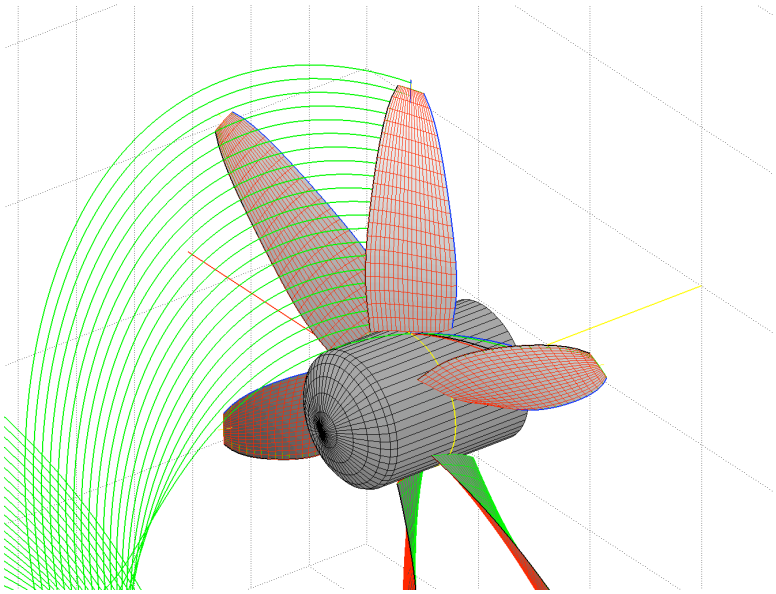
OpenProp: An Open-source Design Tool for Propellers and Turbines

B. P. Epps[1] (SM), **M. J. Stanway**[1] (SM), and **R. W. Kimball**[2] (AM)

1: Graduate student, Massachusetts Institute of Technology, Cambridge, MA
2: Professor, Maine Maritime Academy, Castine, ME



An open-sourced computational tool for the design and analysis of optimized propellers and turbines is presented. The design tool, called OpenProp*, is based on well-proven vortex lattice lifting line methods utilized by the US Navy as well as commercial designers. This paper presents the methodology and numerical implementation of* OpenProp*, with multiple examples of designs, including actual parts fabricated from the code using 3D printing technology.*

INTRODUCTION

OpenProp is a suite of open-sourced propeller and turbine design codes written in the MATLAB® programming language (Kimball 2007). The codes are based on the same lifting line propeller design theory utilized in codes employed by the US Navy for preliminary parametric design of marine propellers (Kerwin 2007). OpenProp is designed to be a GUI-based user-friendly tool that can be used by both propeller design professionals as well as novices in the propeller design field, though basic engineering knowledge is assumed.

OpenProp began in 2001 with the propeller code PVL developed by Kerwin as part of his MIT propeller design course notes (Kerwin 2007). The first Matlab version of this code, MPVL, incorporated Grahical User Interfaces for parametric design and preliminary bladerow design (Chung 2007). Geometry routines were later added which interfaced with the CAD program Rhino to generate a 3D printable propeller (D'epagnier 2007). These prior codes were capable of designing propellers using a simple Lerb's criteria optimizer routine. Using a generalized optimizer routine implemented by Epps (presented herein), the code was then extended to design ducted propellers (Stubblefield 2008).

OpenProp utilizes a vortex lattice lifting line representation of the blades with constant-diameter helical vortices to represent the blade wakes. The method incorporates a standard wake alignment procedure to accurately represent moderate blade loading and can design both propellers and axial flow turbines using the same numerical representation. The code also has an analysis capability to estimate the performance curve of a given design for use in off-design evaluation. This paper presents the methodology of the numerical implementation of both the propeller and turbine design capabilities. Multiple examples of designs are presented including validation comparisons and examples of actual parts fabricated from the code using 3D printing technology. Examples of both propeller and turbine designs are presented.

The long term goal of the project is to provide a user-friendly, accurate, and validated open-sourced code which can be used to design and prototype a variety of propellers and turbines including:

- Marine Propellers (free tip, ducted, and multicomponent)
- Marine Hydrokinetic Turbines (free tip, ducted, and multicomponent)
- Hydraulic turbines (propeller type and Kaplan)

A team of researchers at MIT, Maine Maritme Academy and University of Maine have contributed to the current OpenProp code. The code has been validated and demonstrated in both free-tip and ducted propeller design, and prototype free-tip propellers have been manufactured via 3D printing from the direct outputs of OpenProp.

The turbine design implementation is underway and in validation stages. Work has also been done to add analysis capability to enable off-design performance prediction for propeller or turbine designs. Coupling with the open-sourced code XFOIL is also underway, giving the code flexibility in designing and analyzing arbitrary foil shapes. For marine applications, cavitation prediction tools have also been developed utilizing XFOIL and the pressure distribution predictor. Future additions will include enhanced CAD and CAM modeling, strength analysis and coupling with electric motors. Visit the OpenProp website (http://openprop.mit.edu) for more information.

Data flow

OpenProp uses data structures to store the input parameters, design, geometry, and operating states of a particular propeller or turbine design. The data flow is illustrated in figure 1. The input data (such as propeller diameter, rotation rate, etc.) are defined by the user either though the GUI or by running a short script; these input data are organized in the **input.** data structure. The **input.** data is fed into the ***optimizer*** subroutine, which determines the optimum propeller/turbine design, for the input operating conditions: The output of the ***optimizer*** is a propeller/turbine **design.** data structure. The **design.** can then be analyzed at off-design conditions (i.e. at user-specified tip speed ratios) in the ***analyzer*** to determine off-design operating **states.**. The **design.** can also be sent to the ***crafter***, which determines the 3D **geometry.** and prepares 3D rapid prototyping files necessary for production of the propeller/turbine.

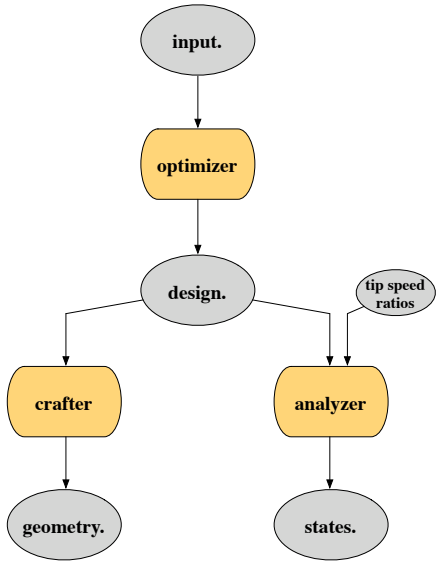


Figure 1: OpenProp information flow chart

METHODOLOGY

The following is the theoretical foundation and an overview of the numerical implementation of the OpenProp propeller/turbine design code . It draws from the theory presented in (Kerwin 2007, Coney 1989, Carlton 1994).

OpenProp is based on *moderately-loaded lifting line theory*, in which a propeller blade is represented by a lifting line, with trailing vorticity aligned to the local flow velocity (i.e. the vector sum of free stream plus induced velocity). The induced velocities are computed using a vortex lattice, with helical trailing vortex filaments shed into the wake at discrete stations along the blade. The blade itself is modeled as discrete sections, having 2D section properties at each radius. Loads on the blade are computed by integrating the load on each section of the blade over the length of the blade. The goal of the propeller or turbine optimization procedures is to determine the optimum circulation distribution along the span of the blade, which yields the best performance, given the inflow conditions and blade 2D section properties.

All formulae in this section are given in dimensional terms. All formulae are developed for the propeller case, in which $\Gamma > 0$. For the turbine case, $\Gamma < 0$ automatically produces all necessary sign changes to model the turbine.

