lat4 = np.array([43.0, 43.2, 43.4, 43.6, 43.8, 44.0, 44.2, 44.4, \ 
                 44.7, 44.9, 45.0, 45.2, 45.5, 45.8, 45.8, 45.8])
lon4 = np.array([-87.9, -87.8, -87.6, -87.4, -87.2, -87.0, -86.8, -86.6, \ 
                 -86.4, -86.2, -86.0, -85.7, -85.4, -85.2, -84.9, -84.7])

lake lat.append(lat4)
lake lon.append(lon4)

# Route 5 - Chicago to Milwaukee
lat5 = np.array([41.7, 42.8, 42.3, 42.6, 42.9, 43.0])
lon5 = np.array([-87.5, -87.6, -87.7, -87.7, -87.7, -87.9])

lake lat.append(lat5)
lake lon.append(lon5)
```

- 1 -
3. Lake Superior Input File (LakeSuperior.py)

```python
## Lake Superior
## gcbbnd. 031522

name = 'Lake Superior'
zoom = 6.8

## Importing
import numpy as np

## Routes

## empty bins for storing routes
lat c = 47.8
lon c = -89.3

## centering points for map

# Route 1 - Duluth to Soo Locks
lat1 = np.array([46.8, 46.8, 46.9, 47.0, 47.1, 47.2, 47.3, 47.4, 47.5, 47.6, 
                 46.5])
lon1 = np.array([-91.1, -91.6, -91.4, -91.2, -90.6, -90.0, -89.4, 
                 -88.8, -88.2, -87.8, -87.4, -87.1, -86.8, -86.5, -86.2, 
                 -85.9, -85.6, -84.9, -84.8, -84.6])

## Route 2 - Marquette to Soo Locks
lat2 = np.array([46.5, 46.6, 46.8, 46.9, 46.8, 46.6, 46.6])
lon2 = np.array([-87.4, -87.0, -86.6, -86.0, -85.6, -84.9, -84.8])

## Route 3 - Thunder Bay to Soo Locks
lat3 = np.array([48.4, 48.3, 48.2, 48.1, 48.0, 47.9, 47.8, 47.7, 47.6, 47.5, 
                 47.6, 47.3, 47.2, 47.1, 46.9, 46.8])
lon3 = np.array([-89.2, -89.0, -88.7, -88.3, -88.0, -87.7, -87.4, -87.1, 
                 -86.8, -86.5, -86.3, -86.1, -85.9, -85.7, -85.5, -85.3, 
                 -85.1, -84.9])

##
```

- 1 -
4. Lake Huron Input File (LakeHuron.py)

```python
# Lake Huron
# gbond, 030322
#

name = 'Lake Huron'
zoom = 7.7

# Importing
import numpy as np

# Routes
lake lat = []
lake lon = []
# empty bins for storing routes
lat c = 44.5
lon c = -83.0
# centering points for map

# Route 1 - Soo Locks to Sarnia
lat1 = np.array([46.0, 45.8, 45.6, 45.4, 45.3, 45.2, 45.0, 44.8, 44.6, 44.4, 44.2, 44.0, 43.5, 43.0])
lon1 = np.array([-83.9, -83.8, -83.7, -83.5, -83.3, -83.1, -83.0, -82.9, -82.8, -82.7, -82.6, -82.5, -82.4])
lake lat.append(lat1)
lake lon.append(lon1)

# Route 2 - Mackinac Strait to Sarnia
lat2 = np.array([45.8, 45.7, 45.6, 45.5, 45.4, 45.3, 45.2, 45.0, 44.8, 44.6, 44.4, 44.2, 44.0, 43.5, 43.0])
lon2 = np.array([-84.7, -84.5, -84.3, -84.0, -83.7, -83.5, -83.3, -83.2, -83.1, -83.0, -82.9, -82.8, -82.7, -82.6, -82.5, -82.4])
lake lat.append(lat2)
lake lon.append(lon2)

# Route 3 - Maitland River to Sarnia
lat3 = np.array([43.7, 43.6, 43.5, 43.4, 43.3, 43.2, 43.1, 43.0])
lon3 = np.array([-81.7, -81.8, -81.9, -82.0, -82.1, -82.2, -82.3, -82.4])
lake lat.append(lat3)
lake lon.append(lon3)
```

-1-
5. Lake Erie Input File (LakeErie.py)

```python
# Lake Erie
# gc54, 030322

name = 'Lake Erie'
zoom = 7.70

# Importing
import numpy as np

# Routes
lake lat = []
lake lon = []
# empty bins for storing routes
lat c = 42.2
lon c = -81.2
    # centering points for map

# Route 1 - Detroit River to Welland Canal
lat1 = np.array([42.1, 42.0, 41.9, 41.9, 41.9, 41.8, 41.9, 42.0, 42.1, 
                 42.2, 42.3, 42.5, 42.6, 42.7, 42.8])
lon1 = np.array([-83.1, -83.1, -83.0, -82.9, -82.8, -82.7, -82.5, -82.1, 
                 -81.7, -81.3, -80.9, -80.5, -80.1, -79.7, -79.4, -79.1])
lake lat.append(lat1)
lake lon.append(lon1)

