Propeller Optimization at Different Blade Counts ~~of Multiple~~ Rotors

Joseph Spencer[*]

Brigham Young University, Provo, Utah, 84601

Propellers can be modified to perform better under a variety of conditions. ~~This report describes a~~ One project completed as part of my research for the Brigham Young University FLOW Lab explores ~~to discover~~ ideal design variables ~~conditions~~ for propeller ~~airfoil~~ performance. This investigation used computer simulations ~~of propellers allow researchers to perform low-risk studies and develop designs more quickly~~ to find the optimal chord length magnification and the optimal pitching angle for a propeller's efficiency. This ~~paper~~ report defines the operating conditions used in this optimization, presents ~~rotors~~ propellers that were optimized at several blade counts, and discusses areas of possible future development in this research. I ~~This paper~~ found that for the operating conditions used, ~~rotors~~ propellers with the chord reduced to the optimization's lower limit, 50 percent, with ~~slight~~ negative pitching angles were the most efficient but gave up much of their power. When I added a minimum limit to the rotor's power, a rotor with a positive twist angle but still a very reduced chord length was optimal. These results could be applied to ~~rotors~~ propellers designed to produce the maximum output power for a given input power requirement, without regard to other ~~conditions~~ desired outputs or with consideration for only the power output ~~conditions like the thrust and torque coefficients~~. Future work could combine these findings with other conditions to find the most efficient propeller that can also provide a required amount of thrust or operate at a variety of different rotational velocities.

~~Nomenclature~~

~~(Nomenclature entries are identified with their default units.)~~

~~J~~ = Advance Ratio, *dimensionless*

~~α~~ = Angle of Attack, *rad*

~~ϕ~~ = Angle of Rotation, *rad*

~~M_b~~ = Bending Moment, $N \times m$

~~c~~ = Chord length, *m*

---

[*]Undergraduate Researcher, Brigham Young University FLOW Lab.

D = Diameter, m

c_d = Drag Coefficient, *dimensionless*

η = Efficiency, *unitless*

ρ = Fluid Density, kg/m^3

v_∞ = Freestream Velocity, m/s

c_l = Lift Coefficient, *dimensionless*

C_P = Power Coefficient, *dimensionless*

RPM = Revolutions Per Minute, $\frac{rev.}{min.}$

σ = Rotor Solidity *dimensionless*

C_T = Thrust Coefficient, *dimensionless*

C_Q = Torque Coefficient, *unitless*

I. Introduction

PROPELLERS 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 was optimized to perform more efficiently at a specific advance ratio. Because different blade counts could be more appropriate for different applications[‡], I optimize and compare blade counts of 2, 3, and 4 were each optimized and compared in this report. I then compares the optimized propellers of each different blade count. This report produces useful code that could be applied to different rotors in the future.

A. Context

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 manufacturing propellers in which hundreds of slightly different propellers are compared. In the potential-flow code (PFC) simulation used in this report, I instead computationally compared the properties of in which hundreds of slightly different propellers are compared to obtain a final result. A PFC is a program for modeling the pressure distribution of a rotor. PFCs were first programmed The first potential flow codes, programs for modeling the pressure distribution of rotors, were developed in the 1960s and 1970s. Xfoil, the 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, the Julia wrapper for Xfoil, was developed by Taylor McDonnell[§].

---

*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`

[‡]Differently sized propellers can produce different efficiencies and noise levels, among other things. `https://hartzellprop.com/are-more-propeller-blades-better/`

[§]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`

B. ~~Paper Outline~~

Inthis paper I first explain ~~reviews~~ the procedure used in the ~~rotor~~ propeller optimization, including the objective function ~~equation~~ used in the optimization and the initial conditions and constraints applied. Next, ~~it reviews~~ I investigate the results and compare~~s~~ them with each other. I examine what is happened when I tweak the input constraints. ~~Finally,~~ I then draw conclusions based on the results of this ~~experiment~~ optimization and discusses their meaning. After doing this, I add one more constraint to the optimization function and examine the different output of the function compared to the original optimization. All code used for each section of this report can also be accessed through a GitHub repository[¶].

