Propeller Optimization at Different Blade Counts

Joseph Spencer *
Brigham Young University, Provo, Utah, 84601

propellers can be modified to perform better under a variety of conditions. One project completed as part of my research for the Brigham Young University FLOW Lab explores ideal design variables for propeller performance. This investigation used computer simulations to find the optimal chord length magnification and the optimal pitching angle for a propellor's efficiency. This report defines the operating conditions used in this optimization, presents propellors that were optimized at several blade counts, and discusses areas of possible future development in this research. I found that for the operating conditions used, propellors with the chord reduced to the optimization's lower limit, 50 percent, with negative pitching angles were the most efficient. These results could be applied to propellors designed to produce the maximum output power for a given input power requirement, without regard to other desired outputs. Future work could combine these findings with other conditions to find the most efficient propellor that can also provide a required amount of thrust or operate at a variety of different rotational velocities.

I. Introduction

Propellors come in a variety of different shapes and sizes. This paper describes how I used one propeller, an APC 10x7 rotor* with a NACA 4412 airfoil†, as a baseline to optimize a propeller's efficiency at a specific advance ratio. Because different blade counts could be more appropriate for different applications, I optimize and compare blade counts of one, two, and three in this report. I also compares the optimized propellors of each different blade count.

Computer simulations are very useful for modeling propellers. It would require significant investment of time, space, and money to perform an optimization like this by physically propellers. In the simulation used in this report, I instead computationally compared the properties of hundreds of slightly different propellers to obtain a final result. The first potential-flow codes, programs for modeling the pressure distribution of rotors, were developed in the 1960s and 1970s. Xfoil, a program used to create the airfoil model used in this project, was first developed by MIT in the 1980s, with its current version dating to 2013. Xfoil.jl was developed by Taylor McDonnell.‡

*Undergraduate Researcher, Brigham Young University FLOW Lab.

*Rotor geometry obtained from the UIUC database at `https://m-selig.ae.illinois.edu/props/volume-1/propDB-volume-1.html`

†propeller geometry obtained from supporting information for CCBlade.jl at `https://github.com/byuflowlab/CCBlade.jl/tree/master/data`

‡More information about both Xfoil and Xfoil.jl is available at `https://flow.byu.edu/Xfoil.jl/stable/` and `https://en.wikipedia.org/wiki/XFOIL`

This paper first reviews the procedure I used in the rotor optimization, including the objective function used in the optimization and the initial conditions and constraints applied. Next, I review the results and compare them with each other. Finally, I draw conclusions based on the results of this optimization and discusses their meaning. All code used for each section of this report can also be accessed through a GitHub repository.[§]

II. Methods

Computational stimulations of propellers allow researchers to perform low-risk studies and develop designs more quickly. I performed this project using Julia programming language.[¶] Jula is available for free and is useful for a variety of reasons. Like other languages, it can easily store data in vectors and perform rapid calculations with these objects. It is compiled just-in-time with precompiled functions, which makes it run faster than interpreted languages like python without needing to be totally compiled before running like C[‖]. Julia packages, including those used during this semester, are also easily available from a package manager.

A. Models

Some Julia packages used for this project include Xfoil.jl,[**] CCBlade,jl,[††] and SNOW.jl.[‡‡] Xfoil.jl analyzes the lift, drag, and moment coefficients of an airfoil from its geometry and angle of attack using a potential-flow code with a panel method. This divides the airfoil into many small surfaces and finds the lift, drag, and moment forces at each of the smaller surfaces.

CCBlade.jl uses blade element momentum (BEM) theory.[§§] BEM theory, which combines blade element theory and momentum theory, calculates the axial and tangential induction factors and the twist angle, the angle between the free stream velocity and the tangential velocity of the propeller. After these three quantities are found, they are combined to find forces acting in all directions on the propeller[1].

The propeller in this design was created beforehand and included in the documentation for CCBlade.[¶¶] Airfoil data could be created using XFoil.jl, but I selected this data because its domain had already been widened using Viterna extrapolation to include a complete range of angles of attack[***]. The airfoil polars used to analyze this are presented in the following figure.

[§] This repository can be accessed in the Rotor-Design branch of `https://github.com/JoeSpencer1/497R-Projects/tree/Final-Report`

[¶] Julia is available at `https://julialang.org`.