# Route 2 - Toledo to Welland Canal
lat2 = np.array([41.7, 41.8, 41.9, 41.9, 41.9, 41.8, 41.9, 42.0, 42.1, 
                 42.2, 42.3, 42.5, 42.6, 42.7, 42.8])
lon2 = np.array([-83.4, -83.2, -83.0, -82.9, -82.8, -82.7, -82.5, -82.1, 
                 -81.7, -81.3, -80.9, -80.5, -80.1, -79.7, -79.4, -79.1])
lake lat.append(lat2)
lake lon.append(lon2)

# Route 3 - Detroit River to Cleveland
lat3 = np.array([42.1, 42.0, 41.9, 41.9, 41.9, 41.8, 41.9, 41.7, 41.6, 41.5])
lon3 = np.array([-83.1, -83.1, -83.0, -82.9, -82.8, -82.7, -82.5, -82.2, 
                 -82.0, -81.7])
lake lat.append(lat3)
lake lon.append(lon3)
```

- 1 -
6. Lake Ontario Input File (LakeOntario.py)

```python
##
## Lake Ontario
## gcbond, 030322
##
name = 'Lake Ontario'
zoom = 7.8

## Importing
import numpy as np

## Routes
lake lat = []
lake lon = []
    # empty bins for storing routes
lat c = 43.6
lon c = -78.0
    # centering points for map

# Route 1 - Welland Canal to St. Marys River
lat1 = np.array([ 43.2, 43.3, 43.4, 43.5, 43.6, 43.7, 43.8, 44.0, 44.1])
lon1 = np.array([-79.3, -79.2, -78.6, -78.0, -77.4, -76.8, -76.6, -76.5, -76.4])
lake lat.append(lat1)
lake lon.append(lon1)

# Route 2 - Hamilton to St. Marys River
lat2 = np.array([ 43.3, 43.3, 43.3, 43.4, 43.5, 43.6, 43.7, 43.8, 44.0, 44.1])
lon2 = np.array([-79.8, -79.6, -79.4, -79.2, -78.6, -78.0, -77.4, -76.8, -76.6, -76.5, -76.4])
lake lat.append(lat2)
lake lon.append(lon2)

# Route 3 - Toronto to St. Marys River
lat3 = np.array([ 43.6, 43.7, 43.7, 43.7, 43.7, 43.7, 43.7, 43.7, 43.8, 44.0, 44.1])
lon3 = np.array([-77.4, -79.0, -78.7, -78.4, -78.1, -77.8, -77.5, -77.2, -76.8, -76.6, -76.5, -76.4])
lake lat.append(lat3)
lake lon.append(lon3)
```
7. Run File (RunFile.py)

```python
##
## Run File
## gcbond, 030322
##
## Input Files -----------------------------------------------
## Ship Profile
from AlgomaEquinox import *
# other ships
## Lake
from LakeErie import *
# Lake Michigan, Lake Superior, Lake Huron, Lake Ontario, Lake Erie
## Wind Data
file = 'WindData.nc'

## Parameters -----------------------------------------------
## Rotor Parameters
N = 1
# number of rotors
dia = 5
# m, diameter of rotors
h = 30
# m, height of rotors
A = np.pi * dia * h * N
# m^2, surface area of rotor(s)
SR = 5
# spin ratio
## Constants
Cd = 0.2
# drag coefficient
Cl = 12.5
# lift coefficient
rho = 1.225
# kg/m^3, density of air
mu = 1.81 * 10**-5
# kg/m^s, dynamic viscosity of air
etaP = 0.75
# propulsive efficiency
fc = 288
# g/kWh, MDO fuel consumption
cO2 = 610
# g/kWh, MDO CO2 emissions

