

Investigation and Optimization of Blade Tip Winglets Using an Implicit Free Wake Vortex Method

Stephen Lawton¹ and Curran Crawford²
University of Victoria, Victoria, British Columbia, Canada

Abstract.

Novel outer-blade geometries such as tip winglets can increase the aerodynamic power that can be extracted from the wind by tailoring the relative position and strengths of trailed vorticity. This design space is explored using both parameter studies and gradient-based optimization, with the aerodynamic analysis carried out using LibAero, a free wake vortex-based code introduced in previous work. The starting design is the NREL 5MW reference turbine, which allows comparison of the aerodynamic simulation for the unmodified blade with other codes. The code uses a Prandtl-Weissinger lifting line model to represent the blade, and vortex filaments as the flow elements. A fast multipole method is implemented to accelerate the influence calculations and reduce the computational cost. This results in higher fidelity aerodynamic simulations that can capture the effects of novel geometries while maintaining sufficiently fast run-times (on the order of an hour) to allow the use of optimization. Gradients of the objective function with respect to design variables are calculated using the complex step method which is accurate and efficient. Since the vortex structure behind the rotor is being resolved in detail, insight is also gained into the mechanisms by which these new blade designs affect performance. It is found that adding winglets can increase the power extracted from the wind by around 2%, with a similar increase in thrust. It is also possible to create a winglet that slightly lowers the thrust while maintaining very similar power compared to the standard straight blade.

1. Introduction

There has been a great deal of research investigating the effects of adding winglets to plane wings, see for example references [1, 2, 3]. However there is less literature considering the potential benefits of adding winglets to a rotor. Van Bussel [4] used momentum theory to investigate HAWT winglets, and also carried out a significant amount of experimental work. He explains the power increase to be from the downstream shift of the wake vorticity. Imamura *et al.* [5] used a free-wake vortex-lattice code to investigate the aerodynamic effects of winglets on wind turbines. Johansen [6] carried out Navier-Stokes based simulations using the CFD code EllipSys3D, and suggests that the shift in vorticity alone does not account for the power gains from the winglet, the reduction of tip effects being the dominant feature. Reduction in tip effects here means a reduction in spanwise flow and a diffusing effect on the tip vortex, this results in a reduction in downwash on the main blade and so reduced induced drag, increasing power production. Gaunaa and Johansen [7] found that adding a downwind winglet increased power more than an upwind winglet. They also found that the downwind winglet always had negative power production, and



shifted the tip vortex outwards radially, increasing the loading on the outer part of the main blade. In [8] it was found that winglets could be analysed using a prescribed wake method, but only when the model is calibrated with a free wake model and the load distribution is not changed too greatly. Chattot [9] used a Goldstein vortex model with an optimizer to look at blade tip modifications, also finding that backwards sweeping and downwind winglets move the tip vortex away from the blade, increasing the power captured.

It is clear that more needs to be done to design an optimum blade featuring winglets. The analysis here is performed using *LibAero*, a free-vortex wake aerodynamic modelling library developed at the University of Victoria [10]. It has been developed with the purpose of providing a fast and accurate aerodynamic analysis of a wind turbine for use in optimization. This means it must produce accurate results across the full design space, so it must be able to deal with novel geometries such as wingletted and swept rotors. This makes BEM unsuitable, since the effect of such novel geometries on the overall performance is obscured by the number of empirical and semi-empirical correction factors inherent in the method, and the fact that non-straight blades violate some of the BEM assumptions. The other main consideration in the development of *LibAero* is the computational expense, or run-time of the simulations. Optimization requires a very large number of design iterations in order to produce an optimal design solution, hence the widespread use of BEM for design work. This makes CFD impractical unless large amounts of computing power are available. The need to set up a mesh with most CFD approaches also makes it very difficult to iterate across a large design space with different geometries when taking the CFD route. It is also necessary to evaluate designs in the time domain for fatigue analysis and to assess unsteady performance, which is extremely expensive using CFD. CFD however does not require the use of two-dimensional airfoil data, since the blade can be fully resolved if desired.