II. Methods Procedure

Computational stimulations of propellers allow researchers to perform low-risk studies and develop designs more quickly. I ~~This project was~~ 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 ~~at runtime~~, which makes it run faster than interpreted languages like ~~p~~Python without needing to be totally compiled before running after any change is made like C**. ~~Being compiled just-in-time makes Julia faster than interpreted languages like python.~~ Julia packages, including those used in ~~during~~ this ~~semester~~ optimization and in previous projects, are also readily ~~easily~~ available from a package manager.

A. Models ~~Computer Code~~

Some Julia packages used for this project include Xfoil.jl[††], CCBlade,jl[‡‡], QuadGK.jl[§§], and SNOW.jl[¶¶]. Xfoil.jl, discussed previously, 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. It divides the airfoil into many small surfaces and finds the lift, drag, and moment forces at each of the smaller surfaces. ~~This investigation used SNOW.jl to repeat itself until it converged to an optimal solution.~~

The airfoil in this design was created beforehand and included in the documentation for CCBlade.jl***. Airfoil data could be created using XFoil.jl, but I selected this provided data because its domain had already been widened using Viterna extrapolation to include a complete range of angles of attack[†††]. Airfoil polars of this airfoil made using code

---

[¶]This repository can be accessed through `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`

[§§]QuadGK.jl can be found at `https://github.com/JuliaMath/QuadGK.jl`

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

***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/`

from my first project are presented in figure (1).

In the rotor design, the rotor is created and evaluated using Xfoil.jl and CCBlade.jl, as described previously. Data about the rotor, including its loads in the normal and tangential directions and its torque, were recorded. This data was then multiplied by a safety factor of n = 1.1 to determine the maximum allowable loads before the rotor would break or a different material would be required. n = 1.1 was the safety factor defined in the project prompt, but it was included in the code as an optional argument that could be adjusted for other circumstances. After constraints were determined, rotor variable types called rotortest were created with variable chord thickness magnification and rotation angle and the program used SNOW.jl to find the rotor with the optimal twist angle and chord thickness magnification for the objective function.

The objective function used in this optimization is very simple, as shown in equation (1).

$$f(x_1, x_2, x_3...) = \eta \tag{1}$$

In addition to restrictions on the moment ant torque mentioned previously, the chord thickness was kept within a factor of two of the original and the twist angle was kept between $-90°$ and $90°$. These constraints are listed in table (II.A). They ensured that the optimization found a reasonable solution confined to real limits, shown in table (II.A).

The optimized rotors each had other features in common between them, shown in the last entry of table (II.A). Ning (page 200) stated that the advaance ratio 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.

$$J = \frac{v_\infty}{nD} \tag{2}$$

4

Table I: Optimization objective, parameters, and constraints

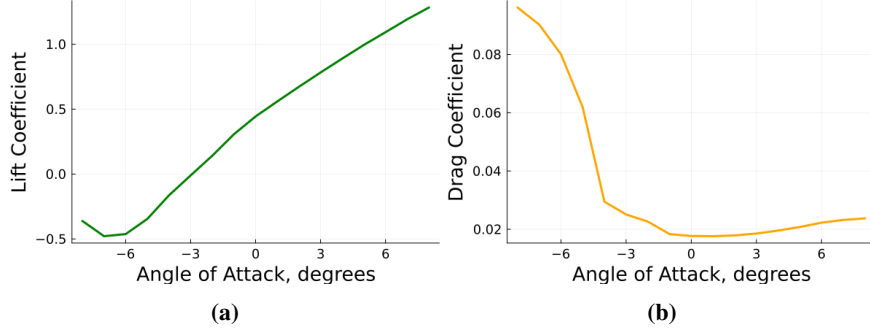|	
Maximize:	η
By varying:	scale of the chord, c
	twist angle, ϕ
Subject to	total torque $\leq 110\%$ original
	normal moment $\leq 110\%$ original
	tangential moment $\leq 110\%$ original
	$-90° <$ twist angle $< 90°$
	$50\% <$ chord magnification $< 200\%$

Fig. 1   (a) Lift and (b) drag coefficients of NACA 4412 airfoils

After the airfoil was created, I lofted it over the APC 10x7 rotor used in the project, and evaluated the propeller's 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 explained later in this report and listed in the bottom entry of table (1).

