

HARP_Opt User's Guide

by Danny Sale

Introduction

The HARP_Opt (Horizontal Axis Rotor Performance Optimization) code utilizes a multiple objective genetic algorithm and blade-element momentum (BEM) theory flow model to design horizontal-axis wind and hydrokinetic turbine rotors. Genetic algorithms solve optimization problems by mimicking the principles of biological evolution. Using rules modeled on biological reproduction and gene modification, genetic algorithms repeatedly modify a population of individuals to create subsequent generations of "superior" individuals. HARP_Opt utilizes the MATLAB Genetic Algorithm solver to perform this optimization, and the WT_Perf BEM theory code to predict rotor performance metrics.

HARP_Opt optimizes a rotor's performance for steady and uniform flows (no sheared or yawed flows). A variety of rotor control configurations can be designed using HARP_Opt, including fixed or variable rotor speed and fixed or variable blade pitch operation. Blades with circular or non-circular roots can be designed using HARP_Opt.

HARP_Opt can function as a single or multiple objective optimization code. The primary optimization objective is to maximize the turbine's annual energy production (AEP). Annual energy production is calculated using a Rayleigh, Weibull, or user-defined flow distribution. Maximum power is bounded, and maximum power point tracking (MPPT) is a combined objective with AEP. For hydrokinetic turbines, additional constraints are defined such that cavitation will not occur. An additional objective can be activated, in which HARP_Opt performs a structural analysis to minimize the blade mass. For the structural analysis, the blade is modeled as a thin shell of bulk isotropic material, and the blade mass is minimized using a maximum allowable strain as the constraint. Maximizing energy production and minimizing blade mass are conflicting objectives, thus HARP_Opt will identify the set of Pareto optimal solutions. To meet these objectives, HARP_Opt calculates the optimal blade shape (twist, chord, and airfoil/hydrofoil distributions) and optimal control of the rotor speed and blade pitch.

The HARP_Opt code is designed to be user friendly, and is operated using a graphical user interface (GUI) and easy to manipulate text input files. Output from the HARP_Opt code is summarized in automatically generated Excel spreadsheets and text files.

The primary purpose of this document is to guide the user on how to run the HARP_Opt code. Detailed explanations of the underlying theory and algorithms within HARP_Opt are currently a work in progress, but will eventually be included in this document.

Before using this code, it is recommended that users become familiar with the WT_Perf code, available at:

wind.nrel.gov/designcodes/simulators/wtperf/

Installing HARP_Opt

Currently, HARP_Opt is only supported for Windows operating systems. If you do not own MATLAB you can use the compiled version of HARP_Opt. To run the compiled version of HARP_Opt, you will need to first install the MATLAB Component Runtime (MCR). The MCR is a standalone set of shared libraries that enable the execution of MATLAB scripts on computers without an installed version of MATLAB and associated licensing. The MCR is free and only needs to be installed once. To run the compiled version of HARP_Opt v2.00.00, you will need to install the MCR v7.9, which is available for download at:

wind.nrel.gov/designcodes/simulators/HARP_Opt

The GUI of HARP_Opt requires a minimum screen resolution of 1024 x 768, but a resolution of 1280 x 1024 or greater is recommended. In addition, the input and output capabilities of HARP_Opt require Microsoft Excel to be installed. HARP_Opt may not work on older processors which do not support SSE2 instructions.

Distributed Files

The folder structure and distributed files are described in this section.

HARP_Opt.exe

The compiled standalone executable for HARP_Opt. Execute this file within a Windows environment to run the code.

HARP_Opt.pdf

This user's guide in PDF format.

Change_Log.txt

The list of changes to HARP_Opt for the various releases.

Input_Files\Airfoil_Data\

This folder holds text files which contain data for aerodynamic coefficients (*.dat files) and x-y coordinate profiles (*.prof files).

Input_Files\Flow_Data\

This folder holds text files which contain data for flow probability distributions (*.dat files).

Input_Files\GUI_Presets\

This folder holds MATLAB figure files (*.fig) which allow the saving & loading of presets for the HARP_Opt user interface.

Input_Files\Templates\

The files in this folder are used by HARP_Opt and should not need to be edited by the user. The only exception is the Excel file **Output_Template.xls**, which can be edited by the user to change the format (graph colors, font size, etc.) of the Excel output file.

Output_Files\

Each time HARP_Opt is executed, a new folder is created which stores the following files and folder structure:

*_Input.bmp

A screenshot of the user interface created when the **Begin Optimization** button is pressed. This image shows the input parameters which produced the output files.

*_Output.xls

The Excel output file containing various graphs and data. Use this file to visualize the HARP_Opt output data.

*.wtp

WT_Perf input file(s) associated to the optimal solutions created by HARP_Opt.

*.oup

WT_Perf output file(s)—contain rotor performance data. These files are associated to the optimal solutions found by HARP_Opt.

*.bed

WT_Perf blade-element output file(s)—contain performance data of the individual blade elements. These files are associated to the optimal solutions found by HARP_Opt.

\WT_Perf.exe

A copy of the WT_Perf executable.

*_Initial_Population.dat

Contains the values of the design variables for the initial population of the genetic algorithm.

\Airfoil_Data\

This folder contains the airfoil data files used in the particular HARP_Opt run.

*.dat

These are the airfoil files containing 2D aerodynamic coefficients.

\3D_Airfoil_Data\

This folder only exists when a **Stall Delay** model has been used.

*.dat

These are the airfoil files containing corrected 3D aerodynamic coefficients; these files use a different naming convention than the 2D airfoil files (explained in later sections)

\Coordinates\

This folder only exists when **Structural Optimization** is selected.

*.prof

These files contain x-y coordinates of the normalized airfoil profiles, & have been interpolated from the airfoil data defined in **\Input_Files\Airfoil_Data*.prof**

*.mat

This MATLAB data file contains all of the data in the ***.prof** files, and can be imported into MATLAB.

\Failed_Cases\

This folder exists only when the **Record Failed Cases** button is selected. When a failed case occurs, the corresponding WT_Perf *.wtp, *.oup, and *.bed files are copied to this folder. These files will contain a prefix containing the letters "o", "b", "c", or "s." These letters notate the type of error which occurred:

"o": an error detected in the *.oup file

"b": an error detected in the *.bed file

"c": cavitation was detected

"s": circular root blade shape constraints were violated

Source_Code\

This folder holds the MATLAB source code files for HARP_Opt. The source code for v2.00.00 is dependent on the "MATLAB Genetic Algorithm & Direct Search Toolbox" and "MATLAB Statistics Toolbox." Note: The source files must be executed from the root directory in order to recognize the folder structure. HARP_Opt.m is the main function.

Using HARP_Opt

The majority of the input parameters are input directly within the HARP_Opt GUI, the exceptions being the text files which contain airfoil data and flow probability data (described in later sections). The GUI contains definitions and appropriate units next to each input field to guide the user. HARP_Opt uses the metric system for all user inputs and output data.