2. Aerodynamic Modelling

2.1. Governing Equations

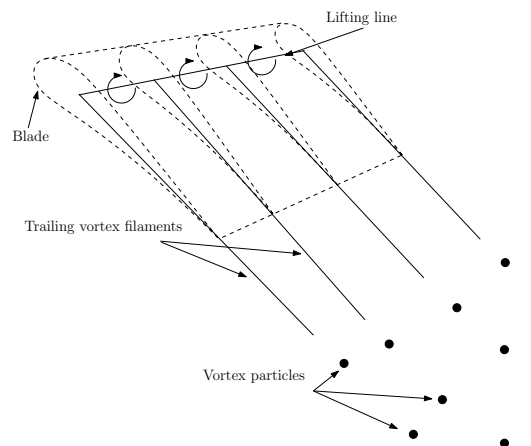


Figure 1. Blade showing the lifting line at the quarter chord position, trailing vortex filaments and vortex particles (particles not used in this study due to discontinuities caused in gradient information) in wake

The model uses a Weissinger lifting line approximation of the blades, as shown in figure 1. This is used since accurately modelling the boundary layer on the blade would contradict the aim of having a computationally fast method to use in an MDO tool. The lifting line approximation is valid for high aspect ratio wings and blades, which is certainly the case for a wind turbine blade. In this approximation, two dimensional airfoil

data (taken from experiments) based on an angle-of-attack and Reynolds number lookup table are used to compute forces. No corrections are made for three-dimensional or rotational effects. These forces are applied at the quarter chord point of the blade, using vortex filaments with circulation calculated according to the Kutta-Joukowski lift equation:

$$\Gamma = \int_A \boldsymbol{\omega} \cdot d\mathbf{A} = \frac{cVc_l}{2}. \quad (1)$$

The Eulerian equation governing the flow is the velocity-vorticity formulation of the Navier-Stokes equations:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \mathbf{u} + \nu \nabla^2 \boldsymbol{\omega} \quad (2)$$

where the velocity \mathbf{u} can be written using a scalar and a vector potential via the Helmholtz decomposition $\mathbf{u} = \nabla \phi + \nabla \times \boldsymbol{\psi}$ as:

$$\frac{\partial}{\partial t} \nabla^2 \boldsymbol{\psi} + ((\nabla \phi + \nabla \times \boldsymbol{\psi}) \cdot \nabla)(\nabla^2 \boldsymbol{\psi}) = (\nabla^2 \boldsymbol{\psi} \cdot \nabla) \nabla \phi + \nu \nabla^4 \boldsymbol{\psi}. \quad (3)$$

The terms in equation (3) represent local change in time, vorticity advection, vortex strength deformation and diffusion respectively. The vortex method is defined in a Lagrangian reference frame, by separating vortex element advection, diffusion and strength deformation equations, which may be written respectively for a point vortex element as

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}_{xt} \quad (4)$$

$$\frac{d\boldsymbol{\omega}}{dt} = \nu \nabla^2 \boldsymbol{\omega}_t \quad (5)$$

$$\frac{d\boldsymbol{\alpha}_t}{dt} = \boldsymbol{\alpha}_t \cdot \nabla u_{xt} \quad (6)$$

In this study we use vortex filaments for the computational elements in the wake, which obviates the need for the strength deformation term to be explicitly calculated. The use of transitioning to vortex particles was found to introduce discontinuities in the solution space, and so caused issues with obtaining objective function gradients to use in an optimization. The induced velocity field due to all the flow elements is calculated using the Biot-Savart law:

$$\mathbf{u}_x = \frac{1}{4\pi} \int_{\tilde{\mathbf{x}} \in V} \frac{\sigma_{\tilde{\mathbf{x}}}(\mathbf{x} - \tilde{\mathbf{x}}) + \boldsymbol{\omega}_{\tilde{\mathbf{x}}} \times (\mathbf{x} - \tilde{\mathbf{x}})}{|\mathbf{x} - \tilde{\mathbf{x}}|^3} dV. \quad (7)$$

where $\sigma_{\tilde{x}}$ represent sources and sinks which can be used to represent non-lifting drag objects in the simulation, such as a turbine hub or tower. Computing this integral can be greatly accelerated using the Fast Multipole Method (FMM) [11], where a tree structure is created of all the flow influences. This means that far away influences can be lumped together into a leaf node, thus reducing the complexity of the N-body problem from $O(N^2)$ to almost $O(N)$, a huge saving when there are thousands of vortex elements. In the *LibAero* implementation of the FMM algorithm a binary tree is used as the data structure. This differs from the normal octree data structure (in three dimensions) where cubes are successively divided into eight sub-cubes. The binary partitioning method is a more natural choice for a wind turbine wake given the natural oblong structure [12, 10].

3. Computational Methods

The Lagrangian governing equations used are inherently time-dependent. This means that a steady state (or periodic) solution is found by evolving the solution in the time domain from an initial wake state until convergence has been achieved. The first implementation of *LibAero* used the standard viscous splitting method with separate advection, diffusion and vortex strength deformation (the latter not necessary for vortex filaments since it is implicitly handled as the filament's length changes) steps. Each operation is performed successively; the flow elements are advected, then diffused then deformed if necessary at each time step. This method is widely used and has been shown to be a consistent and convergent approximation to the Navier-Stokes equations provided the sub-models are valid [13] [14], however it is only first-order accurate in time (the error term is of the order $(\Delta t)^2$) as a direct result of this splitting approach [15]. The advection and diffusion operations are now solved simultaneously, this is possible when the problem is solved with an implicit time-marching approach rather than explicit.