## Running Scripts -------------------------------------------
from RoutePlanning import RoutePlot
from dataextract import GetWind
from hollenbach import PowerEstimate
from spinnythingcalc5 import Calc
from RosePlots import RosePlot
from PostProcessing import Outputs
RoutePlot( lake lat, lake lon, lat c, lon c, name, zoom )
lake u10 m, lake v10 m, t month = GetWind( lake lat, lake lon, file )
PE, vS = PowerEstimate( ioibase, Lpp, Lws, Los, B, T, Taft, Tfwd, Vdisp, CB, SBK, 
SAPP, equfac, AVS, ABT, dTH, D, vkn, nueS, rhoS, rhoair, 
CDA, grav, A )
P gen mm, P con mm, P net mm, P sav mm, f sav mm, c sav mm, 
P gen mm, P con mm, P net mm, P sav mm, f sav mm, c sav mm 
= Calc( N, dia, h, A, SR, Cd, Cl, rho, mu, etaP, fc, cO2, vS, PE, 
lake u10 m, lake v10 m, t month, lake lat, lake lon )
RosePlot( vkn, rho, A, Cl, Cd, N, h, dia, ioibase )
Outputs( P_gen_mm, P_con_mm, P_net_mm, P_sav_mm, f_sav_mm, c_sav_mm, 
...)```
P gen rm, P con rm, P net rm, P sav rm, f sav rm, c sav rm, N, dia, h, \nSR, C, Cl, rho, mu, etaP, fc, co2, VS, PE, iobase, name, vkn )
8. Route Planning Module (RoutePlanning.py)

```python
## Route Planning Module
gcband, 62212822

## Importing
import plotly.graph_objects as go
import numpy as np

def RoutePlot( lake lat, lake lon, lat c, lon c, name, zoom):
    """
    Description:
    A function to plot routes for a lake given sets of latitudes and longitudes.

    Inputs:
    lake lat = A list containing arrays of route latitudes
    lake lon = A list containing arrays of route longitudes
    lat c = A latitude point to center the map upon
    lon c = A longitude point to center the map upon
    name = A string of the lake name

    Returns:
    A .png file of the generated routes
    """

    routes plot = go.Figure()
    ## Defining Traces
    for i in range( len(lake lat)-1, -1, -1):
        routes plot.add_trace(go.Scattermapbox(
            mode = "markers+lines",
            lon = lake lon[i],
            lat = lake lat[i],
            marker = {"size": 10},
            name = "Route {i}".format(i=i+1)))

    ## Formatting Map
    routes plot.update_layout(
        margin ={"l":0,"t":0,"b":0,"r":0},
        mapbox = {
            'center': {"lon": lon c, "lat": lat c},
            'style': "carto-positron",
            'zoom': zoom})

    ## Saving File
    plot = '{i} Routes'.format(name)
    routes plot.write_image('{}{}.png'.format(plot), width=1500, height=1500)
    print('Plot saved to {}.png'.format(plot))

## END
```
9. Data Extraction Module (dataextract.py)

```python
## Data Extraction Module
## gcbond, 022722

## Importing
from netCDF4 import Dataset
import numpy as np
from netCDF4 import num2date
import datetime
from LakeMichigan import *

## Data File
file = 'WindData.nc'
Data = Dataset(file, mode='r', format='NETCDF4')
print(Data)

## Data Extraction

time = Data.variables['time']
latitude = Data.variables['latitude']
longitude = Data.variables['longitude']
v10wnd = Data.variables['v10']
u10wnd = Data.variables['u10']

# print(time)
# print(lat)
# print(lon)
# print(v10)
# print(u10)

# used for checking dimension

t = np.array(Data.variables['time'])
lat = np.array(Data.variables['latitude'])
lon = np.array(Data.variables['longitude'])
v10 = np.array(Data.variables['v10'])
u10 = np.array(Data.variables['u10'])

# converting to numpy arrays

## Converting and Indexing Time

tconv = num2date(t, units=time.units, only_use_cftime_datetimes=False)

cmpths = np.array(tconv)

mar = []
apr = []
may = []
jun = []
ju1 = []
aug = []
sept = []
oct = []
nov = []

# empty bins for months

for i in range(0, len(months)):
    datem = datetime.datetime.strptime(str(months[i]), "%Y-%m-%d %H:%M:%S")
    if datem.month==3:
        mar.append(i)
    elif datem.month==4:
        apr.append(i)
    elif datem.month==5:
        may.append(i)
    elif datem.month==6:
        jun.append(i)
    elif datem.month==7:
        # empty bins for months
```