Turbine Configuration

In the **Speed Control** and **Pitch Control** button groups, the user can select the turbine's control configuration. HARP_Opt models six control configurations:

1. Fixed-Speed Fixed-Pitch (Stall Regulated)

The rotor speed and blade pitch remain constant over all flow speeds. The shape of the blade and selection of rotor speed cause stall to occur, which regulates power output.

2. Fixed-Speed Variable-Pitch (Pitch to Feather)

The rotor speed remains constant over all flow speeds, but once the rated power is achieved the blades begin pitching towards feather (decreasing the angle-of-attack)—which produces constant power output.

3. Variable-Speed Fixed-Pitch (Stall Regulated)

The rotor operates at its optimal tip speed ratio (maximum efficiency), until either the minimum or maximum rotor speeds are met, or until rated power is achieved. Once rated power is achieved, the rotor speed will begin to decrease—causing the blades to stall and maintain constant power output. HARP_Opt will try to minimize this reduction of rotor speed in order to minimize the resulting increase in torque. HARP_Opt attempts for the reduction in power to be a result mostly of the blade stalling due to its shape, as opposed to stall induced by a reduction in rotor speed.

4. Variable-Speed Variable-Pitch (Pitch to Feather)

The rotor operates at its optimal tip speed ratio (maximum efficiency), until either the minimum or maximum rotor speeds are met, or until rated power is achieved. Once rated power is achieved, the rotor speed remains constant and the blades begin pitching towards feather (decreasing the angle-of-attack)—which produces constant power output.

5. Fixed-Speed Variable-Pitch (Pitch to Stall)

The rotor speed remains constant over all flow speeds, but once the rated power is achieved the blades begin pitching towards stall (increasing the angle-of-attack)—which produces constant power output.

6. Variable-Speed Variable-Pitch (Pitch to Stall)

The rotor operates at its optimal tip speed ratio (maximum efficiency), until either the minimum or maximum rotor speeds are met, or until rated power is achieved. Once rated power is achieved, the rotor speed remains constant and the blades begin pitching towards stall (increasing the angle-of-attack)—which produces constant power output.

NumBlade is the number of blades on the rotor. **NumSeg** tells WT_Perf how many elements there will be along the blade. The analysis points are located at the center of the elements.

Prated is the rated mechanical power of the rotor, not to be confused with the rated electrical power of the generator. The rated power is the maximum power that the rotor will produce.

RotorDia is the rotor diameter. Blade radius ($0.5 \times \text{RotorDia}$) is measured as the distance along the precone blade. If **PreCone** = 0, then blade radius is equal to the swept radius, but if **PreCone** is non-zero then blade radius is a number larger than the swept radius.

HubDia is the hub diameter. Note that radius is measured from the center of rotation (i.e. equal to 0 at center of rotation). **HubHt** is the distance measured from the center of rotation to the ground (or seabed).

PreCone should be defined as a positive value, regardless of whether the turbine is downwind or upwind. This is because precone blades only effect aerodynamics in HARP_Opt, the structural effects of precone are not modeled.

ShaftTilt is currently hard coded equal to zero. Ability to model shaft tilted rotors may come in future versions.

OmgMin is the minimum allowable rotor speed. **OmgMax** is the maximum allowable rotor speed. Whenever the **RotorDia** or **OmgMax** input fields are changed, the maximum possible tip speed is calculated and displayed within the GUI. The maximum possible tip speed is a useful number to be aware of since wind turbine aerodynamic noise increases by the fifth power of the tip speed. Also, for hydrokinetic turbines cavitation becomes more likely as the tip speed increases. HARP_Opt has a built in constraint to avoid cavitation, but the user should still carefully select the maximum possible tip speed—especially for variable speed hydrokinetic turbines.

Hydrokinetic Turbine Configuration

By selecting the **Hydrokinetic Turbine Configuration** checkbox, this tells the WT_Perf code to model the turbine as a hydrokinetic turbine, otherwise the turbine is modeled as a wind turbine. The difference between the HARP_Opt analysis of a wind and hydrokinetic turbine is that cavitation is checked for in the case of a hydrokinetic turbine. If modeling a hydrokinetic turbine, the user must also enter values for **Pvapor**, **Patm**, **WatDepth**, and **CavSF**. Also, the airfoil data files (*.dat) must contain values for the minimum pressure coefficients (C_{pmin}) of the hydrofoil.

Pvapor is the water vapor pressure (absolute). **Patm** is the air atmospheric pressure (absolute). **WatDepth** is the depth of the water.

CavSF is a safety factor to avoid cavitation. By setting **CavSF** to a number greater than 1, the distance between the rotor's operating point and the cavitation inception point becomes smaller. Thus, setting **CavSF** to a number greater than 1 requires that the optimization converge on blade shapes with a

higher resistance to cavitation. To avoid cavitation, the following condition must remain true (equation 1), where σ is the cavitation number and C_{pmin} is the minimum pressure coefficient of the hydrofoil:

$$\frac{\sigma}{C_{pmin}} + C_{pmin} \geq 0 \quad (1)$$

Fluid Properties

Rho is the fluid density. **KinVisc** is the fluid kinematic viscosity, which is used to calculate the Reynolds number.

SpdSt and **SpdEnd** define the minimum and maximum flow speeds that the turbine will be optimized over. **SpdDel** is the flow speed increment over this range of flow speeds. I recommend to choose values of **SpdDel** such that about 20-40 flow speeds are defined in this range.

WT_Perf Algorithm Configuration

The **TipLoss** and **HubLoss** flags tell the WT_Perf BEM analysis to use the Prandtl tip- and hub-loss algorithms. The **TipLoss** and **HubLoss** algorithms should both usually be enabled.

The **Swirl** flag tells the code to model a rotational wake, and thus enables the algorithm which calculates the tangential induction factor. **Swirl** should be set to true, unless for specific research purposes.

Setting the **AdvBrake** flag to true invokes the advanced brake-state algorithm instead of the classic momentum brake-state model. **AdvBrake** should be set to true, unless for specific research purposes.

Setting **IndProp** to true tells WT_Perf to use its traditional PROP-style induction algorithm. Setting **IndProp** to false invokes the PROPX induction algorithm, which is more similar to the AeroDyn induction algorithm.

To include the drag term in the axial induction algorithm, set **AIDrag** to true. To include the drag term in the tangential induction algorithm, set **TIDrag** to true.

MaxIter limits the number of iteration attempts in the WT_Perf BEM induction factor calculations. **ATol** tells WT_Perf how little you want the BEM induction factors to change from one iteration to the next in order to consider it converged.

Stall-Delay Models

One of the simplifying assumptions of BEM theory is that spanwise flow along the length of the blade is neglected. But in reality the presence of spanwise flow can cause significant augmentation to the lift, drag, and pressure distributions, especially near the blade root. HARP_Opt uses stall-delay models to make corrections to the lift and drag coefficients, accounting for spanwise flow. At present, no corrections are made to the minimum pressure coefficients because a suitable model is unavailable. In addition, the

stall-delay models are currently only usable for the fixed-speed control configurations—the HARP_Opt code will be updated in the future to allow the use of stall-delay models for all the available control configurations.