3.1. Advection Modelling

An implicit Crank-Nicolson discretization is used for the vortex element advection equation:

$$\mathbf{x}_{t+\Delta t} - \frac{\Delta t}{2} \mathbf{u}_{\mathbf{x}(t+\Delta t)} = \mathbf{x}_t + \frac{\Delta t}{2} \mathbf{u}_{\mathbf{x}(t)} \quad (8)$$

This equation cannot be solved explicitly for the new positions $\mathbf{x}_{t+\Delta t}$. The velocities $\mathbf{u}_{\mathbf{x}(t+\Delta t)}$ defined at the new element locations $\mathbf{x}_{t+\Delta t}$ are dependent on both the new location of that element, but also on the new locations of all other elements through the Biot-Savart integral expression for velocity. This non-linear system of equations must therefore be solved iteratively.

To do so we can define $\mathbf{F}(\mathbf{x}_{t+\Delta t})$

$$\mathbf{F} = -\mathbf{x}_{t+\Delta t} + \frac{\Delta t}{2} \mathbf{u}_{\mathbf{x}(t+\Delta t)} + \mathbf{x}_t + \frac{\Delta t}{2} \mathbf{u}_{\mathbf{x}(t)} \quad (9)$$

The aim is then to find a solution to $\mathbf{F}(\mathbf{x}_{t+\Delta t}) = 0$. Broyden's method [16] is used. This method is similar to the Newton-Raphson method:

$$\mathbf{x}_{t+\Delta t}^{n+1} = \mathbf{x}_{t+\Delta t}^{n+1} - (J^n)^{-1} \mathbf{F}(\mathbf{x}_{t+\Delta t}^n) \quad (10)$$

where J denotes the Jacobian matrix of \mathbf{F} with respect to $\mathbf{x}_{t+\Delta t}$. The difference between this method and a Newton-Raphson is that instead of re-evaluating the Jacobian after every updated $\mathbf{x}_{t+\Delta t}$ followed by inverting this matrix, it is instead updated via the Sherman-Morrison formula [17]:

$$(J^n)^{-1} = \frac{(J^{n-1})^{-1} + \Delta \mathbf{x}_{t+\Delta t} - (J^{n-1})^{-1} \Delta \mathbf{F}^n}{\Delta \mathbf{x}_{t+\Delta t}^T (J^{n-1})^{-1} \Delta \mathbf{F}^n} (\Delta \mathbf{x}_{t+\Delta t}^T (J^{n-1})^{-1}) \quad (11)$$

where

$$\Delta \mathbf{x}_{t+\Delta t} = \mathbf{x}_{t+\Delta t}^n - \mathbf{x}_{t+\Delta t}^{n-1} \quad (12)$$

$$\Delta \mathbf{F}^n = \mathbf{F}^n - \mathbf{F}^{n-1} \quad (13)$$

This has the advantage that the Jacobian matrix is not evaluated more times than is necessary, a significant computational saving since the Jacobian is $O(N^2)$ to populate. There is also a significant cost to inverting this matrix depending on the method used (the open-source C++ matrix library Eigen is used for linear algebra operations within the code). Equations (10) and (11) can then be iterated until convergence is achieved. Convergence can be determined by evaluating the value of \mathbf{F} in equation (9). Since we are solving $\mathbf{F} = \mathbf{0}$, when some norm of \mathbf{F} becomes less than an acceptable tolerance (the tolerance used in this work was 10^{-15} , this was found to be necessary to give smooth gradients for use in optimization) the method is converged. It is sometimes necessary to re-evaluate the Jacobian several times before the residual will drop below an acceptable tolerance, this typically occurs at higher element densities when there is more vortex filament roll-up occurring and more pronounced non-linearities in the simulation, this is performed as needed to meet the convergence criterion for that time-step. Overall, steady-state convergence is achieved by time-stepping until an iterative convergence criterion is met (based on difference between blade loadings at successive time steps).