-1-
```python
jul.append(i)
elif datom.month==8:
    aug.append(i)
elif datom.month==9:
    sep.append(i)
elif datom.month==10:
    oct.append(i)
elif datom.month==11:
    nov.append(i)
else:
    print('FAILURE in binning months')
    break
# binning months
t month = [mar, apr, may, jun, jul, aug, sep, octo, nov]

## Indexing Lat and Lon ------------------------------------------
lat i = np.zeros( (len(lake lat)), dtype=object)
lon i = np.zeros( (len(lake lat)), dtype=object)
# bins for routes
lake u10 = np.zeros( (len(lake lat)), dtype=object)
lake v10 = np.zeros( (len(lake lat)), dtype=object)
# bins for wind values
for j in range( 0, len(lake lat)):
    lat bin = np.zeros( (len(lake lat[j])))
    lon bin = np.zeros( (len(lake lat[j])))
    lake u10[j] = np.zeros( (len(lake lat[j]), \
    len(t month), len(t month[0]) ) )
    lake v10[j] = np.zeros( (len(lake lat[j]), \
    len(t month), len(t month[0]) ) )
lat i[j] = lat bin
lon i[j] = lon bin
# bins for route indexes and wind values
for j in range( 0, len(lake lat)):
    for i in range(0, len(lake lat[j])):
        lat[i][j][0] = np.where(lat == lake lat[j][i])[0]
        lon[i][j][0] = np.where(lon == lake lon[j][i])[0]
        # retrieving lat, lon indexes for routes

## Retrieving Wind Data ----------------------------------------
for i in range( 0, len(lake u10) )
    for k in range( 0, len(t month[k])):
        for l in range( 0, len(lake u10[i])):
            lake u10[i][j][k][l] = u10[t month[k][l]][lat i[i][j]] \
            [lon i[i][j]]
        lake v10[i][j][k][l] = v10[t month[k][l]][lat i[i][j]] \
            [lon i[i][j]]
lake u10 = lake u10.tolist()
lake v10 = lake v10.tolist()
lake u10 m = []
lake v10 m = []
for i in range( 0, len(lake u10)):
    m u = lake u10[i].mean(axis=2)
    m v = lake v10[i].mean(axis=2)
    lake u10 m.append(m u)
    lake v10 m.append(m v)

# END ----------------------------------------------------------
```


```python
## Hollenbach's Method for Resistance
## gcbond, 030622
##
## Importing -----------------------------
import numpy as np
import matplotlib.pyplot as plt

##

def PowerEstimate(iobase, Lpp, Lws, Los, B, T, Taft, T fwd, V disp, CB, SBK, 
SAPP, equifac, AVS, ABT, dTH, D, vkn, nueS, rhoS, rhoair, 
CDA, grav, A):

    ""
    Description:
    An implementation of Hollenbach's Method for Resistance to obtain an
    estimate for the effective power required for a given ship profile.
    Inputs:
    iobase = base file name, refers to ship profile
    Lpp = length between perpendiculars in meters
    Lws = length over waterline in meters
    Los = length over wetted surface in meters
    B = beam in meters
    T = draft in meters
    Taft = draft at aft perpendicular in meters
    T fwd = draft at fwd perpendicular in meters
    V disp = displaced volume in m^3
    CB = block coefficient
    SBK = wetted surface of bilge keels in m^2
    SAPP = surface area of appendages in m^2
    equifac = appendage equivalency factor
    AVS = transverse area above waterline in m^2
    ABT = bulbous bow cross sectional area at forward perpendicular in m^2
    dTH = diameter of bow thruster tunnel in m
    D = propeller diameter in m
    vkn = ship service speed in kn
    nueS = freshwater kinematic viscosity at 15 degC in m^2/s
    rhoS = density of freshwater at 15 degC in kg/m^3
    rhoair = density of air in kg/m^3
    CDA = air drag coefficient for ship
    grav = gravity in m^2/s
    A = surface area of rotors in m^2
    Returns:
    PE = Effective power required to move the vessel at the given service speed.
    vs = ship service speed in m/s.
    ""

    ## Setup ---------------------------------
    # Computational Length
    if Los < Lpp:
        Lc = Los
    elif Los > 1.1*Lpp:
        Lc = 1.8677*Lpp
    else:
        Lc = Lpp + 2.*T *(Los-Lpp)
    # Wetted Surface Area
    s = np.array([-0.6837, 0.2771, 0.6542, 0.6422, 0.6075, 0.0275, 
                  -1, ...]
```
\begin{verbatim}
62 k = s[3]*lpp/lws + s[2]*lws/lpp + s[3]*CB + s[4]*lpp/b + \ 63 s[5]*B/T + s[6]*lpp/T + s[7]*(Taft-Tfwd)/lpp + s[8]*D/T 64 S = k*lpp*(B+2*T) # m^2, surface area 65 66 # Reynolds number 68 vS = vkm*l052./3600. # converting km to m/s 69 Re = vS*Lc/nueS 70 71 # Froude number 72 Fr = vS/ np.sqrt(grav*Lc) 73 74 ## Frictional Resistance Coefficient 75 CF = 0.075 / (np.log10(Re)-2)**2 76 77 ## Residualy Resistance Estimate 78 # Coefficient Values 79 if CB < 0.49: 80    blinean = -0.87674 81 elif 0.49 <= CB < 0.6: 82    blinean = -0.57424 - 25 * (0.6 - CB)**2 83 else: 84    blinean = -0.57424 85 bmean = np.array([ blinean, 13.3893, 90.596, 4.6614, -39.721, -351.483, 86 -1.14215, -12.3296, 459.254 ])
87 bmin = bmean.copy() 88 bmin[6] = -0.91424 89 dmean = np.array([ 0.054, -1.228, 0.497 ])
90 dmin = np.array([ 0, 0, 0, 0 ])
91 amean = np.array([ 0.3302, -0.8006, -6.0256, -3.5632, 9.4405, 92 0.8146, 0, 0, 0, 0 ])
96 emean = np.array([ 2.1701, -0.1682 ])
97 emin = np.array([ 1., 0. ])
110 111 # Speed independent k factors 112 # High Froude number factor 113 kFmean = np.where( Fr < Frcritmean, 1., (Fr / Frcritmean )**2*(Fr / Frcritmin )) 114 kFmin =1. # This is done to avoid dividing by zero (Frcritmin = 0)
115 116 # Length factor 117 klmean = emean[0]*lpp**emin[1]
118 klmin = emin[0]*lpp**emin[1]
\end{verbatim}
# Beam-draft ratio factor
if B/T < 1.99:
    kBTmean = 1.99 * amean[0]
else:
    kBTmean = (B/T)**amean[0]
    if B/T < 1.99:
        kBTmin = 1.99 * amin[0]
    else:
        kBTmin = (B/T)**amin[0]