Propeller velocity/force diagram

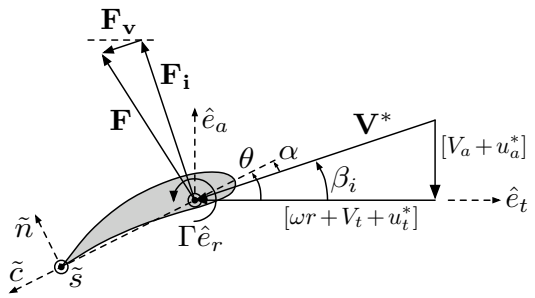


Figure 2: Propeller velocity/force diagram, as viewed from the tip towards the root of the blade. All velocities are relative to a stationary blade section at radius r.

The velocity/force diagram shown in figure 2 illustrates the velocities and forces (per unit span) on a 2D blade section in the axial $\hat{e}_a$ and tangential $\hat{e}_t$ directions. The propeller shaft rotates with angular velocity $\omega\hat{e}_a$, such that the apparent tangential inflow at a 2D section at radius r is $-\omega r\hat{e}_t$. Also shown on figure 2 are the axial and tangential (swirl) inflow velocities, $\mathbf{V_a} = -V_a\hat{e}_a$ and $\mathbf{V_t} = -V_t\hat{e}_t$; induced axial and tangential velocities, $\mathbf{u_a^*} = -u_a^*\hat{e}_a$ and $\mathbf{u_t^*} = -u_t^*\hat{e}_t$ (note that typically $u_t^* < 0$ when using this definition, so $\mathbf{u_t^*}$ actually points in the $\hat{e}_t$ direction); and the total resultant inflow velocity, $\mathbf{V}^*$, which has magnitude

$$V^* = \sqrt{(V_a+u_a^*)^2+(\omega r+V_t+u_t^*)^2} \tag{1}$$

and is oriented at pitch angle,

$$\beta_i = \tan^{-1}\left[\frac{V_a + u_a^*}{\omega r + V_t + u_t^*}\right] \tag{2}$$

to the $\hat{e}_t$ axis. Also shown on figure 2 are the angle of attack, α; blade pitch angle $\theta = \beta_i + \alpha$; circulation, $\Gamma\hat{e}_r$; (inviscid) Kutta-Joukowski lift force, $\mathbf{F_i} = \rho\mathbf{V}^* \times (\Gamma\hat{e}_r)$; viscous drag force, $\mathbf{F_v}$, aligned with $\mathbf{V}^*$; and total force per unit radius, $\mathbf{F} = \mathbf{F_i} + \mathbf{F_v}$. The total force per unit radius can be decomposed into axial and tangential components

$$\mathbf{F_a} = [F_i \cos\beta_i - F_v \sin\beta_i]\ (\hat{e}_a) \tag{3}$$
$$\mathbf{F_t} = [F_i \sin\beta_i + F_v \cos\beta_i]\ (-\hat{e}_t) \tag{4}$$

where the magnitudes of the inviscid and viscous force per unit radius are

$$F_i = \rho V^* \Gamma \tag{5}$$
$$F_v = \tfrac{1}{2}\rho (V^*)^2 C_D c \tag{6}$$

Assuming the Z blades are identical, the total thrust and torque on the propeller are

$$\begin{aligned}\mathbf{T} &= Z\int_{r_h}^{R} \mathbf{F_a} dr \\ &= \rho Z \int_{r_h}^{R} [V^*\Gamma\cos\beta_i - \tfrac{1}{2}(V^*)^2 C_D c \sin\beta_i] dr\ (\hat{e}_a)\end{aligned} \tag{7}$$

$$\begin{aligned}\mathbf{Q} &= Z\int_{r_h}^{R} r\hat{e}_r \times \mathbf{F_t} dr \\ &= \rho Z \int_{r_h}^{R} [V^*\Gamma\sin\beta_i + \tfrac{1}{2}(V^*)^2 C_D c \cos\beta_i] r dr\ (-\hat{e}_a)\end{aligned} \tag{8}$$

where r_h and R are the radius of the hub and blade tip, respectively. The fluid-dynamic power acting on the propeller is the product of torque and angular velocity.

$$\begin{aligned}P &= Q(-\hat{e}_a)\cdot\omega\hat{e}_a \\ &= -\rho Z\omega \int_{r_h}^{R} [V^*\Gamma\sin\beta_i + \tfrac{1}{2}(V^*)^2 C_D c\cos\beta_i] r dr\end{aligned} \tag{9}$$

where the leading $(-)$ sign indicates that power is being put into the fluid by the propeller (i.e. the torque resists the motion).

The useful power produced by the propeller is TV_∞, where V_∞ is the free-stream speed (i.e. ship speed), and the efficiency of the propeller is

$$\eta = \frac{TV_\infty}{Q\omega} \tag{10}$$

Turbine representation

In this section, we demonstrate that a turbine can be represented in the above formulation simply by using a negative circulation, $\Gamma < 0$. If $\Gamma < 0$, then $\{C_L, F_i, u_a^*, u_a^*, f_0, \alpha\} < 0$ as well, via equations $\{(11), (5), (14), (15), (33)\}$.

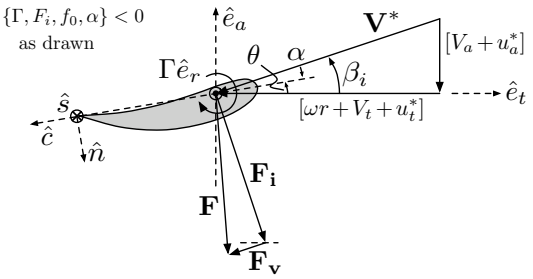


Figure 3: Turbine velocity/force diagram, as viewed from the tip towards the root of the blade. All velocities are relative to a stationary blade section at radius r.

The turbine velocity/force diagram is shown in figure 3, with $\{\Gamma, F_i, f_0, \alpha\} < 0$ as drawn. In this case, the turbine still rotates with angular velocity $\omega\hat{e}_a$, but the direction of the circulation is reversed (as drawn). This amounts to $|\Gamma|(-\hat{e}_r) = \Gamma\hat{e}_r$ with $\Gamma < 0$.

With, $\{\Gamma, F_i\} < 0$, equations (3) and (4) yield the axial and tangential forces in the turbine case:

$$\begin{aligned}\mathbf{F_a} &= [|F_i|\cos\beta_i + F_v\sin\beta_i]\ (-\hat{e}_a) && \text{(as drawn)} \\ &= [F_i\cos\beta_i - F_v\sin\beta_i]\ (\hat{e}_a) && \text{(eqn. 3)} \\ \mathbf{F_t} &= [|F_i|\sin\beta_i - F_v\cos\beta_i]\ (\hat{e}_t) && \text{(as drawn)} \\ &= [F_i\sin\beta_i + F_v\cos\beta_i]\ (-\hat{e}_t) && \text{(eqn. 4)}\end{aligned}$$

Similarly, the total thrust

$$\mathbf{T} = \rho Z \int_{r_h}^{R} [V^*|\Gamma|\cos\beta_i + \tfrac{1}{2}(V^*)^2 C_D c\sin\beta_i] dr\ (-\hat{e}_a) \quad \text{(as drawn)}$$

$$= \rho Z \int_{r_h}^{R} [V^*\Gamma\cos\beta_i - \tfrac{1}{2}(V^*)^2 C_D c\sin\beta_i] dr\ (\hat{e}_a) \quad \text{(eqn. 7)}$$

torque

$$\mathbf{Q} = \rho Z \int_{r_h}^{R} [V^*|\Gamma|\sin\beta_i - \tfrac{1}{2}(V^*)^2 C_D c\cos\beta_i] r dr \quad (\hat{e}_a) \quad \text{(as drawn)}$$

$$= \rho Z \int_{r_h}^{R} [V^*\Gamma\sin\beta_i + \tfrac{1}{2}(V^*)^2 C_D c\cos\beta_i] r dr \quad (-\hat{e}_a) \quad \text{(eqn. 8)}$$

and power

$$P = \rho Z\omega \int_{r_h}^{R} [V^*|\Gamma|\sin\beta_i - \tfrac{1}{2}(V^*)^2 C_D c\cos\beta_i] r dr \quad \text{(as drawn)}$$

$$= -\rho Z\omega \int_{r_h}^{R} [V^*\Gamma\sin\beta_i + \tfrac{1}{2}(V^*)^2 C_D c\cos\beta_i] r dr \quad \text{(eqn. 9)}$$

are predicted correctly by equations (7), (8), and (9) when $\Gamma < 0$ for the turbine. (Note that $P > 0$ indicates that power is being extracted from the fluid by the turbine.) Therefore, equations (3),

(4), (7), (8), and (9) can be used for either a propeller ($\Gamma > 0$) or turbine ($\Gamma < 0$) case. Furthermore, the same vortex lattice code can be used for both the propeller and turbine cases! All subsequent equations herein are given using the *propeller* sign conventions.