CCBlade.jl uses blade element momentum (BEM) theory[‡‡‡] to calculate a propeller or turbine's behavior. BEM theory, which combines blade element theory and momentum theory, calculates the axial and tangential induction factors and the pitching 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].

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 force data output by CCBlade.jl across the rotor blade using an integration function from the QuadGK.jl package. With the rotor blade's dimensions known, the moments and torque found by integrating with QuadGK.jl provided reasonable limits for use in the optimization.

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

B. Optimization

I preformed two different optimizations in my propeller design. One optimization to only maximized its efficiency, and another maximized efficiency while maintaining torque within 90% of the original. These two optimizations converged to different propeller designs.

Propeller blade thickness in both designs was not allowed to increase or decrease by a factor of more than 2, and the propeller could not pitch more than $90°$ in either direction. The propeller's torque and normal and tangental moments did not exceed 110% of the values calculated from the CCBlade.jl output when the original propeller was created. Equation (3) show how the advance ratio was calculated. v_∞ represents the free stream velocity, n is rotational velocity in revolutions per second, and D is the outer diameter.

$$
\begin{aligned}
J &= \frac{v_\infty}{nD} \\
&= \frac{12}{\left(\frac{6000}{60}\right) \times 0.254}
\end{aligned}
\tag{3}
$$

Table (1) shows shared features and design variables of the propellers optimized. The advance ratio, J, is available from other variables already listed in the table by equation (3). These values provided all the environmental and design variables I used.

parameter	default value	minimum value	maximum value
chord length,	100% length	50%	200%
pitching angle,	$0°$	$-90°$	$90°$
rotational velocity	6000 RPM		
blade count	3 blades	2, 3, or 4	
hub-to-tip ratio	10%		
air density	1.225 kg/m^3		
outer diameter	0.254 m (10 in.)		
velocity	12 m/s (26.84 mph)		
advance ratio	0.472		

Table 1   Initial values and limits in propeller design

1. *No Minimum Torque*

In both optimizations, I modified the chord length and pitching angle for propellers at the same advance ratio to

find the most efficient one. The objective function, design variables, and constraints of the optimization without a minimum torque are displayed in equation (4), with the advance ratio $J = 0.475$ calculated from equation (3).

$$\text{maximize : } \eta \text{ at } J = 0.475$$

$$\text{with respect to : } c \qquad\qquad \theta$$

$$\text{subject to : } 50\% \leq c \leq 200\% \quad -90° \leq \theta \leq 90° \tag{4}$$

$$M_n \leq 1.1 M_{n0} \qquad\qquad M_t \leq 1.1 M_{t0}$$

$$T \leq 1.1 T_0$$

~~The objective function, equation~~ (**??**) ~~was optimized 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 ~~designed~~ prepared so that SNOW.jl found the maximum of the objective function by finding the minimum of its negative. This is similar to the technique and sign convention described ~~used~~ by Martins and Ning in ~~page 10 of~~ *Engineering Design Optimization*[2]. SNOW.jl can also use fixed environmental variables and set limits or constraints on design variables to contain the domain of possible solutions that can be optimized. The program written for this project used both of these inputs to obtain the desired solutions, so SNOW.jl was a very useful tool.

One discovery gained from these findings is that blades with smaller chord length make more efficient propellers. This finding is related to the rotor solidity, described in equation (5). The optimization using SNOW.jl found the minimum chord length as the most efficient each time. Another evidence that increasing solidity decreases efficiency can be observed in figure (4) in the conclusion. The rotor solidity increased proportionally to the blade count. As it increased, the efficiency at each advance ratio decreased.

$$\sigma = \frac{c}{s}$$

$$= \frac{c n_b}{2\pi \sqrt{\frac{r_h^2 + r_t^2}{2}}} \tag{5}$$

All of the propellers optimized had the lowest chord length allowed by the program used. Equation (5) 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 turbines with long, thin blades, 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].