# Length-beam ratio factor
if Lpp/B <= 7.11:
    kLBmean = (Lpp/B)**amean[1]
else:
    kLBmean = 7.11**amean[1]
    if Lpp/B <= 7.11:
        kLBmin = (Lpp/B)**amin[1]
    else:
        kLBmin = 7.11**amin[1]

# Wetted length ratio factor
if Los/Lws <= 1.05:
    kLLmean = (Los/Lws)**amean[2]
else:
    kLLmin = 1.05**amean[2]
    if Los/Lws <= 1.05:
        kLLmin = (Los/Lws)**amin[2]
    else:
        kLLmin = 1.05**amin[2]

# Aft overhang factor
if Lws/Lpp <= 1.06:
    kAOmean = (Lws/Lpp)**amean[3]
else:
    kAOmean = 1.05**amean[3]
    if Lws/Lpp <= 1.06:
        kAOmin = (Lws/Lpp)**amin[3]
    else:
        kAOmin = 1.05**amin[3]

# Trim correction factor
kTrmean = (1 + (Taft-Tfwd)/Lpp)**amean[4]
kTrmin = (1 + (Taft-Tfwd)/Lpp)**amin[4]

# Propeller factor
if D/Taft < 0.43:
    kPmean = 0.43**amean[5]
elif 0.43 <= D/Taft <= 0.84:
    kPmean = (D/Taft)**amean[5]
else:
    kPmean = 0.84**amean[5]
    if D/Taft < 0.43:
        kPmin = 0.43**amin[5]
    elif 0.43 <= D/Taft <= 0.84:
        kPmin = (D/Taft)**amin[5]
    else:
        kPmin = 0.84**amin[5]

## Resistances
CRBmean = CRstdmean*kFmean*kLmean*kBTmean*kLBmean*kLLmean*kAOmean \
- 3 -
CRBTmin = CRstdmin*kFmin*kLmin*kTmin*kBmin*kLLmin*kAOmin*kTrmin*kPrmin

# Residual Resistance Coefficient
CRmean = CRBTmean*B*T / (10*S)
CRmin = CRBTmin*B*T / (10*S)

# Correlation Allowance
if Lpp < 175:
    CA = (0.35 - 0.002*Lpp)*10**-3
else:
    CA = 0

# Appendage Resistance
RAPP = 0.5*rhoS**2*S APP*equifac*CF
CDTH = 0.003 + 0.003*(10*dTH/T - 1)
RTH = rhoS**2*np.pi*dTH**2*CDTH
CAPP = (RAPP + RTH) / (0.5*rhoS**2*S)

# Environmental Resistance
CAAS = (CDA*rhoair*(AVS + A)) / (rhoS*S)
Conv = CAAS # for trial condition

## Total Resistance
CTmean = CF + CRmean + CA + CAPP + Conv
CTmin = CF + CRmin + CA + CAPP + Conv
CTmax = 1.204*CTmean
RTmean = 0.5*rhoS**2*S*CTmean
RTmin = 0.5*rhoS**2*S*CTmin
RTmax = 0.5*rhoS**2*S*CTmax