Section lift and drag coefficients

In the design optimizer, the lift coefficient of each 2D blade section is given in terms of its loading by

$$C_L = \frac{F_i}{\frac{1}{2}\rho(V^*)^2 c} = \frac{2\Gamma}{(V^*)c} \tag{11}$$

The section drag coefficient, C_D, is a constant specified by the user.

During the circulation optimization procedure, the chord, c, is chosen in order to restrict the lift coefficient to a given maximum allowable absolute value, $C_{L_{\max}}$, such that

$$C_L = C_{L_{\max}} \cdot \frac{\Gamma}{|\Gamma|} \tag{12}$$

$$c = \frac{2|\Gamma|}{(V^*)C_{L_{\max}}} \tag{13}$$

It is important to restrict the maximum lift coefficient in order to prevent flow separation and cavitation at the leading edge of the propeller/turbine blade. The absolute values in (12) and (13) are necessary for the turbine case, in which $\Gamma < 0$ and $C_L = -C_{L_{\max}}$, but $c > 0$.

Vortex lattice formulation

OpenProp employs a standard propeller vortex lattice model to compute the axial and tangential induced velocities, $\{u_a^*, u_t^*\}$. In the vortex lattice formulation, a Z-bladed propeller is modeled as a single representative radial lifting line, partitioned into M panels. A horseshoe vortex filament with circulation $\Gamma(i)$ surrounds the i^{th} panel, consisting of helical trailing vortex filaments at the panel endpoints ($r_v(i)$ and $r_v(i+1)$) and the segment of the lifting line that spans the panel. The induced velocities are computed at *control points* on the lifting line at radial locations $r_c(m)$, $m = 1 \ldots M$, by summing the velocity induced by each horseshoe vortex

$$u_a^*(r_c(m)) \equiv u_a^*(m) = \sum_{i=1}^{M} \Gamma(i)\bar{u}_a^*(m,i) \tag{14}$$

$$u_t^*(r_c(m)) \equiv u_t^*(m) = \sum_{i=1}^{M} \Gamma(i)\bar{u}_t^*(m,i) \tag{15}$$

where $\bar{u}_a^*(m,i)$ and $\bar{u}_t^*(m,i)$ are the axial and tangential velocity induced at $r_c(m)$ by a unit-strength horseshoe vortex surrounding panel m. Since the lifting line itself does not contribute to the induced velocity,

$$\bar{u}_a^*(m,i) = \bar{u}_a(m,i) - \bar{u}_a(m,i+1) \tag{16}$$

$$\bar{u}_t^*(m,i) = \bar{u}_t(m,i) - \bar{u}_t(m,i+1) \tag{17}$$

where $\bar{u}_a(m,i)$ and $\bar{u}_t(m,i)$ are the axial and tangential velocities induced at $r_c(m)$ by a unit-strength helical vortex filament at $r_v(i)$, with the vector direction of the circulation approaching the lifting line by right-hand rule. These velocities are computed using the approximations by Wrench (1957) for a constant-pitch helical vortex line:

For $r_c(m) < r_v(i)$:

$$\bar{u}_a(m,i) = \frac{Z}{4\pi r_c}(y - 2Zyy_0F_1)$$

$$\bar{u}_t(m,i) = \frac{Z^2}{2\pi r_c}(y_0F_1)$$

For $r_c(m) > r_v(i)$:

$$\bar{u}_a(m,i) = -\frac{Z^2}{2\pi r_c}(yy_0F_2)$$

$$\bar{u}_t(m,i) = \frac{Z}{4\pi r_c}(1 + 2Zy_0F_2)$$

where

$$F_1 \approx \frac{-1}{2Zy_0}\left(\frac{1+y_0^2}{1+y^2}\right)^{0.25}\left(\begin{array}{l}\frac{1}{U^{-1}-1}+\\ \frac{1}{24Z}\left[\frac{9y_0^2+2}{(1+y_0^2)^{1.5}}+\frac{3y^2-2}{(1+y^2)^{1.5}}\right]\\ \cdot\ln\left|1+\frac{1}{U^{-1}-1}\right|\end{array}\right)$$

$$F_2 \approx \frac{1}{2Zy_0}\left(\frac{1+y_0^2}{1+y^2}\right)^{0.25}\left(\begin{array}{l}\frac{1}{U-1}-\\ \frac{1}{24Z}\left[\frac{9y_0^2+2}{(1+y_0^2)^{1.5}}+\frac{3y^2-2}{(1+y^2)^{1.5}}\right]\\ \cdot\ln\left|1+\frac{1}{U-1}\right|\end{array}\right)$$

$$U = \left(\frac{y_0\left(\sqrt{1+y^2}-1\right)}{y\left(\sqrt{1+y_0^2}-1\right)}\exp\left(\sqrt{1+y^2}-\sqrt{1+y_0^2}\right)\right)^Z$$

$$y = \frac{r_c}{r_v \tan\beta_w}$$

$$y_0 = \frac{1}{\tan\beta_w}$$

and β_w is the pitch angle of the helical vortices in the wake. Consistent with moderately-loaded lifting line theory, we set $\beta_w = \beta_i$ in order to 'align' the wake with the local flow at the blade.

For an infinite-bladed propeller, these equations become

For $r_c(m) < r_v(i)$:

$$\bar{u}_a(m,i) = \frac{Z}{4\pi r_v \tan\beta_w}$$

$$\bar{u}_t(m,i) = 0$$

For $r_c(m) > r_v(i)$:

$$\bar{u}_a(m,i) = 0$$
$$\bar{u}_t(m,i) = \frac{Z}{4\pi r_c}$$

The infinite blade approximation is made when comparing the vortex lattice model in OpenProp to the performance of an actuator disc representation of a propeller or turbine.

Hub effects

Following Kerwin (2007), the hub is modeled as an image vortex lattice, with the image trailing vortex filaments having equal and opposite strength as the real trailing vortex filaments and radii

$$r_i(i) = \frac{r_h^2}{r_v(i)} \tag{18}$$

The axial velocity influence function, $\bar{u}_a(m,i)$, then becomes difference between the axial velocity induced by a trailing vortex at $r_v(i)$ and that induced by a trailing vortex at $r_i(i)$. The same modification is made for $\bar{u}_t(m,i)$.

The image vorticity is shed through the trailing surface of the hub and rolls up into a *hub vortex* of radius, r_o. The drag due to the hub vortex is

$$D_h = \frac{\rho Z^2}{16\pi}\left[\ln\left(\frac{r_h}{r_o}\right)+3\right][\Gamma(1)]^2 \quad (-\hat{e}_a) \tag{19}$$

For practical purposes, it suffices to set $\frac{r_h}{r_o} = 1$, which sets the logarithm to zero.