The starting point $\mathbf{x}_{t+\Delta t}^0$ for the new positions of elements at the new timestep can be determined in several ways. To speed up convergence it makes sense to estimate the new positions using an explicit method such as a forward Euler and use this as the initial condition for the implicit solve (the implicit solve then acts as a correction to the forward Euler method, making this approach very similar in nature to a predictor-corrector method.) The classical Runge-Kutta method (RK4) was used earlier in this research and was a significant improvement over a simple explicit scheme, however it was found to still produce gradients that were not smooth enough for the optimizer to reliably converge. It should be emphasised that the use of different discretization schemes is of much greater concern when using a vortex code within a gradient-based optimizer, rather than simply evaluating a design.

This is clearly going to be more expensive to solve at each time step than an explicit scheme such as the Forward Euler method. The advantage of using such an implicit method is in significantly improved stability and gradient quality, so larger timesteps can be used without problems in a time-domain simulation. In this work 20 deg azimuthal resolution was used for the wake. With an explicit method error accumulates with each time step, when using an implicit method the residual and hence error at each time step can be driven

arbitrarily low. Explicit methods are subject to well-documented stability problems, particularly over long time intervals, and this can prevent iterative convergence from being reached, as well as making the solution too noisy to obtain accurate solution gradients with respect to design variables that are needed to perform optimization.

This time-stepping approach to obtaining a solution is used instead of formulating the problem as steady-state, so the wake is populated as the simulation progresses through time. A far wake model is also used, consisting of concentric vortex tubes that begin where the resolved wake ends. In this work a maximum of six wake revolutions are resolved for each filament chain, then the wake transitions into the far wake model. Filaments beyond six wake revolutions are deleted from the simulation, while new filaments are created at the blade. The far-wake model takes its circulation components and radius from the final wake revolution, and so changes with each iteration as the resolved portion of the wake develops. At each timestep the bound circulation on the blade will change, this is reflected in the wake by updating the circulation entire filament chain and far-wake model at each timestep, consistent with Kelvin's theorem.

3.2. Viscous Effects Modelling

In this work we employ the classical core spreading method of Leonard [18]. This method has been used extensively, and has many advantages: it is simple to implement and computationally fast. The approach consists of increasing the vortex' viscous core size in such a way as the Laplacian term in the governing equation (5) is satisfied. With a Gaussian core model this is equivalent to increasing the variance σ_t , according to $\sigma_{t+\Delta t} = \sqrt{\sigma_t^2 + 2\nu\Delta t}$, since the diffusion equation prescribes a linear increase in variance. The method is valid when Reynolds numbers are large (ideally above 10^{-5}), and hence a reasonable simplification for the flow around a wind turbine. The core model used here is the Van Garrel [19]. This consists of a small de-singularization parameter $r_c^2 l_0^2$ (where r_c is the viscous core radius and l_0 being the length of the filament) being added to the denominator of the Biot-Savart influence equation (7) to prevent infinite velocities occurring at the elements.

3.3. Optimization

There is a trade-off between speed and robustness when considering optimization algorithms. Many heuristic techniques are capable of finding global optimum solutions within a non-convex space, with a mix of discrete and continuous design variables. Gradient-based algorithms are significantly faster, but can only be applied if the objective is smooth and the design space is continuous. Gradient-based methods are sensitive to the starting point and can get trapped in local optima. In this work a gradient-based algorithm is used since wind turbine blades can be described by continuous variables and is much more efficient considering the computational cost of the aerodynamic code. The CFSQP algorithm provided by Lawrence *et al* [20] is used, this is a variant of Sequential Quadratic Programming algorithms; see Vanderplaats for more details [21].

The complex step method is used to calculate the gradients needed by CFSQP. Instead of stepping the real component of a design variable (as would be done to calculate a finite difference gradient) the imaginary component of the design variable is adjusted by a small step h which then propagate through complex numbers used to represent all other state variables and parameters in the computations. The complex step method is shown in equation 14. The main advantage of this method over using a finite difference approximation is that it does not introduce subtractive cancellation errors, and only one function evaluation is necessary to obtain both the objective function evaluation (the real component of the result is unchanged provided h is sufficiently small) and the gradient of the objective function with respect to that

design variable. The complex step method can however introduce extra computational overhead, since the aerodynamic model is now evaluated using complex arithmetic rather than real.

$$\frac{\partial f}{\partial x} \approx \frac{\text{Im}[f(x + ih)]}{h} \quad (14)$$

The converged solution from the first design variable step can be used to initialise the simulation for the next design variable to quickly get gradients with respect to all design variables, due to negligible feedback from the complex component into the real part of computations.

4. Results

In this section the NREL 5MW turbine is used as the baseline design. All results are obtained for a windspeed of 12m/s (the turbine's rated windspeed)

4.1. Code Validation

LibAero has been validated against experimental data in previous work [22, 10]. As an illustration of the code's validity for this turbine, the unmodified NREL 5MW turbine was simulated and the results are shown in the table below along with the results from FAST [23] for a windspeed of 12m/s , corresponding to a tip speed ratio of 7.