## Mean Effective Power
PE = vS*RTmean
return(PE, vS)

## END--
11. Rotor Calculations (spinnythingcalcs_2.py)

```python
## Calculating the propulsive contribution of Flettner rotor
## gcbond, 022722
#
## Importing
import numpy as np
import matplotlib.pyplot as plt
import sys

def Calc(N, dia, h, A, SR, Cd, Cl, rho, mu, etaP, fc, co2, vS, PE, 
         lake u10 m, lake v10 m, t month, lake lat, lake lon):
    """
    Description:
    Calculates the propulsive power contributions of Flettner rotor(s) at
    points along a specified route.
    Inputs:
    N = number of rotors
    dia = rotor diameter in m
    h = rotor height in m
    A = surface area of rotors in m^2
    SR = rotor spin ratio
    Cd = rotor drag coefficient
    Cl = rotor lift coefficient
    rho = density of air in kg/m^3
    mu = dynamic viscosity of air in kg/m*s
    etaP = propulsive efficiency
    fc = fuel consumed for MDO in g/kWh
    co2 = CO2 emissions for MDO in g/kWh
    vS = ship service speed in m/s
    PE = Effective power required to move the vessel at the given
    service speed
    lake u10 m = A list containing arrays of u10 wind speeds broken down by
    route and month
    lake v10 m = A list containing arrays of v10 wind speeds broken down by
    route and month
    t month = A list containing arrays of the indexes of months for all
    years covered by the data file.
    lake lat = A list containing arrays of route latitudes
    lake lon = A list containing arrays of route longitudes
    Returns:
    P gen mm = monthly mean generated power by route in kW
    P con mm = monthly mean consumed power by route in kW
    P net mm = monthly mean net power by route in kW
    P sav mm = monthly mean propulsive contribution by route in %
    f sav mm = monthly mean fuel savings by route in kg
    c sav mm = monthly mean CO2 savings by route in kg
    P gen rm = route mean generated power in kW
    P con rm = route mean consumed power in kW
    P net rm = route mean net power in kW
    P sav rm = route mean propulsive contribution in %
    f sav rm = route mean fuel savings in kg
    c sav rm = route mean CO2 savings in kg
    """

## Empty Bins
P gen = []
P con = []
```
```python
P net = []
P sav = []
f sav = []
c sav = []
# values by lat, lon
for i in range(0, len(lake u10 m)): 
P gen b = np.zeros(len(t month), len(lake lat[i])-1)
P con b = np.zeros(len(t month), len(lake lat[i])-1)
P sav b = np.zeros(len(t month), len(lake lat[i])-1)
f sav b = np.zeros(len(t month), len(lake lat[i])-1)
c sav b = np.zeros(len(t month), len(lake lat[i])-1)
P gen.append(P gen b)
P con.append(P con b)
P sav.append(P sav b)
f sav.append(f sav b)
c sav.append(c sav b)
P gen mm = []
P con mm = []
P sav mm = []
f sav mm = []
c sav mm = []
# route monthly means
P gen rm = []
P con rm = []
P sav rm = []
f sav rm = []
c sav rm = []
# route means

# Calculations
for i in range(0, len(lake u10 m)): 
    for j in range(0, len(lake lat[i])-1):
        # Finding Ship Heading
        x = np.cos(np.radians(lake lat[i][k+1])) 
* np.sin(np.radians(lake lon[i][k+1]-lake lon[i][k]))
        y = np.cos(np.radians(lake lat[i][k])) 
* np.sin(np.radians(lake lon[i][k])) 
* np.cos(np.radians(lake lat[i][k+1])) 
* np.cos(np.radians(lake lon[i][k+1]-lake lon[i][k]))
        beta = np.arctan2(x, y)
        # heading angle measured with 0 deg North
        # Finding Distance Between Points on Earth
        R = 6371
        # km, radius of the Earth
        d = 2*R*np.arccos(np.sqrt(np.cos(np.radians(lake lat[i][k+1]-lake lat[i][k])/2)**2 
+ np.cos(np.radians(lake lon[i][k]))
* np.cos(np.radians(lake lon[i][k+1])) 
* np.sin(np.radians(lake lon[i][k+1]-lake lon[i][k])/2)**2))
        # km, distance between points using Haversine formula
        # Travel Time
        vs = vs[i]
        travel_t = d*1060 / vs / 3600
        -2.
```
# Find Wind Heading and Magnitude
wspd = np.sqrt(lake u10 m[i][k][j]**2 + lake v10 m[i][k][j]**2)
theta = np.arctan2(lake u10 m[i][k][j], lake v10 m[i][k][j])

# theta measured with 0 deg North
# Finding True Wind
u comp = -vs*np.sin(beta) + lake u10 m[i][k][j]
v comp = -vs*np.cos(beta) + lake v10 m[i][k][j]
true wnd = np.sqrt(u comp**2 + v comp**2)
kappa = np.arctan2(u comp, v comp)
alpha = np.abs(beta - kappa)
if np.degrees(alpha) > 180.:
    alpha = np.radians(360. - np.degrees(alpha))
else:
    alpha = alpha

# angle correction if necessary

# Thrust Calculations
L = 0.5 * rho * A * true wnd**2 * Cl / 1000
D = 0.5 * rho * A * true wnd**2 * Cd / 1000

# kW
FT = L*np.sin(alpha) - D*np.cos(alpha)

# Power Calculations
P gen p = FT * vs * N
P gen[i][j][k] = P gen p

# kW, rotor generated power
Re = rho * SR * true wnd * np.pi*dia / mu
# Reynolds number
Cf = 0.455/(np.log(Re)**2.58 - 1700/Re)

# friction coefficient
Ff = 0.5 * rho * A * (SR*true wnd)**2 * Cf / 1000

# frictional force
P con p = Ff * (SR*true wnd) * N
P con[i][j][k] = P con p

# kW, power consumed by rotor
P net p = (P gen p - P con p) * N * etaP

# kW, net power
if P net p < 0:
    P net p = 0
else:
    P net p = P net p
P save p = P net p / (PE/1000) * 100
P net[i][j][k] = P net p
P sav[i][j][k] = P sav p

# Fuel Savings
f sav p = fc * P net p * travel t / 1000
f sav[i][j][k] = f sav p

# kg, fuel saved
c sav p = co2 * P net p * travel t / 1000

# kg, CO2 emissions saved

## Mean Values
for i in range(0, len(lake u10 m)):
    mm gen = P gen[i].mean(axis=1)
    mm con = P con[i].mean(axis=1)
    mm net = P net[i].mean(axis=1)
    mm sav = P sav[i].mean(axis=1)
    m fsav = f sav[i].sum(axis=1)
    m_csav = c sav[i].sum(axis=1)
P gen mm.append(mm gen)
P con mm.append(mm con)
P net mm.append(mm net)
P sav mm.append(mm sav)
f sav mm.append(m fsav)
c sav mm.append(m csav)

# monthly means/sums
rm gen = P gen[i].mean()
rm con = P con[i].mean()
rm net = P net[i].mean()
rm sav = P sav[i].mean()
rm fsav = f sav[i].sum(axis=1).mean()
rm csav = c sav[i].sum(axis=1).mean()
P gen rm.append(rm gen)
P con rm.append(rm con)
P net rm.append(rm net)
P sav rm.append(rm sav)
f sav rm.append(rm fsav)
c sav rm.append(rm csav)

return( P gen mm, P con mm, P net mm, P sav mm, f sav mm, c sav mm, \ P gen rm, P con rm, P net rm, P sav rm, f sav rm, c sav rm )

# END
def RosePlot(vkn, rho, A, Cl, Cd, N, h, dia, iobase):
    
    Description:
    A function to create a wind rose plot of a Flettner rotor given a ship profile.