In the **Stall-Delay Models** button group, the user has the choice to use no stall-delay models, or to use the Selig-Du and/or Eggers stall-delay models. The drag coefficient behaves differently depending on which stall-delay model is used. Generally, the drag coefficient decreases due to the Selig-Du model, while the Eggers model calls for an increase of the drag coefficient.

If using a stall-delay model, the `\3D_Airfoil_Data\` folder is created in the output directory, which contains airfoil files created automatically by the stall-delay algorithm. The first 4-digit number corresponds to the non-dimensional radial station multiplied by 1000, and the second 4-digit number corresponds to the % thickness multiplied by 10. An example of the naming convention for stall-delay airfoil files is shown in Table 1:

Table 1: File naming convention when using stall-delay

FoilFam	Radial Station (r/R)	% Thick	Filename
S809	0.458	21.3	S809_0458_0213.dat

The current implementation of the stall-delay models apply some approximations. Because the stall-delay models can diverge at very inboard radial stations, any radial station inboard of 35% radial station uses the same corrections as the 35% radial station—this decreases the chances of the equations diverging. In addition, the average of the **SpdSt** and **SpdEnd** flow speeds is used in the Selig-Du stall-delay equations [1], the scaling parameters $a=b=d=1$ are hardcoded in, and the stall-delay corrections are only fully applied up to an angle-of-attack of 15 degrees. Hopefully, these stall-delay models can be calculated directly within WT_Perf in future versions; this will eliminate the need for approximations and provide a significant increase in computational speed.

Optimization Objectives

HARP_Opt can optimize either efficiency or annual energy production (AEP). If **Optimize annual energy production (AEP)** is selected, the user is required to select either a **Rayleigh**, **Weibull**, or **User-Defined Distribution** function to describe the flow speed distribution. If the **Rayleigh Distribution** is selected, the user must enter the long-term mean flow speed, **U**. If the **Weibull Distribution** is selected, the user must enter values for the shape factor, **k**, and the scale factor, **c**. Once the values for **k** and **c** are entered, the GUI calculates and displays the long-term mean flow speed for the Weibull distribution. If the **User Defined Distribution** option is selected, use the **Select File** button to select a data file from the `\Input_Files\Flow_Data\` folder. See the header of the example files in this folder for instructions on how to enter a user defined flow probability distribution. Using the selected flow distribution, HARP_Opt will maximize the AEP directly.

If **Optimize efficiency** is selected, HARP_Opt will attempt to maximize the turbines efficiency over the range of defined flow speeds. Optimizing a rotor for maximum efficiency does not necessarily create a turbine with the highest (AEP). Note that when **Optimize efficiency** is selected, the user can still select a **Flow Distribution** and AEP will be calculated in the output files, but realize that AEP was not actually optimized. More details on how HARP_Opt mathematically defines “optimal” will come in future versions of this document.

Structural Optimization

If the **Perform Structural Optimization** checkbox is selected, HARP_Opt becomes a multi-objective optimization code. The structural objective is to minimize the blade mass while satisfying a maximum allowable strain, and the energy objective is either efficiency or AEP depending on the user selection. When structural optimization is enabled, HARP_Opt requires airfoil files which store the x-y coordinates of the airfoil profiles (*.prof files) in addition to the aerodynamic coefficient data (*.dat files).

During structural optimization, the blade is modeled as a simple beam with bulk isotropic material properties. The blade cross section is modeled as a thin shell (no shear web) and strain is calculated at four strain gages as shown in the figure 1. Strain is calculated using equation 2, where the spanwise bending moments result from the maximum root bending moment experienced over the range of flow speeds. In equation 2, SF is a safety factor, M is the bending moment, c is the distance from neutral axis, E is the modulus of elasticity, I is the area moment of inertia, and the subscript r denotes a local radial value.

$$\epsilon_r = \frac{(SF)M_r c_r}{EI_r} \quad (2)$$

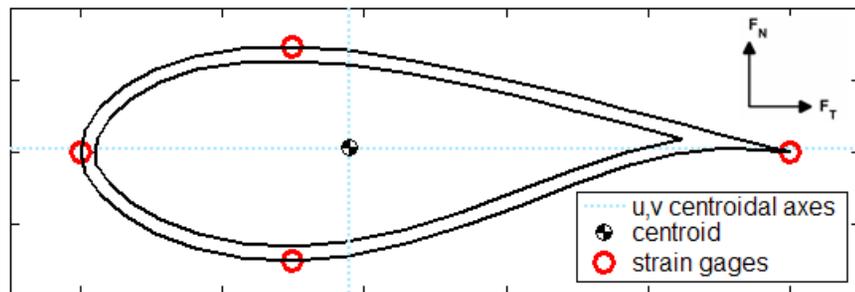


Figure 1: The blade is modeled as a thin shell, and strain is calculated at the four strain gages. Lift and drag forces are resolved into the u,v centroidal axes (which are orthogonal to the chordline) at each blade station. The shell thickness is varied until the strain at all four gages is less than the maximum allowable strain.

E is the modulus of elasticity for the bulk isotropic material, and **MatDensity** is the material's density.

MaxStrain is the maximum allowable strain. HARP_Opt will calculate the shell thickness along the length of the blade such that each blade station does not exceed the maximum allowable strain. **SF** is a safety factor, which is multiplied to the bending moments.

STmin is the minimum possible shell thickness, and **STdel** is the increment at which shell thickness can vary. Setting **DecST** to true will require the shell thickness to be monotonically decreasing from the hub to tip. The maximum allowable strain will still be satisfied for both settings of **DecST**.

Blade Geometry Configuration

HARP_Opt defines the shape of the blade through the use of curve fits. The twist, chord, and thickness distributions are all required to be monotonically decreasing. As figure 2 illustrates, five control points are defined for the twist and chord distributions, and then a 5th order Bezier curve is fit through the control points. The genetic algorithm determines the optimal twist and chord values of the control points, but the radial location of the control points remain fixed. The radial location of the control points are spaced using half-cosine spacing (with higher density towards the hub), and the radial values of the control points are displayed next to **CP radius** within the GUI.

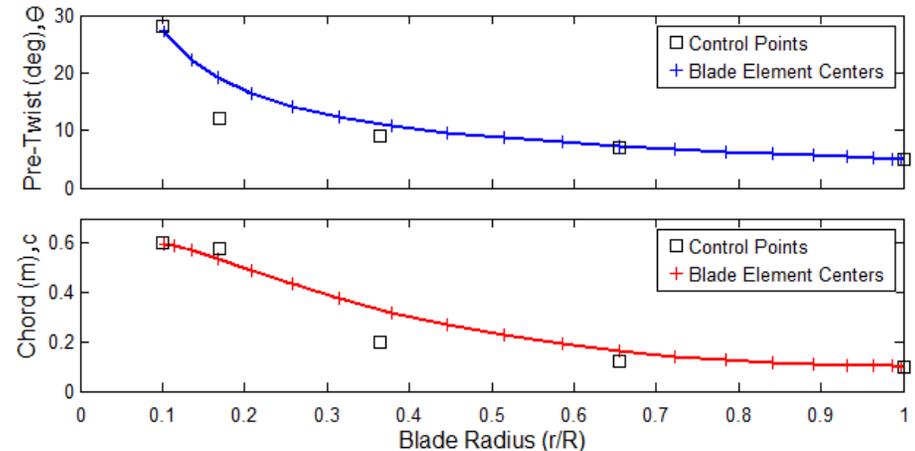


Figure 2: The radial location of the control points are at fixed locations. The optimal twist and chord values of the control points are determined by the genetic algorithm, and a 5th order Bezier curve is fit through the control points to define the intermediate blade stations.