Propeller optimization subroutine

Following Coney (1989), the propeller optimization problem is to find the set of M circulations of the vortex lattice panels that produce the least torque for a required thrust. The torque (8) is

$$Q = \rho Z \sum_{m=1}^{M} \left\{[V_a + u_a^*]\Gamma + \tfrac{1}{2}V^* C_D c[\omega r_c + V_t + u_t^*]\right\} r_c \triangle r_v \tag{20}$$

where $\{\rho, Z, \omega\}$ are constants and $\{\Gamma,\ u_a^*,\ u_t^*,\ V^*,\ c,\ V_a,\ V_t,\ C_D,\ r_c,\ \triangle r_v\}$ are evaluated at each control point radius, $r_c(m)$, in the summation. Note that $\{\Gamma, u_a^*, u_t^*, V^*, c\}$ are functions of the circulation distribution, Γ. The thrust (7) is required to be the specified thrust, T_s

$$T = \rho Z \sum_{m=1}^{M} \left\{[\omega r_c + V_t + u_t^*]\Gamma - \tfrac{1}{2}V^* C_D c[V_a + u_a^*]\right\} \triangle r_v$$
$$- \text{Hflag} \cdot \frac{\rho Z^2}{16\pi}\left[\ln\left(\frac{r_h}{r_o}\right)+3\right][\Gamma(1)]^2 = T_s \tag{21}$$

where Hflag is set to 1 to model a hub or 0 for no hub.

The circulation optimization is performed using the method of the Lagrange multiplier from variational calculus. An auxiliary function,

$$H = Q + \lambda_1(T - T_s) \tag{22}$$

is formed, where λ_1 is the unknown Lagrange multiplier which introduces the thrust constraint (21). Clearly, if $T = T_s$, then a minimum value of H coincides with a minimum value of Q. To find this minima, the partial derivatives with respect to the unknowns are set to zero

$$\frac{\partial H}{\partial \Gamma(i)} = 0 \ \text{ for } \ i = 1 \ldots M \tag{23}$$
$$\frac{\partial H}{\partial \lambda_1} = 0 \tag{24}$$

which results in a system of $M+1$ equations for as many unknowns $\{\Gamma(i = 1 \ldots M), \lambda_1\}$.

The partial derivatives of $\Gamma, \lambda_1, u_a^*, u_t^*, V^*,$ and c with respect to $\{\Gamma(i), \lambda_1\}$ are

$$\frac{\partial \Gamma(m)}{\partial \Gamma(i)} = \begin{cases} 0 & (m \neq i) \\ 1 & (m = i) \end{cases}, \quad \frac{\partial \lambda_1}{\partial \lambda_1} = 1$$

$$\frac{\partial \{u_a^*(m), u_t^*(m)\}}{\partial \Gamma(i)} = \{\bar{u}_a^*(m,i), \bar{u}_t^*(m,i)\}$$

$$\frac{\partial V^*(m)}{\partial \Gamma(i)} = \tfrac{1}{2}(V^*)^{-1}\begin{pmatrix} 2(V_a + u_a^*)\frac{\partial u_a^*(m)}{\partial \Gamma(i)} + \\ 2(\omega r_c + V_t + u_t^*)\frac{\partial u_t^*(m)}{\partial \Gamma(i)} \end{pmatrix}$$
$$= \sin(\beta_i(m))\,\bar{u}_a^*(m,i) + \cos(\beta_i(m))\,\bar{u}_t^*(m,i)$$

$$\frac{\partial c(m)}{\partial \Gamma(i)} = \frac{2}{V^*(m) C_{L_{\max}}}\frac{\partial \Gamma(m)}{\partial \Gamma(i)} \cdot \frac{\Gamma(m)}{|\Gamma(m)|} - \frac{c(m)}{V^*(m)}\frac{\partial V^*(m)}{\partial \Gamma(i)}$$

All other partial derivatives are zero or ignored.

The system of equations {(23), (24)} is non-linear, so an iterative approach must be used to solve them. Note that the system state is characterized by Γ and flow parameters $\{u_a^*,\ u_t^*,\ \beta_i,\ \bar{u}_a^*,\ \bar{u}_t^*,\ V^*,\ c\}$, which all must be self-consistent for the state to be physically-realistic. That is, equations {(14), (15), (2), (16), (17), (1), (13)} must all hold, given Γ. During each solution iteration, flow parameters $\left\{u_a^*, u_t^*, \bar{u}_a^*, \bar{u}_t^*, V^*, \frac{\partial V^*}{\partial \Gamma}, c, \frac{\partial c}{\partial \Gamma}, \lambda_1\right\}$ are frozen in order to linearize {(23), (24)}. The linear system of equations, with the linearized unknowns marked as $\{\breve{\Gamma}, \breve{\lambda}_1\}$, is

$$
\begin{aligned}
\frac{\partial H}{\partial \Gamma(i)} = \quad & \rho Z \sum_{m=1}^{M} \breve{\Gamma}(m) \cdot \begin{pmatrix} \bar{u}_a^*(m,i) r_c(m) \triangle r_v(m) + \\ \bar{u}_a^*(i,m) r_c(i) \triangle r_v(i) \end{pmatrix} \\
& + \rho Z V_a(i) r_c(i) \triangle r_v(i) \\
& + \rho Z \sum_{m=1}^{M} \tfrac{1}{2} C_D \left[\tfrac{\partial V^*(m)}{\partial \Gamma(i)} c(m) + V^*(m) \tfrac{\partial c(m)}{\partial \Gamma(i)} \right] \\
& \qquad \cdot [\omega r_c(m) + V_t(m) + u_t^*(m)] r_c(m) \triangle r_v(m) \\
& + \rho Z \sum_{m=1}^{M} \tfrac{1}{2} C_D V^*(m) c(m) [\bar{u}_t^*(m,i)] r_c(m) \triangle r_v(m) \\
& + \rho Z \lambda_1 \sum_{m=1}^{M} \breve{\Gamma}(m) \cdot \begin{pmatrix} \bar{u}_t^*(m,i) \triangle r_v(m) + \\ \bar{u}_t^*(i,m) \triangle r_v(i) \end{pmatrix} \\
& + \rho Z \breve{\lambda}_1 [\omega r_c(i) + V_t(i)] \triangle r_v(i) \\
& - \rho Z \breve{\lambda}_1 \sum_{m=1}^{M} \tfrac{1}{2} C_D \left[\tfrac{\partial V^*(m)}{\partial \Gamma(i)} c(m) + V^*(m) \tfrac{\partial c(m)}{\partial \Gamma(i)} \right] \\
& \qquad \cdot [V_a + u_a^*(m)] \triangle r_v \\
& - \rho Z \breve{\lambda}_1 \sum_{m=1}^{M} \tfrac{1}{2} C_D V^*(m) c(m) [\bar{u}_a^*(m,i)] \triangle r_v \\
& - \text{Hflag} \cdot \tfrac{\partial \Gamma(1)}{\partial \Gamma(i)} \cdot \lambda_1 \frac{\rho Z^2}{8\pi} \left[\ln\left(\frac{r_h}{r_o} \right) + 3 \right] \breve{\Gamma}(1) \\
= 0 \quad & \text{for} \quad i = 1 \ldots M
\end{aligned} \tag{25}
$$

$$
\begin{aligned}
\frac{\partial H}{\partial \lambda_1} = \quad & \rho Z \sum_{m=1}^{M} \breve{\Gamma}(m) \cdot [\omega r_c + V_t + u_t^*(m)] \triangle r_v \\
& - \rho Z \sum_{m=1}^{M} \tfrac{1}{2} C_D V^*(m) c(m) [V_a + u_a^*(m)] \triangle r_v \\
& - \text{Hflag} \cdot \frac{\rho Z^2}{16\pi} \left[\ln\left(\frac{r_h}{r_o} \right) + 3 \right] \Gamma(1) \cdot \breve{\Gamma}(1) \\
& - T_s \\
= 0 &
\end{aligned} \tag{26}
$$

The system $\{(25), (26)\}$ is solved for the now linear $\{\breve{\Gamma}, \breve{\lambda}_1\}$, and the new $\breve{\Gamma}$ is used to update the flow parameters. First, induced velocities $\{u_a^*, u_t^*\}$ are updated via $\{(14), (15)\}$ and 'repaired' by smoothing the velocities at the blade root and tip. This minor smoothing is critical to enable the entire system of equations to converge, because the alignment of the wake and the vortex influence functions which are fed into the next solution iteration are very sensitive to irregularities in the induced velocities. This smoothing is reasonable in the vortex-lattice model, since it introduces no more error than ignoring hub or tip vortex roll-up, or other flow features. Next, the wake angle, β_i, is updated via (2), and the non-linear terms are updated: $\left\{ \bar{u}_a^*, \bar{u}_t^*, V^*, \frac{\partial V^*}{\partial \Gamma}, c, \frac{\partial c}{\partial \Gamma}, \lambda_1 \right\}$. This process is repeated until convergence of the entire system, yielding an optimized circulation distribution and a physically-realistic design operating state. Initial values of $\left\{ \beta_i, V^*, \frac{\partial V^*}{\partial \Gamma}, \frac{\partial c}{\partial \Gamma} \right\}$ are computed with $\{u_a^*, u_t^*\} = 0$. The Lagrange multiplier is initialized at $\lambda_1 = -1$, and the section chord lengths at $c = 0$.