[‖] This article describes some of the benefits of Julia. `https://towardsdatascience.com/5-ways-julia-is-better-than-python-334cc66d64ae`

[**] Xfoil.jl is available at `https://github.com/byuflowlab/Xfoil.jl`

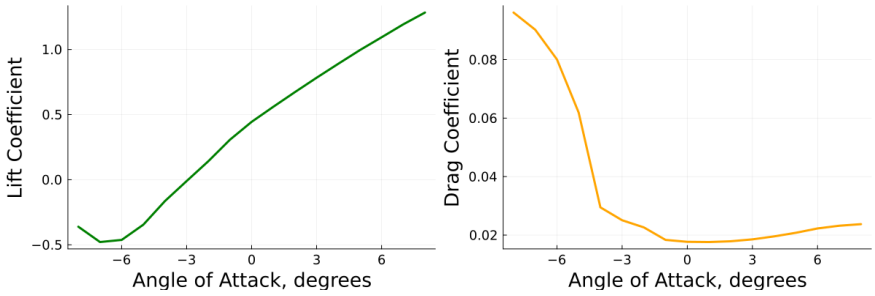
[††] CCBlade.jl is available at `https://github.com/byuflowlab/CCBlade.jl`

[‡‡] SNOW.jl is available at `https://github.com/byuflowlab/SNOWl.jl`

[§§] More information about BEM theory can be found at `https://en.wikipedia.org/wiki/Blade_element_momentum_theory`

[¶¶] Same address listed in introduction note, `https://github.com/byuflowlab/CCBlade.jl/tree/master/data`

[***] Further explanation of this method is available at `https://flow.byu.edu/CCBlade.jl/stable/howto/`



(a) lift coefficient **(b) drag coefficient**

Fig. 1 Lift and drag experienced by NACA 4412 airfoils
These plots show how lift and drag change for an airfoil at different angles of attack.

I then placed the airfoil on the APC 10x7 rotor used in the project, and evaluated its torque and power coefficients across a range of advance ratios with CCBlade.jl. Propellers in the optimization were compared at a specific advance ratio, $J = 0.472$, which is listed and explained later in this report in the bottom entry of table II.B. I found the propeller's total torque and total moment from the torque output of the propeller's CCBlade.jl evaluation and from integrating the normal and tangential stresses.

B. Optimization

Once I found the propeller's initial torque and moment at this advance ratio, I multiplied them by a safety factor of $n = 1.1$. This safety factor provided maximum limits for the propeller torque and moment to ensure that the SNOW.jl optimization would find the most efficient propeller that still had reasonable input torque and power requirements to be run by the same motor and made of the same materials as the initial propeller.

In addition to restrictions on the moment and torque mentioned previously, the cord thickness was kept within a factor of two of the original and the pitching angle was kept between $-90°$ and $90°$. These constraints are listed in table 1 after the torque and moment constraints. They ensured that the optimization found a reasonable solution confined to realistic limits.

In this optimization, I modified the chord length and pitching angle to find the most efficient propeller. The optimization is shown in equation 1.

$$\begin{aligned} \text{maximize :} \quad & \eta \\ \text{with respect to :} \quad & c & \theta \\ \text{subject to :} \quad & 50\% \le c \le 200\% & -90° \le \theta \le 90° \\ & M_n \le 1.1 \times M_{n0} & M_t \le 1.1 \times M_{t0} \\ & T \le 1.1 \times T_0 \end{aligned} \tag{1}$$

Table II.B shows shared features of each of the propellers optimized. The advance ratio, J, is available from other variables already listed in the table by equation 2, in which v_∞ is the free stream velocity, n is the rotational velocity in revolutions per second, and D is the outer diameter.

$$J = \frac{v_\infty}{nD}$$
$$J = \frac{12}{(\frac{6000}{60})0.254} \tag{2}$$

Input value limits in propeller design

parameter	default value	minimum value	maximum value
chord length,	100% length	50%	200%
pitching angle,	0°	−90°	90°
rotational velocity	6000 RPM		
blade count	2 blades	1 to 3)	
hub-to-tip ratio	10%		
air density	1.225 kg/m^3		
diameter	0.254 m (10 in.)		
velocity	12 m/s (26.84 mph)		
advance ratio	0.472		

I optimized the objective function, shown in equation 1, using SNOW.jl. SNOW.jl, developed by the FLOW Lab, finds either the maximum or minimum of a function. The code used to design the propeller was prepared so that SNOW.jl found the maximum by finding the minimum of its negative. This is similar to the technique and sign convention described by Martins and Ning in *Engineering Design Optimization*[2]. SNOW.jl can also work with constraints to limit the domain of possible solutions that can be optimized, using fixed environmental variables and limits for design variables. This program used both of these types of inputs to obtain the desired solutions.