Setting **ElmSpc** to "Equal" results in the blade element centers being equally spaced along the length of the blade. "Cosine" spacing results in a higher density of points (smaller blade segments) near the hub and tip of the blade, which is better suited for the hub and tip loss models.

The percent thickness (t/c) distribution defines the spanwise placement of airfoils. The value of the control points correspond to the percent thickness ($100 \cdot t/c$) values of the airfoils input by the user in the **ThickVals** input box. The genetic algorithm determines the optimal radial location of the control points for the percent thickness distribution.

HARP_Opt has two methods for modeling the percent thickness distribution, which can be selected via the **Piecewise Constant** and **Piecewise Linear** buttons in the **Airfoil/Hydrofoil Distribution** panel. Figure 3 illustrates the difference between the **Piecewise Constant** and **Piecewise Linear** options for modeling the percent thickness distribution. For the **Piecewise Constant** distribution, the percent thickness remains constant between the control points. For the **Piecewise Linear** distribution, the percent thickness varies linearly between the control points.

The **TwistLB**, **TwistUB**, **ChordLB**, and **ChordUB** input fields are used to enter lower and upper bounds for the control points controlling twist and chord, respectively. The (r/R) LB and (r/R) UB fields are used to bound the radial placement of airfoils. For example, if the user enters "30 24" in the **ThickVals** field and then "0.25 0.3" and "0.25 0.7" in the (r/R) LB and (r/R) UB fields, then the 30% thick airfoil will remain at $r/R=0.25$ and the 24% thick airfoil will remain between $r/R=0.3-0.7$. Note that it is valid to leave the (r/R) LB and (r/R) UB fields blank, entering no bounds for the airfoil locations; and it is also valid to enter "NaN" resulting in the genetic algorithm determining the appropriate upper/lower bounds automatically.

For the **Piecewise Constant** distribution, for N values entered into the **ThickVals** field, you need to input $N-1$ values into the (r/R) LB and $N-1$ values into the (r/R) UB fields. For the **Piecewise Linear** distribution, for N values entered into the **ThickVals** field, you need to input N values into the (r/R) LB and N values into the (r/R) UB fields. $N \geq 1$.

The **Piecewise Constant** model results in discontinuities in the dimensional thickness, as well as discontinuities in the aerodynamic coefficients along the length of the blade. The **Piecewise Linear** model is considered an improved model, but as figure 4 illustrates, many interpolated airfoils will exist along the span of the blade. This requires the aerodynamic coefficients (lift, drag, and pressure) to be interpolated using percent thickness as the variable. Shape Preserving Interpolation (SPI) has been developed for HARP_Opt as a possibly more realistic alternative to simple linear interpolation. Figures 5 and 6 shows the basic difference between linear interpolation and SPI. In this example of figures 5 and 6, the 24% and 30% thick airfoils are the only "true" airfoils (their lift and drag polars have been determined experimentally). SPI results in the C_{Lmax} moving along a straight line, while linear interpolation produces erratic behavior of C_{Lmax} . Furthermore, SPI appears to preserve the characteristic that drag will increase sharply at C_{Lmax} , while linear interpolation does not preserve this characteristic.

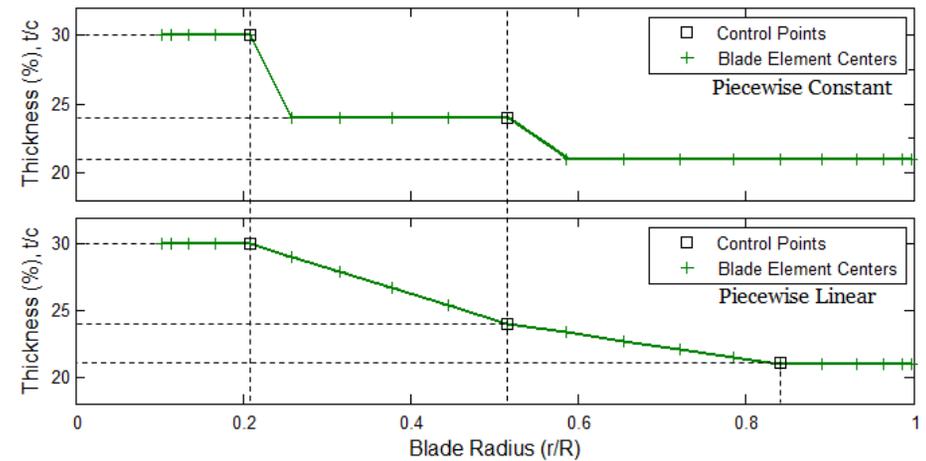


Figure 3: A comparison between the **Piecewise Constant** and **Piecewise Linear** options for modeling the percent thickness distribution. In this example, the user has 3 airfoils available, with percent thicknesses of 30, 24, and 21. Therefore, the user will enter "30 24 21" into the **ThickVals** input field. The genetic algorithm then determines the optimal radial placement of these 3 airfoils.

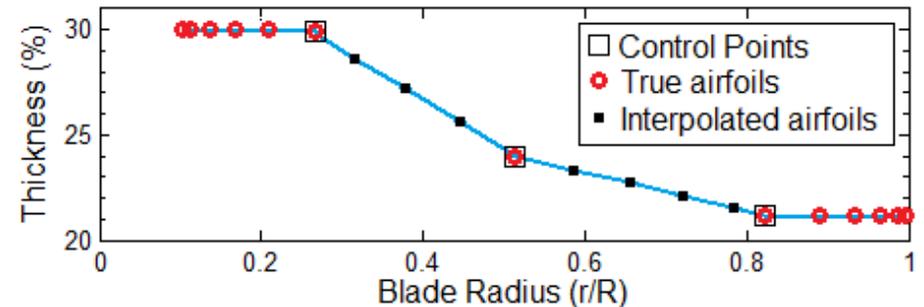


Figure 4: When using the Piecewise Linear model, SPI is used to determine aerodynamic coefficients of interpolated airfoils along the blade span.