    Inputs:
    vkn = ship service speed in kn
    rho = density of air in kg/m^3
    A = surface area of rotors in m^2
    Cl = rotor lift coefficient
    Cd = rotor drag coefficient
    N = number of rotors
    h = rotor height in m
    dia = rotor diameter in m
    iobase = base file name, refers to ship profile

    Returns:
    A .png file of the generated rose plot.

    # Setup
    P gen z = []
    # empty bin for generated power
    vs = vkn[0] * 1852 / 3600
    # extracting service speed

    # Wind
    sigma = np.linspace(0, np.pi, 1000)
    # true wind angle
    wind = np.arange(0.5, 3, 0.5)
    # m/s, wind speed

    # Calculations
    for i in range(0, len(wind)):
        u = -vs + wind[i] * np.cos(sigma)
        v = wind[i] * np.sin(sigma)
        va = np.sqrt(u**2 + v**2)
        gamma = np.arctan2(v, u)
        delta = np.abs(gamma)
        L = 0.5 * rho * A * vs**2 * Cl / 1000
        D = 0.5 * rho * A * vs**2 * Cd / 1000
        FT = L * np.sin(delta) - D * np.cos(delta)
        P gen = FT * vs * N
        P gen z.append(P gen)

    # Plotting
    plt.figure(figsize=(10,6))
    -1-
```python
ax = plt.subplot(11, polar=True)
ax.set_theta_zero_location('E')
ax.set_xticks(np.linspace(0, 180, 345))
ax.set_thetamin(0)
ax.set_thetamax(180)
ax.plot(sigma, P, label='red', label=r'$v$ T$=\text{m/s}$'.format(wind[0]))
ax.plot(sigma, P, label='blue', label=r'$v$ T$=\text{m/s}$'.format(wind[1]))
ax.plot(sigma, P, label='green', label=r'$v$ T$=\text{m/s}$'.format(wind[2]))
ax.plot(sigma, P, label='purple', label=r'$v$ T$=\text{m/s}$'.format(wind[3]))
ax.plot(sigma, P, label='orange', label=r'$v$ T$=\text{m/s}$'.format(wind[4]))
angle = np.deg2rad(57.5)
ax.legend(loc='lower left', bbox_to_anchor=(-.5 + np.cos(angle) / 2, .1 + np.sin(angle) / 2))
ax.set_title('Propulsive Power Generated by a Flettner Rotor \n' r'\{ with $V \ s=\text{kn} \ n=\text{m} \ d=\text{m} \'.format(lobas, vkn[0], h, d))
label_position = ax.get_label_position()
ax.text(np.math.radians(label_position-40), (ax.get_rmax()+10)/2, 'kW', 
rotation=0, ha='center', va='center')
ax.text(np.math.radians(label_position+1000), (ax.get_rmax()+1000)/2, 
'True Wind Angle', 
rotation=0, ha='center', va='center')
ax.text(np.math.radians(label_position-300), (ax.get_rmax()+1000)/2, 
'Relative To Ship Heading', 
rotation=0, ha='center', va='center')

# Saving
plt.savefig('{} Rose Plot.png'.format(lobas))
print('Plot saved to {}'.format(lobas))
```