With this information, I decided to find what the optimal chord length actually was. Figure (2) shows what happened when I removed the original 50% lower limit of the optimizer and instead set it to 1% in the function argument. SNOW.jl found a propeller that was much more efficient, and had exactly 1% of the original chord length.
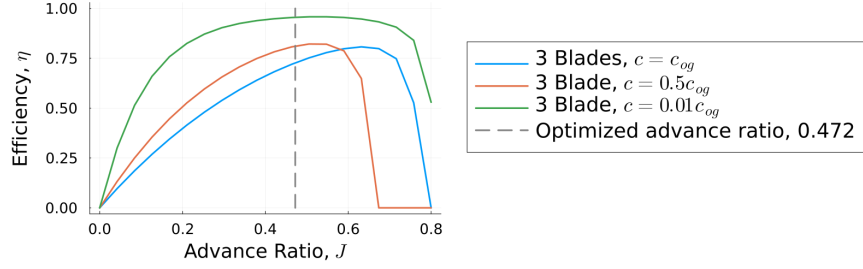
Fig. 2   Comparison of propellers with different cord lengths

A propellor with a chord length that much smaller is unrealistic to actually use, though, because it would produce hardly any thrust compared to the original propeller. One APC 10x7 propeller found on the internet had a 1.02 cm hub thickness to start with[§§§], and lowering the chord to 1% would decrease it to 102µm. It appears that even in extreme cases, decreasing chord length always increases propeller efficiency even when it is most likely no longer desired.

One significant problem with the first optimization is that a propeller's efficiency can be obtained as a function of its advance ratio, thrust coefficient, and power coefficient. The equation for efficiency is described by Andrew Ning in *Computational Aerodynamics*[4], using the relations in equation (6).

$$
\begin{aligned}
\eta &= J\frac{C_T}{C_P} \\
&= \frac{TV_\infty}{TV_\infty + \frac{1}{2}\dot{m}(V_w - V_\infty)^2}
\end{aligned}
\tag{6}
$$

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

When I kept J constant in this research, inspection of figure (4) in the results section reveals that the optimizer maximized equation (4) 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 outputs much less power and may be better suited for an entirely different function than the one it started with.

Another discovery made in the optimization without a minimum torque requirement was that this propeller was most efficient with negative blade pitching angle. To investigate this result further, I ran another simulation with the Julia code I had written to see what happened to this blade pitching angle when the propeller was increase back to its original chord length. The plot in figure (3) shows the original propeller compared to the optimized propeller, shown

---

[§§§]This propeller was found at `https://www.apcprop.com/product/10x7e/`.

in figure (4) and discussed in the results section, and a propeller with the optimized pitching angle but the same chord magnification as the original propeller.
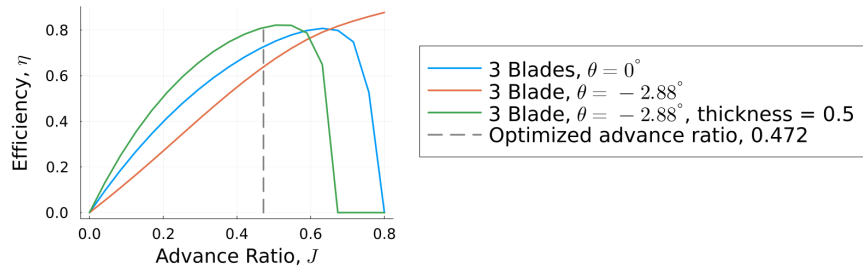

Fig. 3   Comparison of identical propellers at different pitching angles to optimized propeller

The different curves plotted in figure (3) show that the propeller only became more efficient at the $-2.88°$ pitching angle when its chord length was also reduced significantly. Otherwise, it became less efficient at the advance ratio measured but more efficient at higher advance ratios. This showed that rotors with different chord length are better suited for different rotational velocities and free stream velocities, discussed in equation (3). The optimization described previously increased the efficiency, but only for much thinner propeller blades.