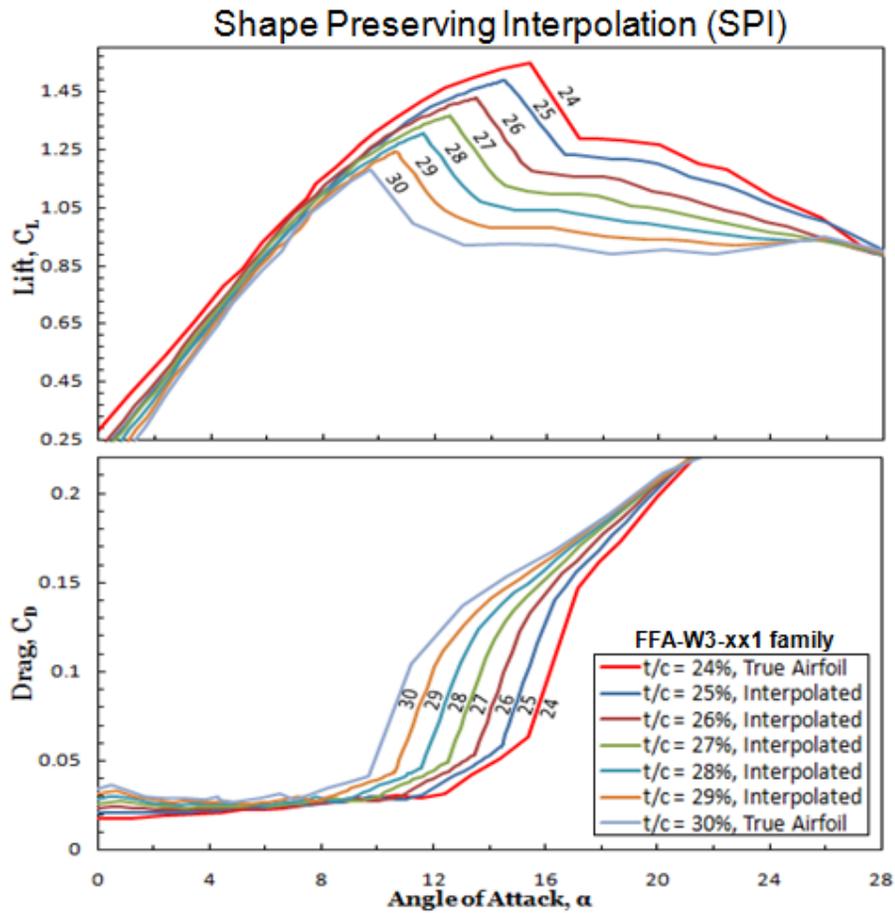


Figure 5: Interpolation of lift and drag coefficients with percent thickness as the variable using Shape Preserving Interpolation (SPI).

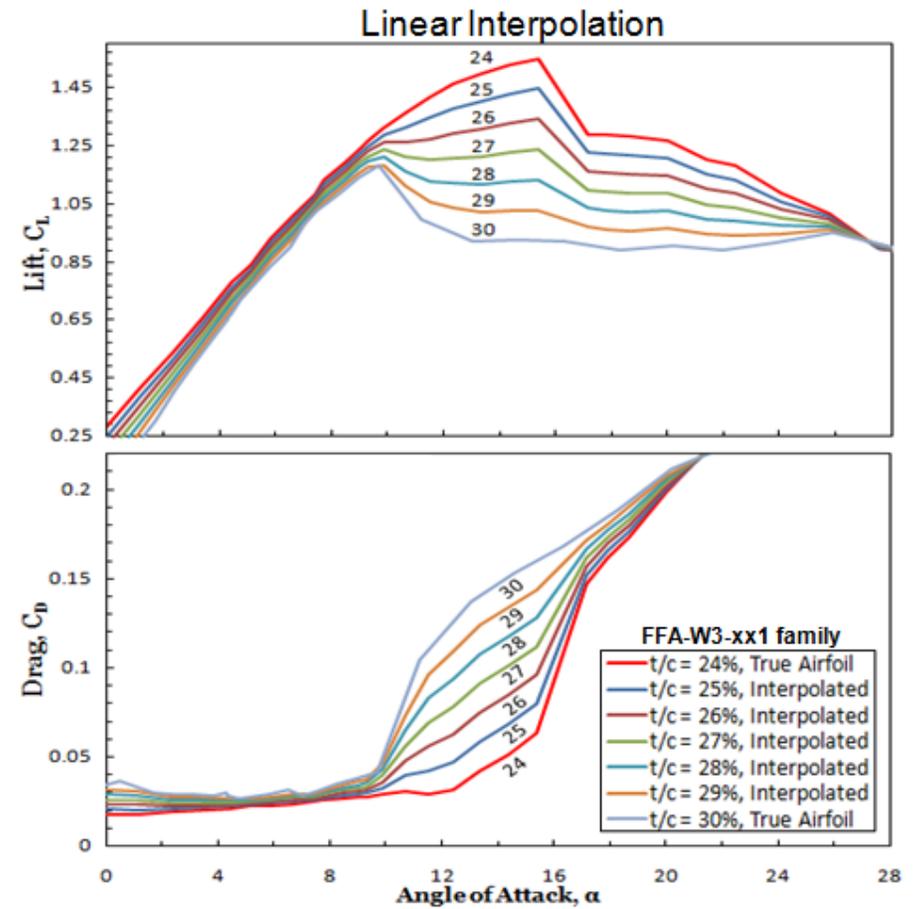


Figure 4: Interpolation of lift and drag coefficients with percent thickness as the variable using simple linear interpolation.

Blades with circular roots can be modeled by selecting the **Model blade with circular root** button. When modeling blades with circular roots, you will need to create additional airfoil data files which represent a circle (see the example files included). The lift and drag coefficients (and minimum pressure coefficient if modeling hydrokinetic turbines) must be included in a file named XXX.dat. If performing a structural optimization, you will also need to create a file containing the x-y coordinates for a circle; this file must be named XXX.prof. “XXX” is the string entered into the FoilFam field.

For circular root blades, the spanwise distributions for twist, chord, and percent thickness become more complicated to determine. The process will be explained in the order in which HARP_Opt calculates, and is illustrated in figure 7.

- 1) The chord distribution now uses 8 control points (CP), and an 8th order Bezier curve defines the chord profile. CP's 4-8 are at fixed radial locations, and the optimal chord values of the CP's 4-8 are determined by the genetic algorithm. The chord value of CP₃=CP₄, and the chord value of CP₂=CP₁. The radial distance between CP₁ and CP₂, and between CP₃ and CP₄ is a fixed value (hard-coded within HARP_Opt and scales with blade radius). CP₁ is the root chord, and its chord value must be determined iteratively in order to satisfy the requirement that thickness (dimensional and non-dimensional) remain monotonically decreasing. This iterative process to determine the root chord is illustrated in figure 8.
- 2) The radial value of CP₁ for the twist moves with the radial location of maximum chord. The twist is defined with a 5th order Bezier curve, except that twist now remains constant inboard of the maximum chord.
- 3) The percent thickness distribution in region R₁ is defined identically as described previously (can be **Piecewise Constant** or **Piecewise Linear**).
- 4) The dimensional thickness in region R₂ is calculated simply by combining chord and percent thickness: $t=c*(t/c)$.
- 5) The dimensional thickness at the left and right endpoints of region R₃ are known, so a cubic polynomial is fit between these endpoints. This polynomial has a fixed slope of 0 at the left endpoint and the slope at the right endpoint is forced to match the slope at the beginning of region R₂.
- 6) The percent thickness in region R₄ is now simply the dimensional thickness divided by the chord. The airfoil profiles and aerodynamic coefficients in region R₄ are interpolated from a circle at the root to the airfoil at the first percent thickness control point.