	Power (kW)	Thrust (kN)
FAST + AERODYN	5297	690
LibAero N=15	4734	623
LibAero N=35	4786	634
LibAero N=55	4812	641

It can be seen from this table that both rotor power and thrust are underpredicted by *LibAero* as compared with the FAST results. Here a 20 deg azimuthal resolution was used for the 15 blade element case, and this azimuthal resolution was also scaled with the number of blade elements N to illustrate the grid convergence properties of the method. It is not necessary however to keep these two dimensions consistent (such as a CFL condition in a CFD method), and a more detailed convergence study varying these resolution parameters can be found in [22]. Increasing the resolution of the simulation does give closer agreement. Differences are expected however due to the fundamentally different nature of the models, and no experimental results exist for this turbine for further comparison purposes. To determine the optimum resolution to use a convergence study with respect to N was also carried out. Figure 4.1 shows how the final predicted power and thrust vary with resolution for the baseline blade and for two different winglets. As can be seen from these figures, while the values of thrust and power are changing as the resolution is increased (indicating that we do not have full grid convergence) the relative differences between wingletted and non-wingletted designs are very similar. This suggests that it is valid to use *LibAero* at the lower resolution to explore the winglet design space, as the relative trends are most important here. This is important since the computational time increases rapidly at higher resolutions. At higher resolutions there is more self-influence from the wake, and wake roll-up becomes pronounced. This means that each iteration takes a lot longer to solve due to the non-linearities in the system, and the inner loop implicit solve requires a lot more iterations to

converge. With 15 blade elements a typical function evaluation takes under an hour to converge to a steady state solution, with 55 elements this is increased to around 12 hours. Steady state here means that wake velocities evaluated at 20 points in the near wake have converged to a tolerance of $10^{-10}m/s$, the 2-norm is used here. Tight stopping tolerances must be used in order for accurate gradients to be calculated, on-blade performance metrics are converged well before gradient information has converged. As with all of the results presented here, six wake revolutions were resolved before transitioning into the far-wake model, and once the resolved wake is populated it continues to evolve until convergence is reached.

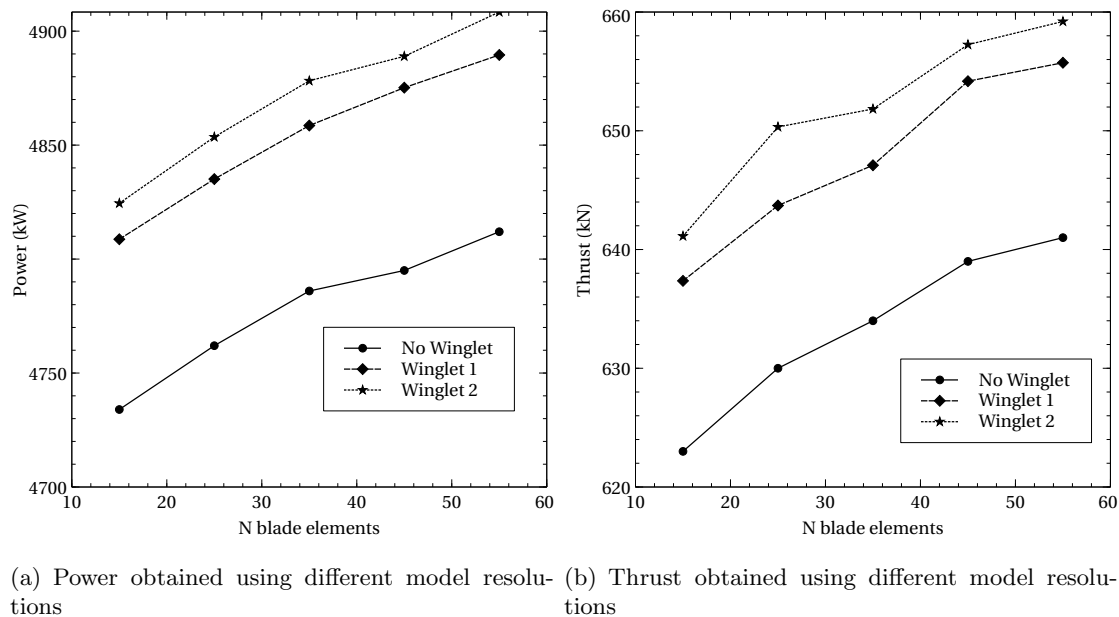


Figure 2. Power and thrust varying with resolution, baseline blade and 2 different winglets

4.2. Initial Parameter Study

To gain insight into the winglet design space a parameter study was carried out. Figure 3 shows how the power and thrust of the turbine varies with winglet length for simple winglets added at 90 deg to the main blade. Here a positive winglet length indicates it is purely in the upstream (upstream with respect to wind direction, not turbine rotation) direction, and negative indicates a downstream winglet. Both upstream and downstream winglets increase power as well as thrust, with the downstream design increasing power the most. So it appears that there is real benefit to adding a winglet of this type.