III. Results and Discussion

Although the chord length changed during the optimization for all blade counts, the propeller's profile remained the same throughout the optimization. Changing the pitching angle only rotated the propeller blade, and changing the chord magnified the size of its entire profile uniformly along its entire length. The rotor and airfoil themselves retained the same identification numbers throughout the optimization.

Figure 2 shows that while the optimized design increased the efficiency of even the propeller with the highest blade count above the propeller pre-optimization, it dramatically decreased the thrust and torque coefficients. The efficiency at higher advance ratios also decreased. The range of advance ratios over which the propeller maintained high efficiency was smaller. This means that maximizing the propeller efficiency at a single advance ratio by only changing the chord length and twist angle may require other propeller performance measures to be sacrificed.

The plots in figure (2) show that the optimized efficiency was higher than the original efficiency at every blade count. This was the desired result, as stated by the optimization function in equation 1. Figure (2) shows, though, that I did not obtain this increased efficiency without sacrificing other performance measures. While the efficiency is higher, the thrust coefficient and torque coefficient are both much lower. These results are presented below in figure (2).

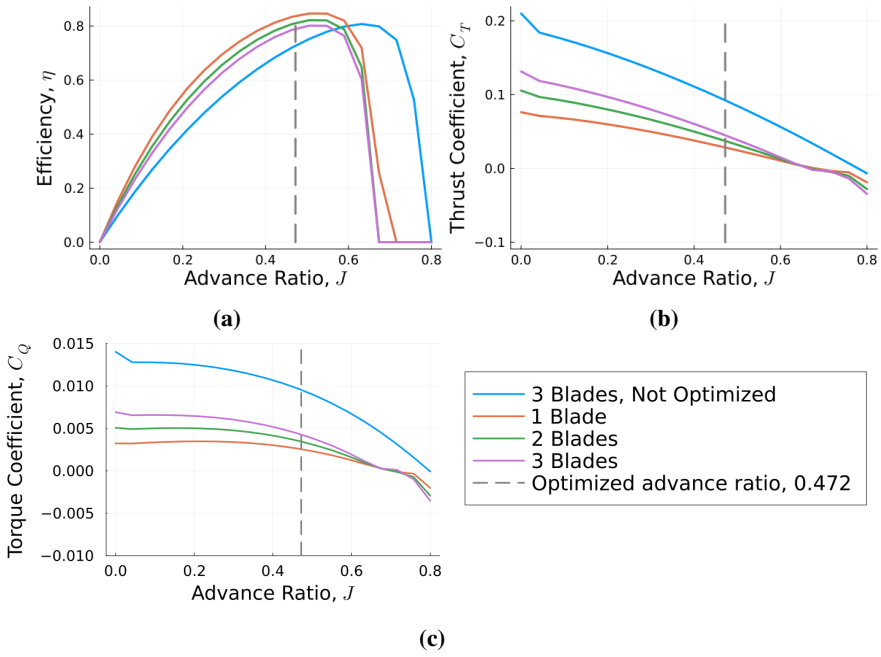


Fig. 2 Efficiency (a), thrust coefficient (b), and torque coefficient (c) compared at different advance ratios

If necessary, I could provide minimum thrust and torque coefficients as parameters along with the maximums shown below the objective function in table 1. I could use these additional constraints to find the pitching angle and chord magnification that would provide the maximum efficiency while still maintaining some required thrust. The optimizer might not reduce the propeller to its minimum possible chord length in every case as it did in this optimization.

Table III shows the optimal pitching angles and chord magnifications found by SNOW.jl for each different propeller blade count. At the conditions listed in table II.B, the optimal chord length magnification was constant, and the optimal twist angle was less than -3° for each propeller. These results prove a few things about the propeller designed.