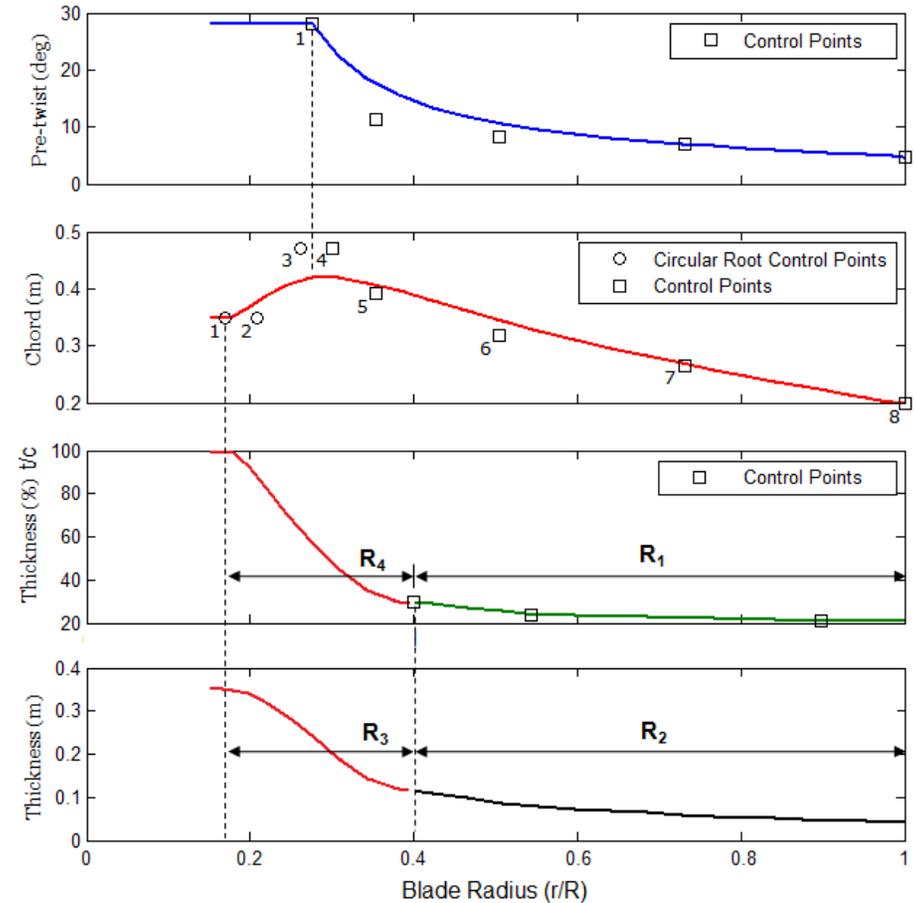


Figure 5: Blade geometry for blades with circular roots.

The root chord must be determined iteratively to ensure that the dimensional and non-dimensional thickness distributions are monotonically decreasing. Figure 8 illustrates an example where small root chord values can result in infeasible blade shapes. **RtChordMin** and **RtChordMax** are the minimum and maximum possible root chords used for this iteration. **RtTranSt** defines the radial location (r/R) of CP₁ for the chord profile (see figure 7); **RtTranSt** is the radial location (r/R) where the circle begins transitioning into an airfoil. **RtTranEnd** defines the radial location (r/R) of CP₄ for the chord profile; **RtTranEnd** is the maximum (r/R) location where the maximum chord could occur.

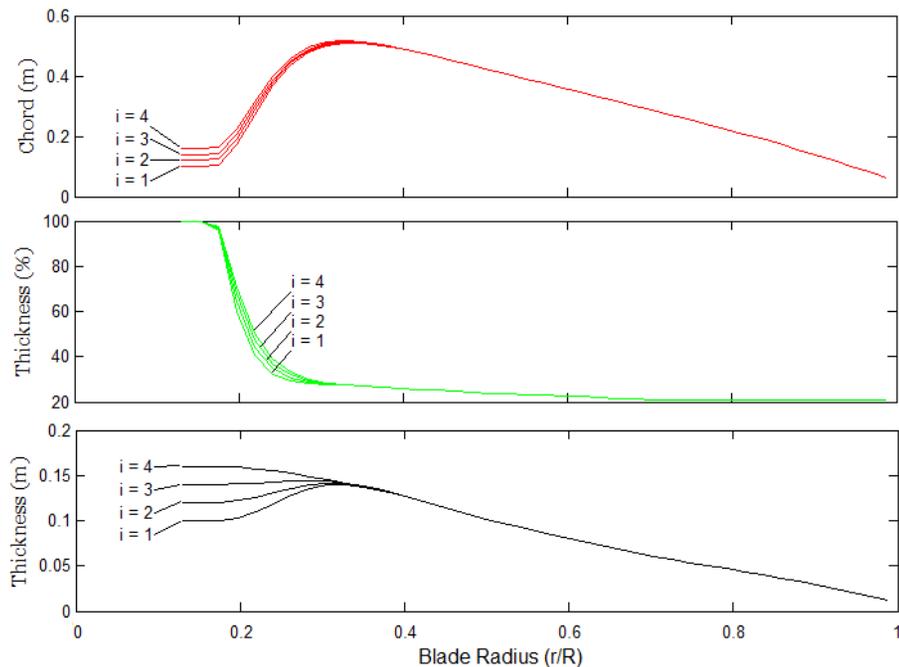


Figure 6: This examples illustrates the iterative process used to determine the root chord. Four iterations are shown: the root chord increases incrementally by 0.01 m (hard coded within HARP_Opt). Iterations $i=1, 2,$ and 3 result in infeasible blade shapes (root chord too small), but $i=4$ satisfies the requirements for monotonically decreasing dimensional and percent thickness.

Genetic Algorithm Configuration

The **Seed initial population** checkbox has been disabled in HARP_Opt v2.00.00, this functionality will return in future versions with an improved implementation.

PopSize determines the number of individuals that will be created for each generation. Increasing the population size enables the genetic algorithm to more thoroughly search the solution space and thereby obtain a better result. However, the larger the population size, the longer the genetic algorithm takes to compute each generation.

The number of elite individuals is determined by the value of **EliteCt**. The elite count is the number of individuals with the best fitness values in the current generation that are guaranteed to survive to the next generation. When **EliteCt** is at least 1, the solution can only improve, or at least stay the same form one iteration to the next. However, setting a high value for **EliteCt** can cause the fittest individuals to dominate the population, which can make the search less effective and may cause a pre-mature convergence. **EliteCt** is disabled during multi-objective optimization (when structural optimization is

enabled), this is because the genetic algorithm handles elite individuals differently during multi-objective optimization.

ParetoFrc limits the number of individuals that are on the Pareto front. For example, if **ParetoFrc** is equal to 0.35, the solver will try to limit the number of individuals in the current population that are on the Pareto front to 35% of the population size (**PopSize**).

