

Propellor Optimization at Different Blade Counts

Joseph Spencer *

Brigham Young University, Provo, Utah, 84601

Propellors 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 propellor performance. This investigation used computer simulations to find the optimal chord length magnification and the optimal rotation angle for a rotor's efficiency. This report defines the operating conditions used in this optimization, presents rotors that were optimized at several blade counts, and discusses areas of possible future development in this research. It found that for the operating conditions used, rotors with the chord reduced to the optimization's lower limit, 50 percent, with slight negative pitching angles were the most efficient. These results could be applied to rotors 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 rotor that can also provide a required amount of thrust or operate at a variety of different rotational velocities.

Nomenclature

J	advance ratio
α	angle of attack (radians)
ϕ	angle of rotation (radians)
M_b	bending moment (N·m)
c	chord length (m)
D	diameter (m)
C_d	drag coefficient
η	efficiency
ρ	fluid density (kg/m ³)
v_∞	freestream velocity (m/s)
C_l	lift coefficient
θ	pitc angle

*Undergraduate Researcher, Brigham Young University FLOW Lab.

C_P	power coefficient
RPM	revolutions per minute
σ	rotor solidity
C_T	thrust coefficient
C_Q	torque coefficient

I. Introduction

PROPELLORS come in a variety of different shapes and sizes. This paper describes how one propellor, an APC 10x7 rotor* with a NACA 4412 airfoil†, was used as a baseline to optimize a propellor’s efficiency at a specific advance ratio. Because different blade counts could be more appropriate for different applications, blade counts of one, two, and three were each optimized and are compared in this report. This report also compares the optimized rotors 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, the properties of hundreds of slightly different propellers are instead compared computationally 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 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.‡

This paper first reviews the procedure used in the rotor optimization, including the objective function used in the optimization and the initial conditions and constraints applied. Next, it reviews the results and compares them with each other. Finally, it draws 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. This project was performed using Julia programming language.¶ Julia 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

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

†Propellor 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 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>.

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 angle of rotation, the angle between the free stream velocity and the tangential velocity of the rotor. After these three quantities are found, they are combined to find forces acting in all directions on the rotor[1].

The rotor in this design was created beforehand and included in the documentation for CCBlade.^{¶¶} Rotor data could be evaluated using XFOIL.jl, but this data was selected because it 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.

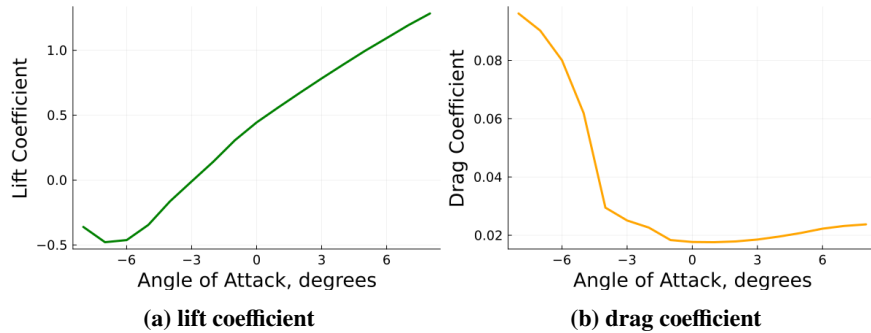


Fig. 1 Lift and drag experienced by NACA 4412 airfoils

These plots show how lift and drag change for an airfoil at different angles of attack.

This airfoil was then placed on the APC 10x7 rotor used in the project, and its torque and power coefficients were evaluated across a range of advance ratios with CCBlade.jl. Rotors 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.A. The

[¶]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/SNOW.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/>

rotor's total torque and total moment were found from the torque output of the rotor's CCBlade.jl evaluation and from integrating the normal and tangential stresses.

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

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, the rotor's chord length and pitching angle were modified to find the most efficient rotor. It is shown in equation (1).

$$\begin{aligned}
 &\text{maximize : } \eta \\
 &\text{with respect to : } c, \theta \\
 &\text{subject to : } T \leq 1.1T_0; M_n \leq 1.1M_{n0}; M_t \leq 1.1M_{t0}; -90^\circ \leq \theta \leq 90^\circ; 50\% \leq c \leq 200\%
 \end{aligned} \tag{1}$$

Table (II.A) shows shared features of each of the rotors 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.

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

Table II: Input value limits in rotor design

parameter	default value	minimum value	maximum value
chord length,	100% length	50%	200%
pitching angle,	0°	−90°	90°
rotational velocity, <i>RPM</i>	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		

The objective function, shown in equation (1), was optimized using SNOW.jl. SNOW.jl, developed by the FLOW Lab, finds either the maximum or minimum of a function and can intake constraint values. The code used to design the rotor was set up 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*[3]. 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

Although the chord length changed during the optimization for all blade counts, the rotor's profile stayed the same throughout the optimization. It Changing the pitching angle only rotated the rotor 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 shape and identification numbers throughout the optimization.