The generalized circulation optimizer described herein was implemented in OpenProp by Epps. Stubblefield (2008) validated the optimizer for unducted and ducted cases against the U.S. Navy code PLL with good agreement in circulation distribution over a wide range of duct loadings.

Turbine optimization subroutine

OpenProp optimizes turbine designs based on a vortex-lattice adaptation of actuator disc theory with swirl and viscous losses. During the design optimization, flow parameters $\{\Gamma, u_a^*, u_t^*, \bar{u}_a^*, \bar{u}_t^*, \beta_i\}$ must be self consistent to define a physically-realistic operating state of the turbine. That is, equations $\{(14), (15), (16), (17), (2)\}$ must hold, given Γ.

In the present optimization scheme, the tangential induced velocity is set to the actuator disc with swirl (*ADS*) value

$$u_t^* \equiv u_{t,ADS}^*$$

The remaining flow parameters $\{\Gamma, u_a^*, \bar{u}_a^*, \bar{u}_t^*, \beta_i\}$ are determined iteratively. Initially setting $u_a^* = u_{a,ADS}^*$ allows one to start a loop that computes β_i via (2), then $\{\bar{u}_a^*, \bar{u}_t^*\}$ via $\{(16), (17)\}$. Then, the circulation distribution is determined by solving the matrix equation

$$[\bar{u}_t^*] \cdot [\Gamma] = [u_{t,ADS}^*]$$

for Γ. Finally, u_a^* is computed via (14), and the loop restarts. Iteration continues until every state variable has converged.

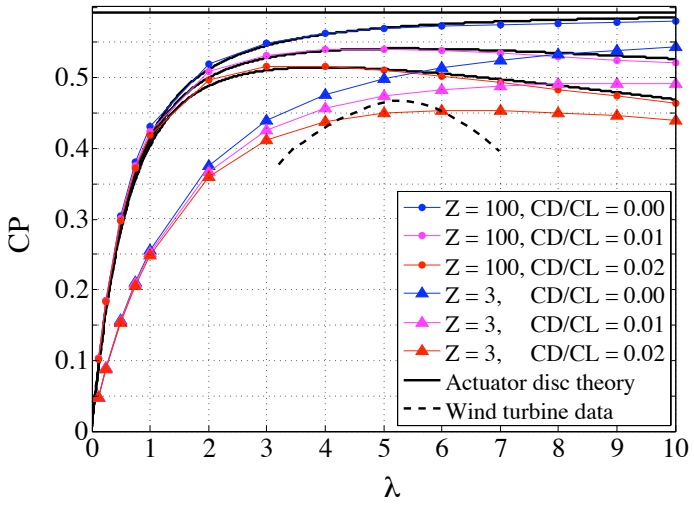


Figure 4: Power coefficient, $C_P = P / \frac{1}{2} \rho V_\infty^3 \pi R^2$, versus tip speed ratio, $\lambda = \frac{\omega R}{V_\infty}$, for optimized turbines. The C_P of turbines designed with 100 blades agrees quite well with actuator-disc-with-swirl-and-viscous-losses theory (Stewart 1976), as shown for three C_D/C_L ratios. Performance data of 3-bladed wind turbines in service, digitized from (Kahn 2006), is also shown.

Using this scheme, OpenProp is able to reproduce the C_P vs. λ performance curves from actuator-disc-with-swirl-and-viscous-losses theory (Stewart 1976), as shown by the (essentially infinite-bladed) $Z = 100$ curves in figure 4. An additional check that this scheme works correctly, which is not depicted in figure 4, is that for very high tip speed ratios ($\lambda > 40$), each of the $Z = 3$ curves asymptotes to its corresponding $Z = 100$ curve, as expected.

Clearly, the scheme presented here could be augmented to set $u_a^* \equiv u_{a,ADS}^*$ and solve for whatever u_t^*, etc. is self-consistent with that. The authors find marginally-worse agreement with actuator disc theory if this approach is used. One point of ongoing work is to reformulate the turbine optimization problem in such a way that does not use actuator disc theory as an input.

Incorrect turbine optimization scheme: One might formulate the turbine optimization problem statement as follows: Find the set of M circulations of the vortex lattice panels that produce the least torque (i.e. the most negative torque, giving the largest power extraction at the specified rotation rate). In other words, solve the propeller optimization problem with no thrust constraint. However, this scheme does **not** yield the largest power extraction possible (i.e. this scheme does **not** reproduce actuator disc theory), as shown in figure 5. The reason for this discrepancy is as follows.

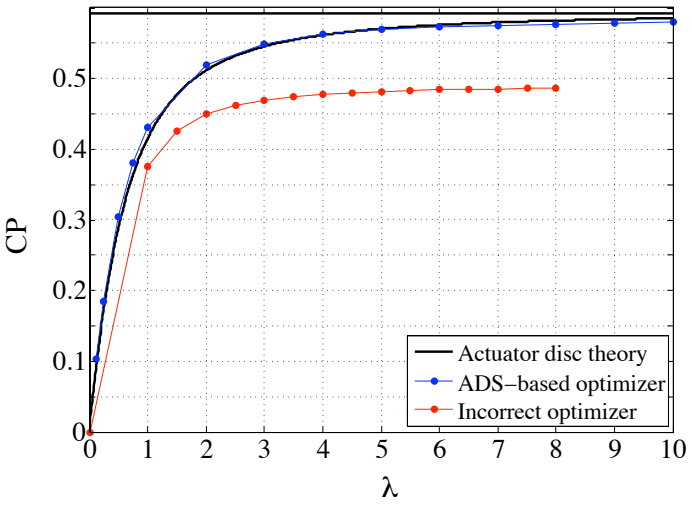


Figure 5: Power coefficient, C_P, versus tip speed ratio, λ, for turbines "optimized" by solving the system of equations: $\frac{\partial Q}{\partial \Gamma(i)} = 0$ for $i = 1,\ldots,M$. Here, $C_D = 0$ and $Z = 80$. Clearly, this scheme does not reproduce actuator-disc-with-swirl theory.

Consider the propeller optimization problem, with inviscid flow, $C_D = 0$, and no thrust constraint. Then, the system of equations for minimizing torque (25) becomes:

$$0 = \rho Z \sum_{m=1}^{M} \breve{\Gamma}(m) \cdot \left(\begin{array}{l} \bar{u}_a^*(m,i) r_c(m) \triangle r_v(m) + \\ \bar{u}_a^*(i,m) r_c(i) \triangle r_v(i) \end{array} \right) + \rho Z V_a(i) r_c(i) \triangle r_v(i) \qquad (27)$$

for $i = 1 \ldots M$. Since the horseshoe influence matrices $\{\bar{u}_a^*, \bar{u}_t^*\}$ are dominated by their diagonal terms, we can approximate

$$\bar{u}_a^*(m,i) \approx \begin{cases} 0 & (m \neq i) \\ \bar{u}_a^*(m,m) & (m = i) \end{cases} \qquad (28)$$

such that

$$u_a^*(m) \approx \Gamma(m) \bar{u}_t^*(m,m) \qquad (29)$$

The system of equations (27) then becomes M independent equations ($i = 1 \ldots M$)

$$0 = \rho Z \cdot \breve{\Gamma}(i) \cdot [\bar{u}_a^*(i,i) r_c(i) \triangle r_v(i) + \bar{u}_a^*(i,i) r_c(i) \triangle r_v(i)] + \rho Z V_a(i) r_c(i) \triangle r_v(i) \qquad (30)$$

which are each satisfied when

$$u_a^*(i) = -\tfrac{1}{2} V_a(i) \qquad (31)$$

Actuator disc theory prescribes $u_a^* = -\frac{1}{3} V_a$ for maximum power extraction. The turbine optimizer formulation presented in this section does not yield turbine designs with circulation distributions that extract as much power from the flow as actuator disc theory with swirl predicts, because solving (27) yields a circulation distribution which produces too much axial induced velocity, thereby reducing the flow rate through the turbine more than it should, resulting in less power available for extraction.