CrossFrc specifies the crossover fraction. The crossover fraction specifies the fraction of the population, other than elite children, that is created by the crossover reproduction function. The remaining fraction of the population is created by the mutation reproduction function. Crossover and mutation are essential to the genetic algorithm optimization. Crossover enables the genetic algorithm to extract genes from the best performing individuals and recombine them into potentially superior children individuals. Mutation makes random changes to an individual, which adds diversity to the population and therefore searches the solution space more effectively. A crossover fraction of 1 means that all children will be crossover children, while a crossover fraction of 0 means that all children will be mutation children—neither of these extremes is an effective optimization strategy.

NumGen is a stopping criteria which determines the maximum number of generations that the genetic algorithm will run for. The genetic algorithm will terminate when **NumGen** has been reached, unless it converges before this number is reached. The genetic algorithm will also terminate if the number of stall-generations exceeds 50 (currently hard coded within HARP_Opt). A stall generation is a generation which has not made an improvement over the previous generation.

The genetic algorithm optimization will converge when the average change in the fitness value, from one generation to the next, is less than the value specified by **GATol**.

Energy Estimates

This section of the GUI attempts to estimate the final outcome resulting from the selected input parameters. **AEP** and **CF** are the estimated annual energy production and capacity factor resulting from the selected input parameters. **AEP** and **CF** are dependent on **Prated**, **RotorDia**, the selected **Flow Distribution**, and the range of flow speeds defined by **SpdSt**, **SpdEnd**, and **SpdDel**. **AEP** and **CF** are calculated assuming a power curve with constant efficiency of $C_p=0.45$ over all flow speeds, except the power remains constant if it exceeds **Prated**. **AEP** and **CF** are intended to be estimates probably only accurate to maybe 5-10% of the values that will be achieved after the HARP_Opt optimization completes. Nonetheless, these estimates can be extremely helpful in planning the appropriate rotor diameter and rated power.

The **Plot Contours** button will create a plot which overlays the estimated **AEP** and **CF** contours for the selected **Flow Distribution**. The axis limits for this plot are defined by the **RotorDia** and **Prated** min and max fields. This plot

is also extremely helpful in understanding the relationships between rotor diameter, rated power, and the flow regime. An example of this contour plot is shown in figure 9.

The **Filename** field allows the user to enter a filename for the associated output files. A new folder will be created in the **\Output_Files** folder which has the name **Filename**, and all of the associated output files will contain the prefix **Filename** as well.

Clicking the **Begin Optimization** button initiates the HARP_Opt algorithm, and saves a screenshot of the GUI into the output directory. The **Save Settings** button opens a file browser menu and allows you to save the current GUI settings to a file. This is not simply a "save as default" button, you can save multiple presets using this save feature. The **Load Settings** button also brings up a file browser which allows you to load any of the GUI presets that have been saved previously.

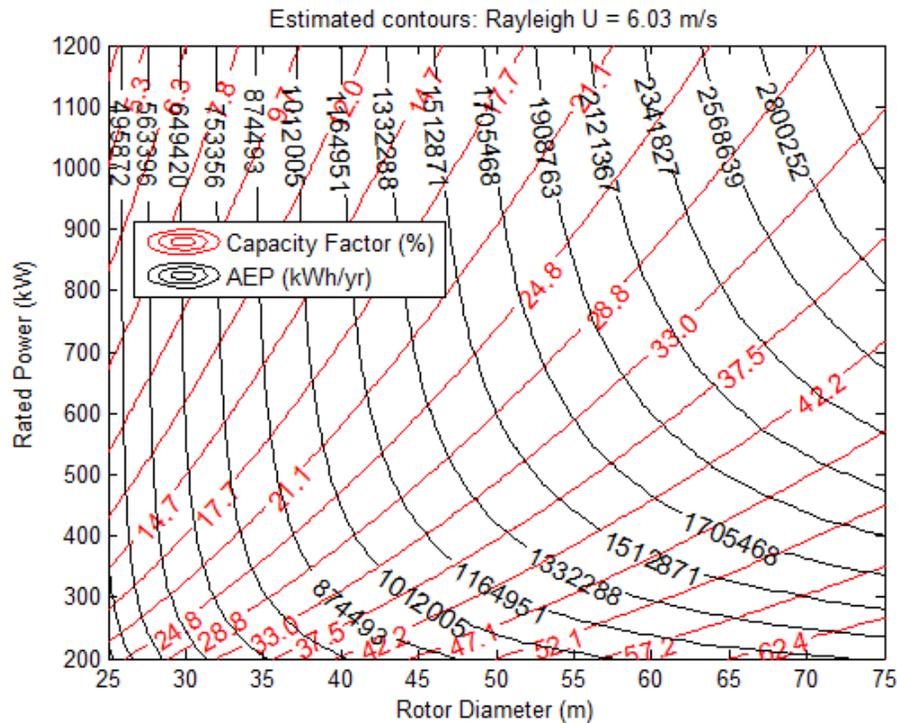


Figure 7: The Plot Contours button produces a contour plot which overlays estimates for AEP and CF. This plot can be helpful in choosing an appropriate rotor diameter and rated power.

Building and Naming the Airfoil Data Files

HARP_Opt accepts files formatted to the existing AeroDyn v12 style or the newer, slightly modified style for the airfoil/hydrofoil data input files.

It is not the intent of HARP_Opt to provide an extensive database of airfoils and hydrofoils, the airfoil/hydrofoil database is designed to be easily expanded by the user. The **Input_Files\Airfoil_Data** folder contains several example files which illustrate the proper format of the airfoil data input files. When you create new airfoil data input files for HARP_Opt, place the files into the **Input_Files\Airfoil_Data** folder. The first line of the airfoil data file must start with the string "AeroDyn Input File" in order for WT_Perf to recognize the correct format. The next two lines are for user comments.

For each block of data, the first line is the number of data tables in the file (each table corresponds to a unique Reynolds number). The next line contains the Reynolds number (in millions) of the current airfoil data table. The next line is the stall angle, the angle of attack at which maximum lift occurs; this value must be entered and is needed by the SPI interpolation algorithm. The next six lines are for AeroDyn's dynamic-stall model, but these values are ignored by WT_Perf; however you must still include these lines (but don't need to enter values).

A multicolumn table of the aerodynamic coefficients follows. The first column is for the angle-of-attack in degrees. The values of angle-of-attack must increase monotonically, and should span from -180 to 180 degrees. The second and third columns are for the lift and drag coefficients, respectively. If modeling a hydrokinetic turbine, add a fourth column for the minimum pressure coefficient data. At the end of each data table, add a line containing the string "EOT". This airfoil file format allows one to use different sets of angles-of-attack for each table.