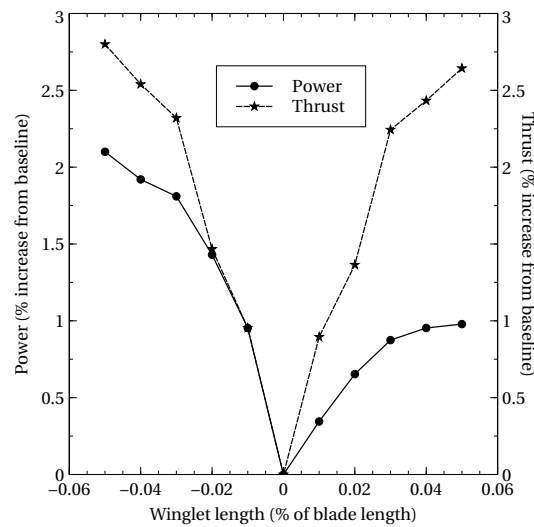


Figure 3. Power and thrust for different winglet lengths, winglets at 90 deg to main blade

4.3. Optimization Results

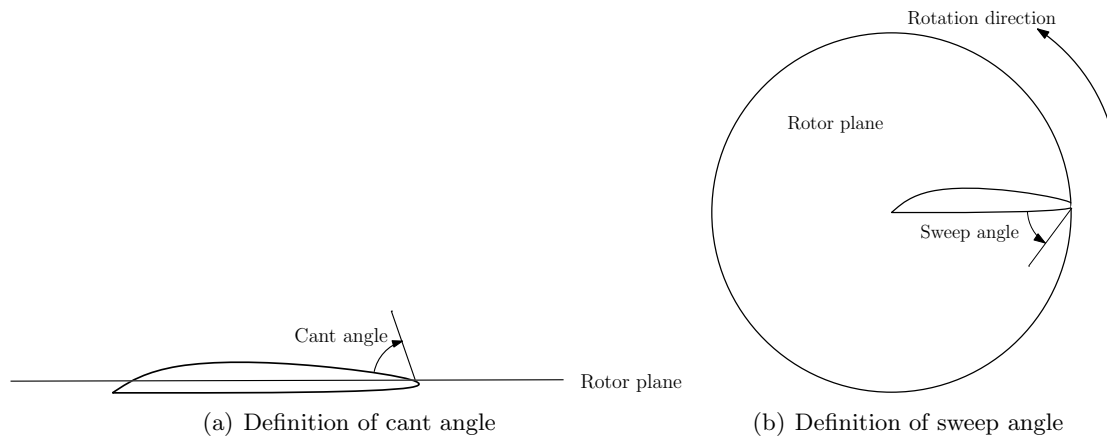


Figure 4. Definition of winglet angles

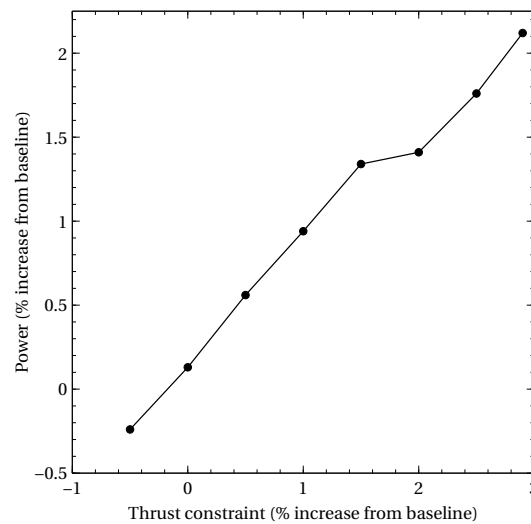
Optimization was carried out using 5 design variables. These are the winglet length, winglet sweep angle, winglet cant angle, winglet root chord and winglet root twist. Cant angle and sweep angle are shown in figure 4. Both chord and twist of the winglet is specified to vary linearly from the root value (winglet root is at the tip of the blade where the winglet begins) to zero at the winglet tip. This was done simply to minimize the number of design variables, and hence computational cost but could be included in future studies.