Table (2) in the results section 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.

Changing the pitching angle of this propeller would change the forces and efficiency produced by a propeller with a blade count of 2 differently than propellers with blade counts of 3 or 4, because the wake instability created by each propeller blade affects the other blades[5]. This change in blade-to-blade interactions is why the curves in figure (4) are shaped differently from each other, even though they are optimizations of propellers with the same airfoil and rotor profiles.

2. Minimum Torque Added

To prevent the optimizer from sacrificing thrust for efficiency, I provided a minimum torque as a parameter along with the maximums shown below the objective function in equation (4). I used this additional constraint to find the pitching angle and chord magnification that would provide the maximum efficiency while still maintaining some required thrust or being powered by the same motor. This way, the optimizer did not reduce the propeller to its minimum possible chord length in every case as it did in this optimization.

I combated the problem of the optimizer maximizing efficiency by minimizing the thrust coefficient as shown in equation (6) by adding a minimum torque requirement to the optimization. The updated requirements matrix with the torque requirement changed from equation (4) is shown below, in equation (7):

$$
\begin{aligned}
\text{maximize}: \quad & \eta \text{ at } J = 0.475 \\
\text{with respect to}: \quad & c \qquad\qquad\qquad \theta \\
\text{subject to}: \quad & 50\% \leq c \leq 200\% \quad -90° \leq \theta \leq 90° \\
& M_n \leq 1.1 M_{n0} \qquad\qquad M_t \leq 1.1 M_{t0} \\
& 0.9 T_0 \leq T \leq 1.1 T_0
\end{aligned}
\tag{7}
$$

With the added information from plot (2) and plot (3), I programmed one final optimization that eliminated the problem of SNOW.jl converging to the propeller with the minimum thickness. I added another constraint to the objective function used for the initial optimization, described by equation (4). I added a minimum constraint to the propeller's torque, $Q \geq 0.9 Q_0$. The torque coefficient C_Q and power coefficient C_P are related by a factor of 2π, so keeping the propeller torque within 90% of the original would also maintain 90% of the original power output, or $P \geq 0.9 P_0$.

III. Results and Discussion

~~The propeller was checked at blade counts of 1, 2, and 3. These results found that for all three blade counds, the optimal propellor was as thin as possible, with a smal negative angle of attack. The finding about thinner rotor blades being more efficient agrees with Saraf, Nouli, Ravalet, and Bakir's findings, but the negative rotation angle was a surprise at first.~~

~~One lesson learned from this optimization is that the user should be careful what is optimized 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.~~

Although the propeller's profile remained the same throughout the first optimization, the chord length decreased for all blade counts. 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. Rotor and airfoil themselves retained the same NACA and APC identification numbers throughout the optimization.

A. ~~Efficiency Equation~~

Figure (4) demonstrates that while the optimized design increased the efficiency of even the propeller with the