Table 2: HARP_Opt airfoil file naming convention

FoilFam	% Thick	Filename
NACA	9	NACA_0090.dat
NACA	15	NACA_0150.dat
FFA	24	FFA_0240.dat
S_8xx	21.5	S_8xx_0215.dat
Circle	100	Circle_1000.dat

HARP_Opt requires the airfoil data filenames to be named using a specific naming convention. The prefix and suffix of the filename should be separated by a "_", where the prefix is the string which is entered into the **FoilFam** input field and the suffix is a 4-digit number which is the % thickness of the airfoil multiplied by 10. Table 2 illustrates by example the required file naming convention. For example, if you wanted to use the two airfoils from the NACA family in a particular design, which are 9% and 15% thick, you would enter "NACA" into the **FoilFam** input field and "9,15" into the **ThickVals** input field.

Only a single family name can be entered into the **FoilFam** input field; however, this does not mean that only a single family of airfoils can be used in the design. The file naming convention shown in the table above allows for flexibility such that multiple families of airfoils can be used along the span of the blade.

For example, let's say we want to use the NACA 4425 (25% thick) and FFA-W3-301 (30% thick) airfoils in the same design. We can enter a new family, called "FFACA" (or whatever we want) into the **FoilFam** field, enter the string "25,30" into the **ThickVals** field, and then rename the airfoil data files as shown in Table 3.

Table 3: Mixing airfoil families, ex. 1

Airfoil	% Thick	New Filename
NACA 4425	25	FFACA_0250.dat
FFA-W3-301	30	FFACA_0300.dat

There is no upper limit on the number of airfoils that can be used (i.e. no limit on the number of values that can be entered into the **ThickVals** field). Hopefully these examples have illustrated how to use any number and combination of airfoils in the HARP_Opt rotor design process.

Customizing the User Output Format

HARP_Opt allows users to customize the format of the output they will receive; this can be done by modifying The "Output_Template.xls" Excel file, located in the **Input_Files\Templates** folder. Any modifications made to the "Output_Template.xls" Excel file will be reflected in the "*_Output.xls" output files you will receive.

Open the "Output_Template.xls" file in Excel, and study how the "Source Data" for each figure is set to read from the appropriate columns. For example, if you wanted to add a new figure to the output, which plotted the Chord vs. Radius, simply create a new scatter plot in the Excel file and then set the "Source Data" to read from the columns which contain the Radius and Chord data. You can then also customize the colors, scale, legend, etc. of the plot, and these customized formats will appear in the "*_Output.xls" file every time you run HARP_Opt now.

A Note on Run-Time Performance

The run-time of HARP_Opt depends on the settings of many variables, and thus can vary greatly. Increasing **PopSize** has the most influential effect on run-time performance, and the run time scales linearly with **PopSize**. The structural optimization algorithm and use of stall-delay models significantly slow down the code. Increasing **NumSeg**, or setting **SpdDel** or **STdel** to a small number can all have a noticeable effect on run-time. If you wish to speed up

the run-time of HARP_Opt, try experimenting with these values to find a good balance between run-time and model fidelity.

For the best performance, it is strongly encouraged to temporarily disable anti-virus programs while using the HARP_Opt code. HARP_Opt creates many text files during each iteration, and some anti-virus programs try to scan every single file before it can be written to the hard drive. HARP_Opt must then wait for the anti-virus program to finish scanning each file before it can continue, massively slowing down the code (I have witnessed ~1000% slowdowns due to McAfee anti-virus).

Known Bugs

- After loading a previously saved GUI preset which has a user defined flow distribution selected, you may receive an error saying it cannot find the file—simply load the flow distribution file again using the Select File button to solve this error. This error occurs when moving HARP_Opt to a different directory from when the GUI preset was originally saved.

Caveats

NREL makes no promises about the usability or accuracy of HARP_Opt, which is essentially a beta code. NREL does not have the resources to provide full support for this code. *You may use HARP_Opt for evaluation purposes only.*

Possible Future Enhancements

HARP_Opt is very much a work in progress. Feature requests and suggestions are welcome. Here are a few ideas for future enhancements to HARP_Opt:

- Add a routine which would export the blade 3D coordinates to an Excel or text file. This would allow the blade shape to be easily imported into a CAD or FEM program for further analysis.
- Include the stall-delay model equations directly within the WT_Perf BEM algorithm—this will allow for the equations to be used more accurately and will also significantly increase run-time performance.
- Include a stall-delay model that would also correct the minimum pressure coefficients—this would improve the accuracy of the cavitation inception predictions.
- Improve the cavitation model to account for air and mineral content of the water, and also surface roughness of the blades. This would improve the accuracy of the cavitation inception predictions.
- Define a new creation function for the genetic algorithm which seeds the initial population with analytical Betz and Glauert optimum blades, in addition to the random blade shapes which are currently being

generated. Also, provide an option to seed with user defined blade shapes.

- Include a cost of energy (COE) model, and make COE an alternative objective. The COE model could perhaps be based on scaling laws at a subcomponent level.
- Improve the structural model to account for shear webs. Add constraints on tip deflection, buckling, and curvature. Incorporate more meaningful design loads, such as 50-year extreme which could be extrapolated using probabilistic methods.
- Incorporate discrete design variables into the optimization, where the design variables can only change by discrete values, as opposed to varying continuously. This would reduce the number of function evaluations needed, and would provide faster convergence while still maintaining high confidence in the final solution.
- Couple HARP_Opt to a more advanced flow model, such as vortex methods, or hybrid BEM/vortex methods. A more advanced flow model would have the ability to model non-planar blades. Additional design variables & constraints could be added to model & optimize features such as swept blades and winglets.
- Add ability to include “flatback” airfoils into HARP_Opt.

Another “feature” that I plan to work on is a significant rewrite of many parts of the HARP_Opt code. I apologize for any poor and inconsistent programming styles within—HARP_Opt is the first program I have written with significant complexity. After reading a few more guides on “best practices” in programming, I feel that it would be beneficial to improve the code’s organization and style to make the HARP_Opt code more understandable, maintainable, and expandable.

Acknowledgements

Jason Jonkman, Marshall Buhl, and Scott Hughes of the NWTC have been extremely helpful and offered many excellent suggestions which helped shape the HARP_Opt code into what it is today. HARP_Opt was written by Danny Sale, and the development of HARP_Opt was funded by the U.S. Department of Energy.

Feedback

Visit the NWTC forums to discuss the HARP_Opt code, available at: <http://wind.nrel.gov/forum/wind/>. We are interested to hear your feedback about the code. Please send your comments and questions to:

Danny C. Sale
Email: sale.danny@gmail.com
Voice: (865) 719-0557

Marshall L. Buhl, Jr.
NWTC/3811
National Renewable Energy Laboratory
1617 Cole Blvd.
Golden, CO 80401-3393
United States of America

Web: <http://wind.nrel.gov>
Email: marshall.buhl@nrel.gov
Voice: (303) 384-6914
Cell: (303) 915-6623
Fax: (303) 384-6901

References

1. Selig, M.S.; Du, Z. “A 3-D Stall-Delay Model for Horizontal Axis Wind Turbine Performance Prediction.” *AIAA-1998-21 Aerospace Sciences Meeting and Exhibit, 36th, and 1998 ASME Wind Energy Symposium, Reno, Nevada, January 12-15, 1998, Collection of Technical Papers*; pp. 12-15.