13. Post Processing Module (PostProcessing.py)

```python
## Post-Processing
## gbond, 030622
#
import matplotlib.pyplot as plt
import numpy as np

def Outputs( P gen mm, P con mm, P net mm, P sav mm, f sav mm, c sav mm, \
              P gen rm, P con rm, P net rm, P sav rm, f sav rm, c sav rm, N, dia, h, \ 
              SR, Cd, Cl, rho, mu, etaP, fc, co2, vs, PE, iobase, name, vkn ):
    """
    Description:
    A function to create plots of results and output file containing results
    for the given lake and ship profile.
    
    Inputs:
    P gen mm = monthly mean generated power by route in kW
    P con mm = monthly mean consumed power by route in kW
    P net mm = monthly mean net power by route in kW
    P sav mm = monthly mean propulsive contribution by route in %
    f sav mm = monthly mean fuel savings by route in kg
    c sav mm = monthly mean CO2 savings by route in kg
    P gen rm = route mean generated power in kW
    P con rm = route mean consumed power in kW
    P net rm = route mean net power in kW
    P sav rm = route mean propulsive contribution in %
    f sav rm = route mean fuel savings in kg
    c sav rm = route mean CO2 savings in kg
    N = number of rotors
    dia = rotor diameter in m
    h = rotor height in m
    A = surface area of rotors in m^2
    SR = rotor spin ratio
    Cd = rotor drag coefficient
    Cl = rotor lift coefficient
    rho = density of air in kg/m^3
    mu = dynamic viscosity of air in kg/m*s
    etaP = propulsive efficiency
    fc = fuel consumed for MDO in g/kWh
    co2 = CO2 emissions for MDO in g/kWh
    vs = ship service speed in m/s
    PE = Effective power required to move the vessel at the given service speed
    iobase = base file name, refers to ship profile
    name = A string of the lake name
    vkn = ship service speed in kn
    
    Returns:
    A single .png plot of the P sav mm value of each route
    A single .png plot of the f sav mm and c sav mm of each route
    A .txt file containing the inputs and results of the run
    """
    
    ## Plotting
    # Power Plot
    plt.figure(figsize=(18,6))
    months = np.arange(1, 10, 1)
```
```python
month = ['Mar.', 'Apr.', 'May.', 'Jun.', 'Jul.', 'Aug.', 'Sep.', 'Oct.', 'Nov.]
for i in range(0, len(month)):
    plt.plot(months, P, s='o', marker='o')
plt.grid()
plt.ylabel('Propulsive Power Contribution Per Rotor [%]')
plt.xlabel('Month')
plt.legend()
plt.title('Propulsive Power Contribution Per Rotor by Route for: \n' + \n    'on {} with $h$={}, $d$={}, and $s$={} km\n'.format(i_base, name, dia, vkm[0]))
plt.xticks(ticks=months, labels=['Mar.', 'Apr.', 'May', 'Jun.', 'Jul.', 'Aug.', 'Sep.', 'Oct.', 'Nov.'])
plt.savefig('propulsive_power_contribution.png', format='png', name, i_base)

plot1 = []
for i in range(0, len(f)):
    ax1.set_title('Propulsive Power Contribution Per Rotor by Route for: \n' + \n    'on {} with $h$={}, $d$={}, and $s$={} km\n'.format(i_base, name, dia, vkm[0]))
    ax1.plot(months, f, s='o', marker='o')
    ax1.set_xlabel('Month')
    ax1.legend()
    for i in range(0, len(f)):
        bx1.plot(months, c, s='o')
        bx1.set_xlabel('Month')
        bx1.legend()
        fig.savefig('savings_plot.png', format='png', name, i_base)
        print('Plot saved to SavingsPlot.png')
        plt.show()
```

```
fp.write('MDQ Specific CO2 Energy \[8.2f\] g/kWh\n'.format(co2))
fp.write('Ship Service Speed \[8.2f\] kn\n'.format(vkn[0]))
fp.write('Effective Power \[8.2f\] kW\n'.format(PEm[0]/1000))
fp.write('\n')
fp.write('\n')
fp.write('Mean Results by Month:\n')
fp.write('\n')
fp.write('\n')
for i in range(0, len(P_gen)):  
    fp.write('Route {}\n'.format(i+1))
    fp.write('\n')
    fp.write('P gen'.rjust(8))
    fp.write('P con'.rjust(8))
    fp.write('P net'.rjust(10))
    fp.write('P sav'.rjust(8))
    fp.write('f sav'.rjust(10))
    fp.write('c sav'.rjust(11))
    fp.write('KW'.rjust(3))
    fp.write('kw'.rjust(9))
    fp.write('KW'.rjust(8))
    fp.write('kw'.rjust(10))
    fp.write('%'.rjust(7))
    fp.write('kg'.rjust(11))
    fp.write('kg'.rjust(11))
    fp.write('\n')
    fp.write('\n')
for j in range(0, len(months)):
    fp.write('month[\n]
    fp.write('P gen'.rjust(9))
    fp.write('P con'.rjust(9))
    fp.write('P net'.rjust(10))
    fp.write('P sav'.rjust(8))
    fp.write('f sav'.rjust(10))
    fp.write('c sav'.rjust(11))
    fp.write('\n')
    fp.write('\n')
fp.write('Mean Results by Route:\n')
fp.write('\n')
fp.write('\n')
for i in range(0, len(P_gen)):  
    fp.write('\n')
    fp.write('P gen'.rjust(7))
    fp.write('P con'.rjust(7))
    fp.write('P net'.rjust(10))
    fp.write('P sav'.rjust(8))
    fp.write('f sav'.rjust(10))
    fp.write('c sav'.rjust(11))
    fp.write('\n')
    fp.write('\n')
fp.write('['.format(i+1)).rjust(2))
fp.write('{{:.2f}}'.format(P gen rm[i]).rjust(10))
fp.write('{{:.2f}}'.format(P con rm[i]).rjust(8))
fp.write('{{:.2f}}'.format(P net rm[i]).rjust(10))
fp.write('{{:.2f}}'.format(P sav rm[i]).rjust(8))
fp.write('{{:.2f}}'.format(f sav rm[i]).rjust(10))
fp.write('{{:.2f}}'.format(c sav rm[i]).rjust(10))
fp.write('
')
fp.close()
print('Output file saved to {}'\'.format(outputfile))
## END ------------------------------------------------------------------------------------------------------