When the objective function is maximum power, with no other constraints imposed on the problem, the optimal design simply uses the winglet to extend the blade, maximizing the swept area. When power coefficient is used as the objective function, the optimal design also uses the possible winglet to simply extend the blade, up to the winglet length limit of 5% blade length. However, when total rotor radius is constrained (meaning the winglet cannot extend beyond the blade tip), the optimal design obtained is a winglet pointing in the downstream direction (90 degree cant angle), with zero sweep angle (sweep is used

Winglet	1	2	3
Thrust constraint (% increase over baseline blade)	No constraint	1.5%	0%
Length (% blade length)	-0.05	-0.0353	0.1453
Sweep angle (deg)	0	5.323	42.34
Cant angle (deg)	90	78.54	74.54
Root chord (m)	1.742	1.324	0.974
Root twist (deg)	6.733	8.28	26.37
Power (% increase over baseline blade)	2.13	1.33	0.132

Table 1. Optimal winglet design parameters for varying thrust constraint

here to mean the component of the winglet that is in the rotor plane, i.e. a swept winglet with 90 deg sweep angle would be purely in the rotor plane. No changes to the main blade are allowed in this design space, we are considering the addition of winglets only). This design gave an increase in power of 2.1%, and an increase in thrust of 2.8%.

**Figure 5.** Maximum power for varying thrust constraint

While increasing power is advantageous, if it comes at the cost of greatly increased thrust then the increased structural loads can outweigh the gains, since the structure must be stronger and hence more expensive. To this end, a constraint on thrust was introduced, as well as the previous constraint on rotor size. The thrust constraint becomes active as soon as it is set below the previously obtained value of 2.8% increase over the baseline thrust. Figure 5 shows how the optimal power output is affected by the thrust constraint. It can be seen that it is possible to slightly increase power (by 0.132%) without allowing thrust to increase. The designs obtained for several values of thrust constraint (compared with the original baseline thrust) are shown in table 1.

To understand the mechanism by which a winglet can increase power production, figure 6 shows the local power coefficient for the original blade and three different winglets. The winglets used are (in the same order presented in the legend of figure 6) the unmodified blade, a purely downwind winglet (winglet 1 in table 1), a purely upwind winglet that is otherwise identical to the downwind winglet, and the winglet obtained with a strict thrust constraint (winglet 3 in table 1). It can be seen here that there are two distinct types of winglet, the first type has increased power generation from the tip region of the main blade, with

a negative contribution to power from the winglet (mainly due to its induced drag). The downwash from the bound circulation on this type of winglet increases the loading on the outboard portion of the blade, and this outweighs the negative effects from drag on the winglet itself. The purely downwind winglet is of this type, as is the optimal winglet obtained when thrust is constrained to its original value. The upwind winglet acts differently however. Here the winglet itself has a positive contribution to rotor power, and the downwash from the winglet slightly reduces the loading on the outer portion of the blade. This also leads to an increase in power, although not as great as for the downwind winglet as can be seen from figure 3.

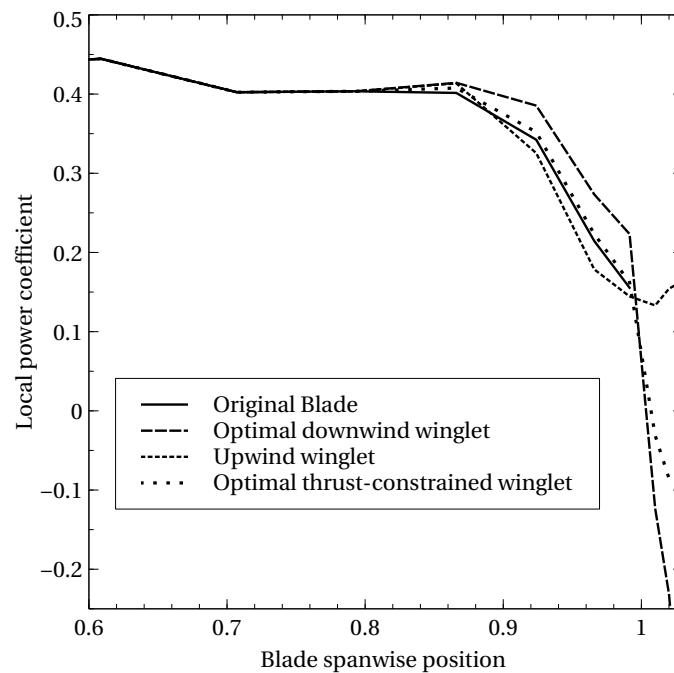


Figure 6. Local power coefficient for several different winglet designs. Blade spanwise positions above 1 indicate stations that are on the winglet and not the main blade

5. Conclusion

These results demonstrate that there are applications of winglets in optimal turbine design, particularly when there are constraints on variables such as rotor diameter (as could be the case in a wind farm environment or constraints due to transportation). It is found that adding winglets can increase the power extracted from the wind by around 2%, with a similar increase in thrust. It is also possible to create a winglet that slightly lowers the thrust while maintaining very similar power compared to the standard straight blade. Winglets can work either by contributing to the rotor power, or by increasing the power generated on the outer portion of the blade, depending on the design.