Geometry subroutine

Once the design operating state of the propeller/turbine is known, the geometry can be determined to give such performance. The 3D geometry is built from given 2D section profiles that are scaled and rotated according to $\{C_L = C_{L,\max}, c, \beta_i\}$, which were determined as part of the design operating state of the propeller/turbine.

The given 2D section geometry includes foil camber and thickness normalized by the chord, $\{\tilde{f}/c, \tilde{t}/c\}$, tabulated as a function of the normalized chordwise coordinate, x/c, as well as the ideal angle of attack, $\widetilde{\alpha}_I$, and ideal lift coefficient, $\widetilde{C}_{L_I}$. The latter are defined as

$$\widetilde{\alpha}_I \equiv \frac{1}{\pi} \int_0^\pi \frac{d\tilde{f}}{dx} dx'$$

$$\widetilde{C}_{L_I} \equiv 2 \int_0^\pi \frac{d\tilde{f}}{dx} \cos(x') dx'$$

where $x = \frac{c}{2}(1 - \cos(x'))$ defines the angular x' coordinate (Abbott 1959). Clearly, $\{\tilde{f}(x), \widetilde{\alpha}_I, \widetilde{C}_{L_I}\}$ scale linearly with maximum camber, $\tilde{f}_0 = \max[\tilde{f}(x)]$.

The 2D section lift coefficient is given in terms of the geometry by

$$C_L = 2\pi(\alpha - \alpha_I) + C_{L_I} \qquad (32)$$

In the OpenProp geometry subroutine, the angle of attack of each blade section is set to its ideal angle of attack ($\alpha = \alpha_I$) in order to prevent flow separation and/or cavitation at the leading

edge of the blade. The lift coefficient of each blade section then becomes the ideal lift coefficient $C_L = C_{L_I}$, by definition. In order to achieve the desired lift coefficient for a given blade section,$C_L = C_{L,\max}$, the ideal lift coefficient is scaled by scaling maximum section camber. Thus, for a given section profile with $\{\tilde{f}_0, \tilde{f}, \tilde{\alpha}_I, \widetilde{C_{L_I}}\}$, the desired maximum camber, camber profile, and ideal angle of attack are

$$\{f_0, f, \alpha_I\} = \frac{C_{L,\max}}{\widetilde{C_{L_I}}} \cdot \{\tilde{f}_0, \tilde{f}, \tilde{\alpha}_I\} \tag{33}$$

The pitch angle of each blade section is then set to

$$\theta = \alpha_I + \beta_i \tag{34}$$

Given the blade 2D section geometry, the OpenProp crafter can then form 3D renderings or export files for rapid prototyping of physical parts.

Analyzer subroutine

This section details the analysis of a propeller/turbine operating at an off-design (*OD*) tip speed ratio,

$$\lambda_{OD} = \frac{\omega_{OD} R}{V_\infty} \tag{35}$$

The 2D Section Lift and Drag Coefficients used in the analyzer subroutine are shown in figure 6. The lift coefficient is given by $C_L = 2\pi(\alpha - \alpha_I) \pm C_{L,\max}$ (eqn. 32) for $|\alpha - \alpha_I| < |\alpha - \alpha_I|_{\text{stall}}$ and is nearly constant for larger angles of attack. The drag coefficient at small angles of attack is $C_D = (C_D/C_L) \cdot (2\pi|\alpha - \alpha_I| + C_{L,\max})$, where C_D/C_L is a constant specified by the user; post stall, the drag coefficient increases linearly until it reaches a value of 2 at $|\alpha - \alpha_I| = 90$ deg. This type of stall model has been used successfully in (Drela 1999).

The net angle of attack is easily computed for a fixed geometry (i.e. θ fixed) by inspection of the propeller velocity diagram (noting that $\alpha = \alpha_I$ and $\beta_i = \beta_{i,\text{design}}$ at the design state)

$$\alpha - \alpha_I = \beta_{i,\text{design}} - \beta_i \tag{36}$$

The Operating States of a propeller or turbine for each given λ_{OD} are computed as follows. An operating state is defined by λ_{OD} and unknown flow parameters $\{V^*, \alpha, C_L, \Gamma, u_a^*, u_t^*, \beta_i, \bar{u}_a^*, \bar{u}_t^*\}$, which all must be self-consistent for the state to be physically-realistic. That is, equations {(1), (36), (32), (11), (14), (15), (2), (16), (17)} must all hold, given λ_{OD}. Since there are M vortex panels, there are $7M + 2M^2$ unknowns and a system of $7M + 2M^2$ non-linear equations that govern the state of the system. This system is solved in OpenProp using an approach similar to a Newton solver.

Since the $7M + 2M^2$ equations are coupled through the parameters $\{\beta_i, \bar{u}_a^*, \bar{u}_t^*\}$, we decouple them by considering two state vectors: $\mathbf{X} = \{V^*, \alpha, C_L, \Gamma, u_a^*, u_t^*\}^\top$ and $\mathbf{Y} = \{\beta_i, \bar{u}_a^*, \bar{u}_t^*\}^\top$. During each solution iteration, state vector $\mathbf{X}$ is updated, and then $\mathbf{Y}$ is updated; this process repeats until convergence of the entire system.

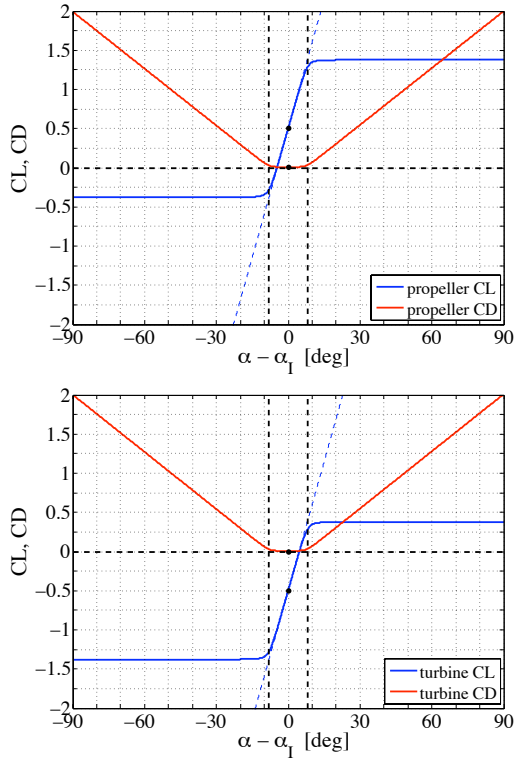


Figure 6: Lift coefficient, C_L, and drag coefficient, C_D, versus net angle of attack, $\alpha - \alpha_I$, for a propeller (top) and turbine (bottom) with specifications $C_{L,\max} = 0.5$ and $C_D/C_L = 0.01$. The vertical dashed lines at $|\alpha - \alpha_I|_{\text{stall}} = \pm 8$ deg indicate the stall angle of attack.