higher blade count above the propeller pre-optimization, it dramatically decreased the thrust and torque coefficients. The efficiency at higher advance ratios also went to zero more quickly, so the range of advance ratios over which the propeller maintained high efficiency was smaller. Maximizing the propeller efficiency at a single advance ratio by only changing the chord length and pitching angle may require other propeller performance measures to be sacrificed.
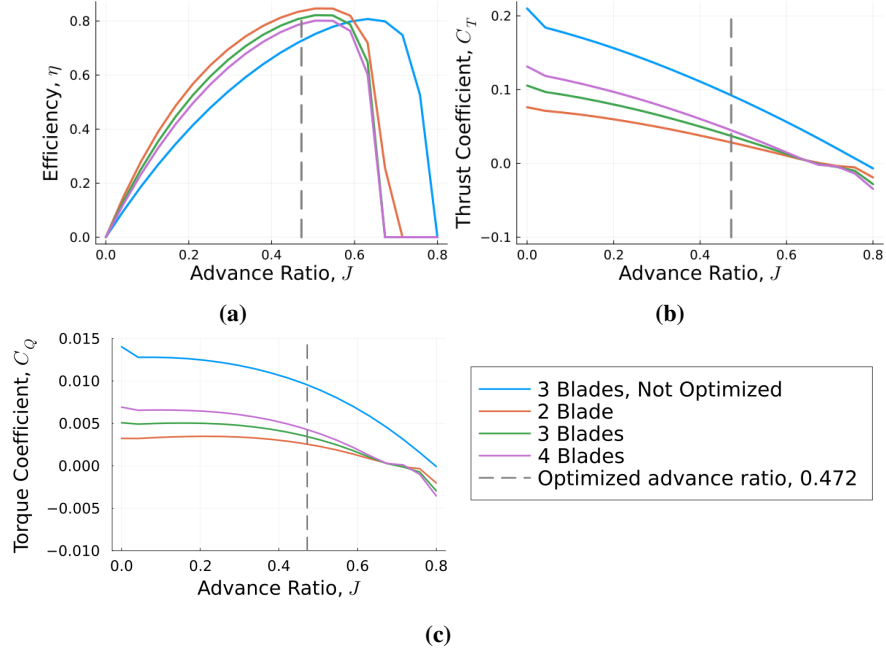

Fig. 4   (a) Efficiency, (b) thrust coefficient, and (c) torque coefficient compared at different advance ratios

Table (2) shows the optimal pitching angles and chord magnifications found by SNOW.jl for each different propeller blade count. At the conditions listed in table (1), the same optimal chord length magnification was found, and the optimal pitching angle was less than $-3°$ for each propeller. These results prove a few things about the propeller designed.

blade count	chord length magnification	pitching angle
3 (Default)	1.0	0°
2	0.500	−2.70°
3	0.500	−2.88°
4	0.500	−2.94°
3 ($P \geq 0.9P_0$)	0.532	2.96°

Table 2   Optimized chord magnification and pitching angles for different blade counts

The plots in figure (5) show two results of the new design, with the minimum thrust requirement. First, the propeller efficiency increases at the measured advance ratio, but not by quite as much as the propeller without the minimum power constraint. Second, it is even more efficient at higher advance ratios. This optimized propeller's final chord length was still decreased to 53.2% of its original chord length, which was near although not quite at the 50% lower limit given to SNOW.jl. Unlike the propellers optimized previously, though, this optimized propeller's pitching angle was positive, 2.96°.
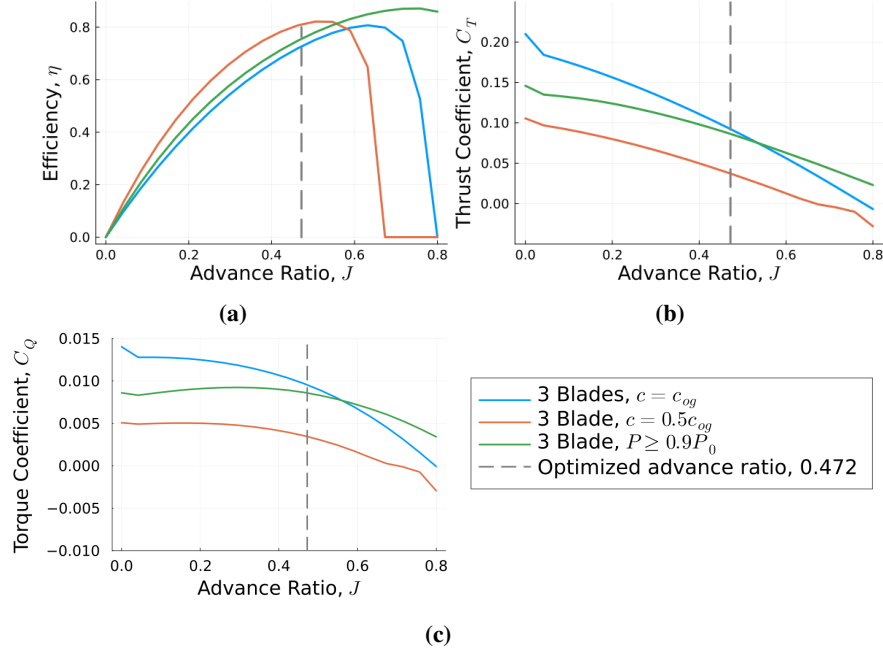