To extend this preliminary study, the aerodynamic analysis will be carried out for a range of wind speeds and at a higher fidelity, with a full grid convergence study to ensure that relevant effects and phenomena are being captured. It is expected that increased resolution will be needed as the tip speed ratio increases. Higher tip speed ratio means there is more self-influence from the wake, and so the vortex structure is inherently more complex to simulate accurately. In this case more vortex roll-up occurs, and when using straight filaments the resolution must be sufficiently high to allow this roll-up to be represented. The number of blade elements must also be sufficient to allow the winglet to be represented accurately, or the interactions

cannot possibly be resolved. It is expected that between 50 and 100 blade elements will be needed for the simulation to be deemed grid-converged.

More complex winglet geometries and the associated increased number of design variables will be considered. To fully capture the tip effects greater resolution is needed; this was not possible for this paper due to computational time restrictions. It should also be expected that the optimum configuration would involve different end geometry on the main blade, since the winglet is changing the flow in this tip region. Optimization should therefore include the main blade as part of the design process (or at least the tip region) in order to find an effective design. It is also planned to consider structural constraints as part of the optimization.

Acknowledgments

The authors express their appreciation for the funding of this work through a University of Victoria Fellowship, BC Ministry of Advanced Education and NSERC.

- [1] Munk M 1923 The minimum induced drag of aerofoils Tech. rep. NACA Report 121
- [2] Whitcomb R 1976 A design approach and selected wind tunnel results at high subsonic speeds for wing-tip mounted winglets Tech. rep. NASA TN D-8260
- [3] Eppler R 1997 *Aerospace Science and Technology* **1** 3–15
- [4] van Bussel G 1990 (4th IEA symposium on the aerodynamics of wind turbines)
- [5] Imamura H e a 1998 *JSME international journal* **41** 170–176
- [6] Johansen J and Sorensen N 2007 (EWEC Conference Proceedings, Madrid, Spain)
- [7] Gaunaa M and Johansen J 2008 (46th AIAA Aerospace Sciences Meeting, Reno, Nevada)
- [8] Gaunaa M, Sorensen N and Dossing M 2011 (49th AIAA Aerospace Sciences Meeting, Orlando, Florida)
- [9] Chattot J 2009 *Computers and Fluids* **38** 1405 – 1410
- [10] Cline S 2010 *Potential Flow Modelling for Wind Turbines* Master's thesis University of Victoria
- [11] Greengard L and Rokhlin V 1987 *Journal of Computational Physics* **73** 325–348
- [12] Cline S, McWilliam M, Lawton S and Crawford C 2011 *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*
- [13] Beale J and Majda A 1981 *Mathematics of Computation* **37** 243–259
- [14] Cottet G H and Koumoutsakos P 2000 *Vortex methods: Theory and practice* (Cambridge University Press)
- [15] Evje S and Karlsen K H 1997 *Preprint, Institute of Mathematics and its Applications (IMA), University of Minnesota*
- [16] Broyden C G 1965 *Mathematics of Computation* **19** 577–593
- [17] Golub G H and Van Loan C F 1996 *Matrix Computations* (Johns Hopkins, Baltimore, MD)
- [18] Leonard A 1980 *JOURNAL OF COMPUTATIONAL PHYSICS* **37** 289–335
- [19] van Garrel A 2003 *Development of a Wind Turbine Aerodynamics Simulation Module* Ph.D. thesis ECN
- [20] Lawrence C, Zhou J L and Tits A L 1997 User's Guide for CFSQP Version 2.5: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints Tech. rep. Electrical Engineering Department and Institute for Systems Research, University of Maryland College Park, MD 20742 institute for Systems Research TR-94-16r1
- [21] Vanderplaats G N 2001 *Numerical Optimization Techniques for Engineering Design. 3rd ed.* (Colorado Springs: Vanderplaats Research & Development Inc.)
- [22] Lawton S and Crawford C 2013 (51st AIAA Aerospace Sciences Meeting)
- [23] J Jonkman S Butterfield W M and Scott G 2009 Definition of a 5-mw reference wind turbine for offshore system development Tech. rep. NREL TP-500-38060