Consider state vector $\mathbf{X}$: It consists of M sets of 6 state variables, one set per vortex panel. The 6 variables for each vortex panel are coupled to one another, but not to the other variables in $\mathbf{X}$. Thus, $\mathbf{X}$ can be partitioned into M state vectors, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$, where $\mathbf{x}_m = \{V^*, \alpha, C_L, \Gamma, u_a^*, u_t^*\}^\top$ with each variable evaluated at $r_c(m)$. Each of these state vectors can be updated independently.

Each vortex panel state vector, $\mathbf{x}_m$, is updated using a Newton solver. Define the residual vector for the m^{th} panel as

$$\mathbf{R}_m = \begin{bmatrix} V^* - \sqrt{(V_a + u_a^*)^2 + (\omega_{OD} r_c + V_t + u_t^*)^2} \\ \alpha - (\alpha_I + \beta_{i,\text{design}} - \beta_i) \\ C_L - (2\pi(\alpha - \alpha_I) + C_{L_I}) \\ \Gamma - \left(\frac{1}{2} C_L V^* c\right) \\ u_a^* - [\bar{u}_a^*] \cdot [\Gamma] \\ u_t^* - [\bar{u}_t^*] \cdot [\Gamma] \end{bmatrix} \tag{37}$$

where each variable is evaluated at $r_c(m)$. In order to drive the residuals to zero, the desired change in the state vector, $\mathbf{dx}_m$, is found by solving the matrix equation

$$0 = \mathbf{R}_m + \mathbf{J}_m \cdot \mathbf{dx}_m$$

where the elements of the Jacobian matrix, $\mathbf{J}_m$, are

$$\mathbf{J}_m(i,j) = \frac{\partial \mathbf{R}_m(i)}{\partial \mathbf{x}_m(j)}$$

The state vector for the next iteration, then, is $\mathbf{x}_m^{\text{next}} = \mathbf{x}_m^{\text{current}} + \mathbf{dx}_m$. By solving one Newton iteration for each of the $m = 1,\ldots,M$ vortex panels, state vector $\mathbf{X} = \{\mathbf{x}_1,\ldots,\mathbf{x}_M\}$ is updated.

Given the new $\mathbf{X}$ values, $\mathbf{Y}$ is updated: β_i is updated via (2), and then $\{\bar{u}_a^*, \bar{u}_t^*\}$ are updated via $\{(16), (17)\}$. In the next solution iteration, these new values of $\mathbf{Y}$ are used to update $\mathbf{X}$, and so on. Since the solution scheme updates both $\mathbf{X}$ and $\mathbf{Y}$ in each iteration, it accounts for the coupled interaction between all $7M + 2M^2$ unknown flow parameters and converges on a physically-realistic operating state of the system.

The system is said to converge when all $6M$ elements of $\mathbf{X}$ have converged. Since β_i is directly related to α and $\bar{u}_a^*$ and $\bar{u}_t^*$ are functions of β_i, once α converges, this implies that $\mathbf{Y}$ has converged as well. For each operating state, the analyzer computes the propeller/turbine thrust, torque, and power coefficients and efficiency.

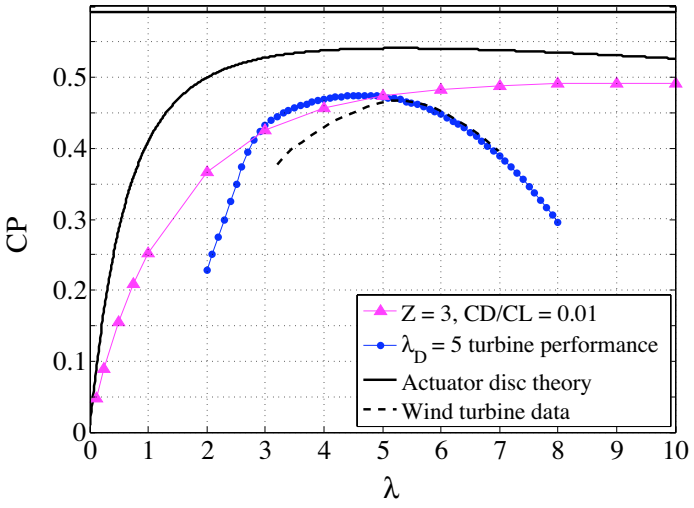


Figure 7: Power coefficient, C_P, versus off-design tip speed ratio, λ, for a turbine designed to operate at $\lambda_D = 5$, with specifications $C_D = 0.01$ and $Z = 3$.

Figure 7 shows the power coefficient versus off-design tip speed ratio performance curve of a turbine with specifications $C_D/C_L = 0.01$ and $Z = 3$, designed to operate at $\lambda_D = 5$. The performance predicted by the analyzer, '•', agrees with the performance predicted by the optimizer, '▲', at $\lambda = 5$, and the performance for higher tip speed ratios compares quite favorably with wind turbine industry performance data (Kahn 2006). For $\lambda < 3$, the power coefficient drops precipitously, as the net angle of attack drops below -8 degrees at many blade sections and the blade stalls.

Figure 7 also shows the design performance of other turbines with $C_D/C_L = 0.01$ and $Z = 3$, optimized for selected tip speed ratios, '▲'. Clearly, if all these turbines are truly optimized, our turbine optimized for $\lambda_D = 5$ should never outperform a turbine operating at its design point. The fact that our turbine outperforms the 'optimized' turbines at $\lambda = 4$ and 3 indicates that the turbine optimizer subroutine is most likely not yielding the best turbines possible. As previously stated, reformulating the turbine optimization subroutine is one focus of ongoing work.

EXAMPLES

Users have the option of working with OpenProp through the MATLAB command line or the graphical user interface (GUI). The GUI provides a parametric analysis interface for preliminary design and a single propeller interface (shown in figure 8) for detail design. Both modes take basic parameters such as the diameter, number of blades, shaft speed, ship speed, and required thrust. OpenProp generates a vortex lattice model of the propeller using these inputs, and optimizes the circulation distribution on this model. Using the parametric and single modes together, the user can design a propeller or turbine relatively quickly.

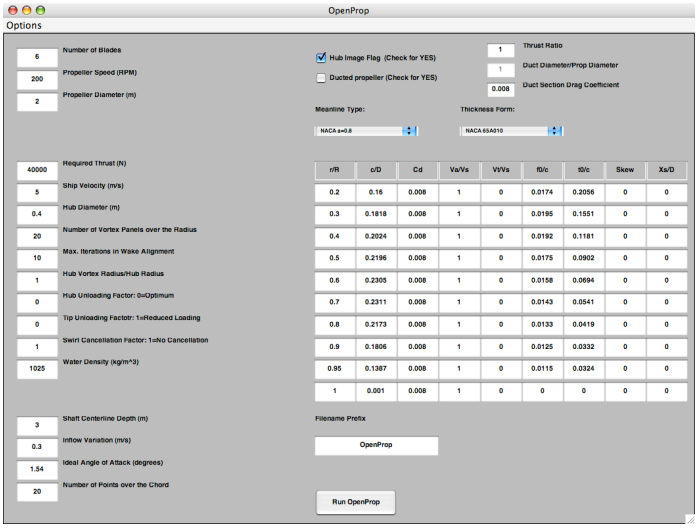


Figure 8: OpenProp propeller design graphical user interface

OpenProp's parametric design mode helps the designer tackle system-level design problems. Consider the ubiquitous problem of choosing a shaft speed: the designer may be given motor specifications and a choice between several gearboxes. His task is to compare the performance of several possible propeller designs appropriately matched to the available motor/gearbox combinations. Using the parametric design GUI, the designer can specify a range of acceptable propeller diameters, shaft speeds, and blade numbers. The parametric design mode then evaluates the performance of a propeller optimized to each of the specified design points. In this way, the designer can use OpenProp to explore the design space for a new propeller and choose suitable propeller specifications.

Once the design point is chosen, the designer then uses the single propeller design GUI (shown in figure 8) or the MATLAB com-

mand line to perform a more detailed design. The user inputs the selected system-level design specifications and can choose between a few options for the foil meanline and thickness profiles. Optimization can then be performed with or without an estimate of viscous losses. Once the optimization is completed, OpenProp produces several text and graphical reports detailing the performance and geometry of the design. Since the single propeller design mode is quite fast (less than one minute on a typical laptop computer), users can optimize and evaluate several competing propeller designs in this step.