Fig. 5   Efficiency (a), thrust coefficient (b), and torque coefficient (c) compared between different optimizations

One problem with this optimization is that a rotor's efficiency is available as a fuction 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 using the relation below, equation (8)

$$\eta = J \frac{C_T}{C_P} \qquad (8)$$

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 (4) 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~~ $-2.74°$ ~~and~~ $-2.94°$ ~~, as shown in table~~ (**??**). ~~These negative angles of rotation were surprising at firs, because angles near~~ $0°$ ~~were expected to be the most efficient. Figure~~ (6), ~~from airfoil analysis conducted earlier in the semester, shows that~~ $0°$ ~~angle of attack did not necessarily correspond to the lower 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.~~
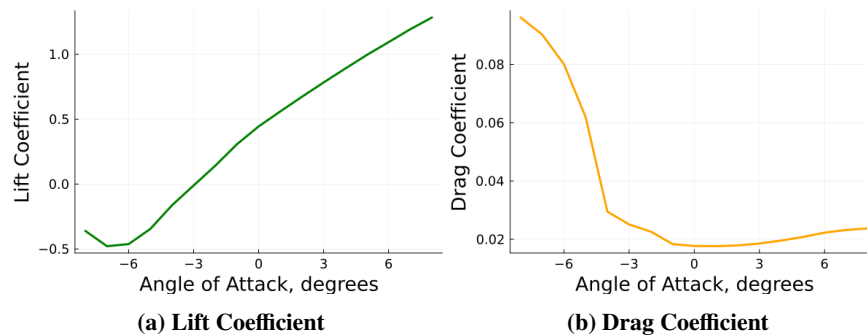

(a) Lift Coefficient   (b) Drag Coefficient

Fig. 6   Lift and Drag Experienced by NACA 4412 Airfoils
These plots show how lift and drag change for an airfoil at different angles of attack.

IV. Conclusion

In this optimization, I used ~~combined~~ several Julia packages to find the optimal combination of chord length and pitching angle ~~of rotation and rotor chord length~~ for ~~a rotor's~~ the maximum efficiency of a propeller at a known advance ratio. I discovered that for the conditions listed in table (1), the optimal propeller has 50% its original chord length, which is the lower limit of the optimization, and a pitching angle between $-2.70°$ and $-2.94°$, depending on the blade count. ~~I found that rotors with very small chord length at negative angles of rotation were the most efficient.~~

After finding this result, I added a constraint to the minimum torque of the propeller, to ensure that it still transferred enough of its power to the air while being efficient. When I did this, I found the optimal pitching angle to be $2.98°$ and the optimal chord length magnification of 0.52. This proved that optimizations with just one additional constraint can produce very different results.

Future work could investigate other possible constraints to place on this optimization, or change the constraints already present. Propellers can operate at many different advance ratios. Other optimizations could vary other aspects of the propeller. For example, the rotor length could be optimized while maintaining a maximum power output, or the shape of the airfoil could be changed entirely or even lofted through different shapes. The rotor could also be modified to an optimal shape. As long as the user understands well the constraints that must be met, optimizations have immense possibilities. ~~Other optimizations could be performed by simple editing of the code used in this research to find which other rotors perform best in other applications or environments.~~

Performing an optimization aided by computers allowed me to explore a broader design space as new ideas were rapidly generated that had not been considered before.

V. Acknowledgements

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] Martins, J. R. R. A., and Ning, A., *Engineering Design Optimization*, Cambridge University Press, 2021.

[3] Sarraf, C., Nouri, H., Ravelet, F., and Bakir, 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.

[5] Felli, M., Guj, G., and Camussi, R., "Effect of the number of blades on propeller wake evolution," *Experiments in fluids*, Vol. 44, No. 3, 2008, pp. 409–418.