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

increased efficiency was not obtained in exactly the way expected. While the efficiency is higher, the thrust coefficient and torque coefficient are both much lower,

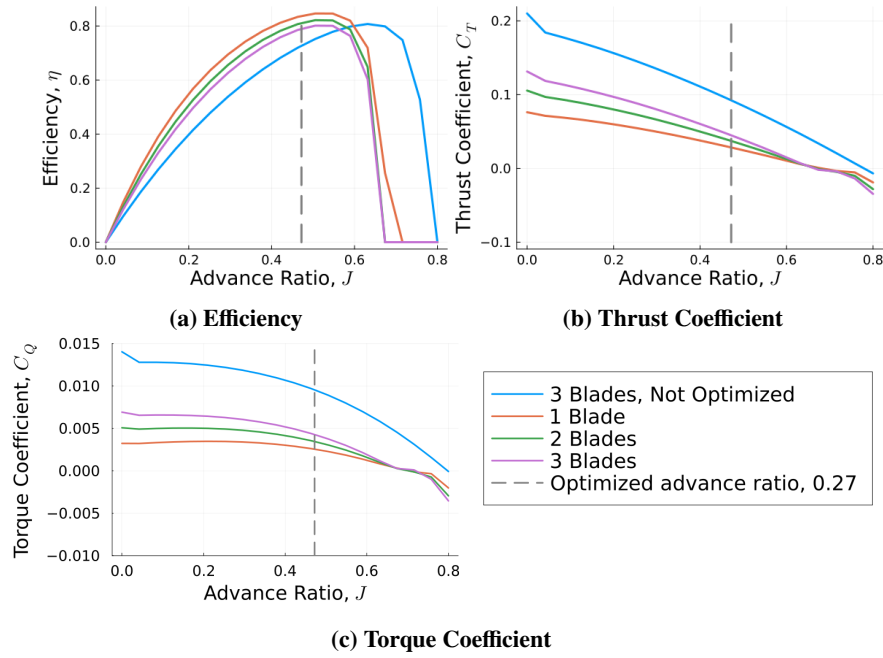


Fig. 2 Efficiency, Thrust Coefficients, and Torque Coefficients Compared at Different Advance Ratios

If necessary, minimum thrust and torque coefficients could be provided as parameters along with the maximums shown below the objective function in table (??). These could find the angle of rotation and chord length that would provide the maximum efficiency while still maintaining some required thrust. The optimizer might not reduce the rotor to its minimum possible chord length in every case as it did in this optimization.

A. Results Table

Table (III.B) shows the optimal pitching angles and chord magnifications for each different rotor blade count. At the conditions listed in table (II.A), the optimal chord length magnification was constant, and the optimal angle of rotation was less than -3° for each rotor. It gradually became more negative as the blade count increased.

Table III (Table IV.B): Optimized chord magnification and rotation 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°

IV. Discussion

The propellor was checked at blade counts of 1, 2, and 3. These results found that for all three blade counts, the optimal propellor was as thin as possible, with a small negative angles of attack. The finding about thinner rotor blades being more efficient agreed with Saraf, Nouli, Ravalet, and Bakir's findings, but the negative rotation angle was surprising at first [4].

One lesson learned from this optimization is that the user should be careful what is optimize for, because the computer will optimize exactly what it is told, even if that is not what the user actually wants. An optimization that is written incorrectly or has unseen loopholes can not only give a misleading answer but also waste a lot of time. In the case of this optimization, I think other constraints could have been added to the optimization

A. Efficiency Equation

One problem with this optimization is that a rotor'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*, page 200 [2] using the relation below, equation (3).

$$\eta = J \frac{C_T}{C_P} \quad (3)$$

With J kept constant, as it was in this research, this equation can be maximized in either by maximizing C_T or by minimizing C_P , which is equal to C_Q multiplied by 2π . Inspection of figure (2) reveals that the optimizer did the latter. The torque coefficient C_Q is significantly lower for each rotor. 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 propellor has a much lower solidity and is better suited for a different environment.

B. Angle of Rotation

The optimal angles of rotation were between -2.70° and -2.94° , as shown in table (III.B). These negative angles of rotation were surprising at first, because angles near 0° were expected to be the most efficient. Figure (3), from airfoil analysis conducted earlier in the semester, shows that 0° angles of attack do not necessarily correspond to the lowest lift and drag coefficients for the NACA 4412 airfoil used in this design. Although that finding was for more simple analysis of an airfoil, it was concluded that a negative angle of rotation was possible.

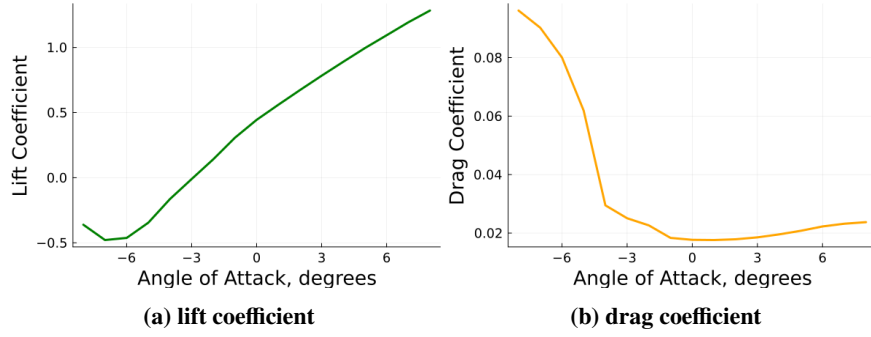


Fig. 3 Lift and drag experienced by NACA 4412 airfoils

These plots show how lift and drag change for an airfoil at different angles of attack.

V. Conclusion

This optimization combined several different Julia packages to find the optimal angle of rotation and rotor chord length for a rotor's efficiency. It found that rotors 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 rotors perform best in other applications or environments.

VI. Acknowledgements

The author would like to thank 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 discovered 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] Ning, A., *Computational Aerodynamics*, Brigham Young University, 2022.
- [3] J Martins, A. N., *Engineering Design Optimization*, Cambridge University Press, 2021.
- [4] 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>.