OpenProp also produces command file scripts to automatically generate a NURBS propeller model in Rhino3D. This model can then be manipulated in Rhino3D, exported to another CAD program, or prepared for 3D printing or another computer-aided manufacturing (CAM) method.

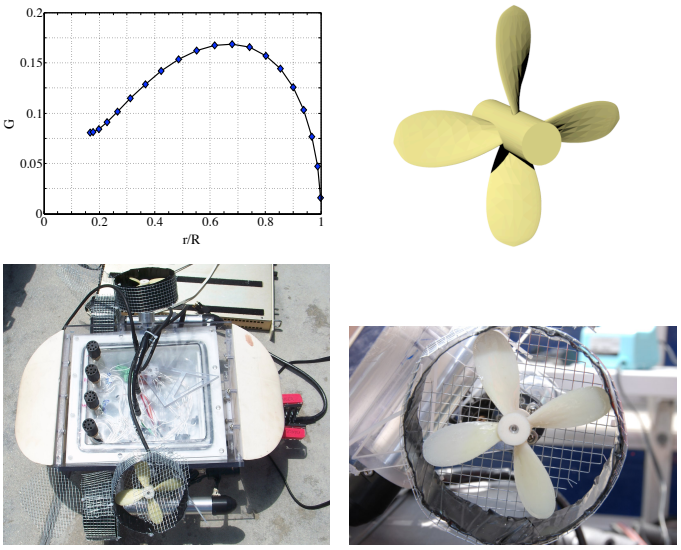


Figure 9: MIT ROV team propeller. Clockwise from top left: Non-dimensional circulation ($G = \frac{\Gamma}{2\pi R V_\infty}$) versus radius, where $R = 0.06$ [m] is the propeller radius and $V_\infty = 0.5$ [m/s] is the design ship speed; 3D CAD rendering of the propeller, using the output geometry from OpenProp; prototype propeller; ROV with caged propellers.

The MIT Remotely Operated Vehicle (ROV) Team used OpenProp to design and 3D print custom optimized propellers for their entry into the 2008 MATE ROV Competition (see figure 9). Using the parametric tools, they decided on a motor/gearbox combination with a shaft speed of 545 rpm. They then used the single propeller tools to design a four-bladed, 12 cm diameter propeller that gave 8.75 N of thrust at an advance velocity of 0.5 m/s. They used the OpenProp output to automatically build a NURBS model in Rhino3D, then generated a mesh and .stl file for a 3D printer. Four propellers were printed in an ABS/polycarbonate blend and ridges were smoothed over with epoxy. The raw printed propellers at this scale were somewhat flexible, so a better future approach might be to use the printed piece to make a mold for a stiffer material. Another alternative would be to lay up glass or carbon fiber, using the printed piece as a core. The team used these propellers successfully in pool missions at the ROV competition, but no performance tests have been done.

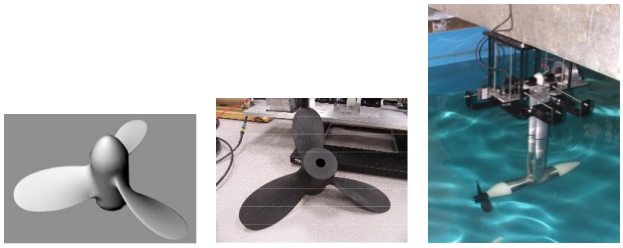


Figure 10: Computer rendering, 3D printed model and testing of OP4148.

D'epagnier (2007) designed and built a propeller to emulate the performance of the NAVY propeller 4148 (Kinnas 1995) as a test case for the OpenProp code. Figure 10 shows the propeller as designed and rendered in Rhino, the physical 3D printed propeller, and the propeller as it was undergoing tests at the tow tank at the University of Maine at Orono at the time of publication.

OpenProp is a continuing work in progress, but has reached the level of development where it has been useful to students not directly involved with the code. It brings the considerable power of vortex lattice analysis to the fingertips of novice and expert propeller designers with its friendly GUI and higher-power command-line interfaces.

CURRENT RESEARCH FOCUS

Efforts currently underway with the OpenProp code development include improvements to the turbine optimizer and the validation testing of OpenProp propeller and turbine designs. Future additions to OpenProp are being planned in the following areas:

- Integration of the 2D foil code XFOIL for arbitrary foil geometries and inclusion of viscous boundary layer effects,
- Integration of the cavitation bucket generator of Peterson (2008) into the OpenProp code,
- Addition of multiple blade row design capability,
- Addition of blade strength analysis capability,
- Extension of Blade outputs to other CAD programs such as SolidWorks, as well as internal generation of 3D Print files.

The goal of the OpenProp suite of codes is to provide accurate and powerful propeller and axial flow turbine design codes for use by both novice users and experienced designers. The open-sourced nature of the code (published under the GNU public License protocol) is intended to be a public resource to enhance the art of propeller and turbine design.

ACKNOWLEDGMENTS

This work is supported by the Office of Naval Research N000140810080, ESRDC Consortium and MIT Sea Grant College Program, NA06OAR4170019. In addition, the authors wish to thank Mr. Robert S. Damus of the Project Ocean, who was instrumental in securing a fellowship that made some of this research possible.

REFERENCE

Abbott, I. H., and Von Doenhoff, A. E. *Theory of Wing Sections*. Dover, 1959.

Carlton, J. S. *Marine Propellers and Propulsion*. Butterworth-Heinemann, 1994.

Chung, H.-L. "An enhanced propeller design program based on propeller vortex lattice lifting line theory". M.S. thesis, MIT, 2007.

Coney, W.B. "A Method for the Design of a Class of Optimum Marine Propulsors". PhD dissertation, Massachusetts Institute of Technology, Cambridge, MA, September 1989.

D'Epagnier, K.P. "A computational tool for the rapid design and prototyping of propellers for underwater vehicles". M.S. thesis, MIT/WHOI, 2007.

D'Epagnier, K.P.; Chung, H.-L.; Stanway, M.J.; and R.W. Kimball. "An Open Source Parametric Propeller Design Tool". *Oceans 2007*, p. 1-8, October 2007.

Drela, M. "Integrated Simulation Model for Preliminary Aerodynamic, Structural, and Control-Law Design of Aircraft." *AIAA SDM Conference*, 99-1394, 1999.

Khan, M.J.; Iqbal, M. T.; and J. E. Quaicoe. "Design Considerations of a Straight Bladed Darrieus Rotor for River Current Turbines". IEEE ISIE 2006, July 9-12, 2006, Montreal, Quebec, Canada

Kerwin, J.E. *Hydrofoils and Propellers*. MIT course 2.23 notes, 2007.

Kimball, R.W.; Epps, B.P.; and M.J. Stanway. OpenProp MATLAB code. Open-source at http://openprop.mit.edu

Kinnas, S.A. University/Navy/Industry Consortium on Cavitation of High Speed Propulsors, fifth meeting, June 1st and 2nd, 1995.

Lerbs, H.W. "Moderately Loaded Propellers with a Finite Number of Blades and an Arbitrary Distribution of Circulation." *Trans. SNAME*, v. 60, 1952.

Peterson, C.J. "Minimum Pressure Envelope Cavitation Analysis Using Two-Dimensional Panel Method" Masters Thesis, MIT, June 2008.

Stewart, H.J. "Dual Optimum Aerodynamic Design for a Conventional Windmill". *AIAA Journal*, v. 14, no. 11, p. 1524-1527, 1976.

Stubblefield , J.M. "Numerically Based Ducted Propeller Desgin using Vortex Lattice Lifting Line Theory", Masters Thesis, MIT, June 2008.

Wrench, J. W. "The calculation of propeller induction factors." Technical Report 1116, David Taylor Model Basin, February, 1957.