Optimized chord magnification and pitching angles for different blade counts

Blade count	chord length Multiplication	Pitching angle
3 (Default)	1.0	0°
2	0.50	−2.70°
3	0.50	−2.88°
4	0.50	−2.94°

First off, propellors with lower solidity appear to be more efficient. All of the propellors optimized had the lowest possible chord length allowed by the program used. Rotor solidity can be calculated with equation 3. This shows that reducing the chord length, c, or the blade count, n_b, reduces the solidity, σ. The finding that lower solidity leads to greater efficiency seems reasonable, because wind turbine propellers, designed to efficiently harvest wind from the environment, have very low solidity according to this equation. This finding about propeller blades with less solidity being more efficient also agrees with Saraf, Nouli, Ravalet, and Bakir's findings.[3]

$$\sigma = \frac{c}{s} = \frac{cn_b}{2\pi\sqrt{\frac{r_h^2+r_t^2}{2}}} \tag{3}$$

Table III shows that although the optimal pitching angles were very similar for all three different blade counts, they became minutely more negative as the blade count increased. This further decrease in the already negative pitching angle could be at least partially explained by the airfoil polars shown in figure 1. Increasing the pitching angle slightly above $-3°$ would make its lift coefficient higher and its drag coefficient slightly lower. Figure 1, from airfoil analysis conducted earlier in the semester, shows that angles of attack near $-3°$ for the NACA 4412 airfoil used in this design correspond to a very low drag coefficient and lower drag forces.

Changing the pitching angle of this propeller would change the forces and efficiency experienced by a propeller with a blade count of 2 differently than propellers with blade counts of 3 or 4, because the wake created by each propellor blade affects airflow over the blade next to it. This is why the curves in figure 2 are shaped differently from each other, even though they are optimizations of propellors with the same airfoil and rotor profiles.

One problem with this optimization is that a propeller's efficiency is available as a function of the advance ratio, the thrust coefficient, and the power coefficient. The equation for efficiency is described by Andrew Ning in *Computational Aerodynamics*,[4] using the relations below, in equation 4.

$$\begin{aligned} \eta &= J\frac{C_T}{C_P} \\ \eta &= \frac{TV_\infty}{TV_\infty + \frac{1}{2}\dot{m}(V_w - V_\infty)^2} \end{aligned} \tag{4}$$

Ning also wrote that when thrust is increased for a propellor, it comes at the expense of decreased efficiency.[4] Conversely, a propellor optimization that maximizes efficiency will decrease its thrust coefficient. The torque and moments output by the optimized propellors did not even come near those of the original propellor. For example, the original propellor blades experienced a normal moment of $3.753 \times 10^6 N \times m$, but the optimized propellor with the same blade count only experienced $3.808 \times 10^5 N \times m$.

When I kept J kept constant in this research, inspection of figure 2 reveals that the optimizer maximized equation 1 by minimizing C_P, which is equal to C_Q multiplied by 2π. The torque coefficient C_Q is significantly lower for each propeller. While the thrust coefficient C_T did not decrease by as much as C_Q, it is still much lower than before. Although more efficient, the newly optimized propeller produces much less power and may be better suited for an entirely different environment than the one it started with.

IV. Conclusion

In this optimization, I used Julia to find the

This optimization combined several different Julia packages to find the optimal angle of rotation and chord length for a propeller's efficiency. It found that propellers with very small chord lengths at negative angles of rotation were the most efficient. Other optimizations could be performed by simple editing of the code used in this research to find which propellers perform best in other applications or environments.

V. Acknowledgements

The author thanks Adam Cardoza for mentoring him in learning the codes used in the FLOW Lab, directing his research, helping him through questions and challenges that he faced during his work.

References

[1] Ning, A., "Using Blade Element Momentum Methods with Gradient-Based Design Optimization," *Structural and Multidisciplinary Optimization*, Vol. 64, No. 2, 2021, pp. 994–1014. https://doi.org/http://dx.doi.org/10.1002/we.1636.

[2] J Martins, A. N., *Engineering Design Optimization*, Cambridge University Press, 2021.

[3] Bakir, C. S. H. N. F. R. F., "Experimental study of blade thickness effects on the overall and local performances of a Controlled Vortex Designed axial-flow fan," *Experimental Thermal and Fluid Science*, Vol. 35, No. 4, 2011, pp. 684–693. https://doi.org/https://doi.org/10.1016/j.expthermflusci.2011.01.002.

[4] Ning, A., *Computational Aerodynamics*, Brigham Young University, 2022.