

Optimization of Multiple Rotors

J. Spencerr *

Brigham Young University, Provo, Utah, 84601

This report describes a project completed as part of my research for the Brigham Young University FLOW Lab. this project drew on skills gained throughout the semester to design and execute a code in the julia programming language to find the optimal blade count, blade thickness magnification, and twist angle for a propellor to achieve its maximum efficiency, η , at a given advance ratio, J . This investigation found that thinner rotors are more efficient, especially at negative angles of attack. The implications of these results and possible areas of future research are then discussed.

Nomenclature

(Nomenclature entries are identified with their default units.)

J	=	Advance Ratio, <i>dimensionless</i>
α	=	Angle of Attack, <i>rad</i>
ϕ	=	Angle of Rotation, <i>rad</i>
M_b	=	Bending Moment, $N \times m$
c	=	Chord length, <i>m</i>
D	=	Diameter, <i>m</i>
c_d	=	Drag Coefficient, <i>dimensionless</i>
η	=	Efficiency, <i>unitless</i>
ρ	=	Fluid Density, kg/m^3
v_∞	=	Freestream Velocity, <i>m/s</i>
c_l	=	Lift Coefficient, <i>dimensionless</i>
C_P	=	Power Coefficient, <i>dimensionless</i>
RPM	=	Revolutions Per Minute, $\frac{rev.}{min.}$
σ	=	Rotor Solidity <i>dimensionless</i>
C_T	=	Thrust Coefficient, <i>dimensionless</i>
C_Q	=	Torque Coefficient, <i>unitless</i>

*Undergraduate Researcher, Brigham Young University FLOW Lab.

I. Introduction

PROPELLORS come in a variety of different shapes and sizes. This paper describes how one propellor, with an APC 10x7 rotor and a NACA 4412 airfoil, was optimized to perform better at a certain advance ratio. The code used for this report could also be applied to different rotors in the future.

This report shows how computer simulations can be used to simulate the performance of propellers without needing to actually create them. It finds that changing the twist angle and the thickness of a propeller does not just shift the curves for its efficiency, thrust coefficient, and torque coefficient; it entirely changes their shape. Propellers with different blade counts also have different curves for these three non-dimensional numbers. All code used for each section of this report can also be accessed through a GitHub repository*.

II. Procedure

This project was performed using julia programming language.[†] julia is available for free and is useful for a variety of reasons, including that It can store data as vectors and matrices and perform rapid calculations with these objects. It compiles functions in advance, so they can be run more quickly than in some other languages. Function files used for this project, including Xfoil[‡], CCBlade[§], SNOW[¶], and FLOWMath^{||}, are all designed for julia. These packages provided useful functionality that simplified the design process.

In the rotor design, an airfoil was first created using Xfoil.jl. This airfoil was then attached to a rotor and evaluated using CCBlade.jl. Data about the rotor, including its moments in the normal and tangential directions and its torque, is 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. After constraints were determined, rotor variable types called *rotortest* were created with variable thickness magnification and rotation angle and the program used Optim.jl to find the rotor with the optimal twist angle and thickness magnification for the objective function.

The objective function used in this optimization is listed in equation (??).

$$f(x_1, x_2, x_3 \dots) = \eta \quad (1)$$

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 twist angle was kept between -90° and 90° . These constraints are listed in table (??). They ensured that the optimization found a reasonable solution confined to real limits, shown in table (??).

*This repository can be accessed in the Rotor-Design branch of <https://github.com/JoeSpencer1/497R-Projects>

[†]julia can be found at <https://julialang.org>.

[‡]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>

^{||}FLOWMath.jl is available at <https://github.com/byuflowlab/FLOWMath.jl>

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 $-90^\circ < \text{twist angle} < 90^\circ$ $50\% < \text{chord magnification} < 200\%$

The optimized rotors each had other features in common between them, shown in the last entry of table (??). Ning (page 200) stated that the advance ratio can be found from other variables already listed in the table by equation (??), in which v_∞ is the free stream velocity, n is the rotational velocity in revolutions per second, and D is the outer diameter [?].

$$J = \frac{v_\infty}{nD} \quad (2)$$

Table II: Input value limits in rotor design

Parameter	Default Value	Minimum Value	Maximum Value
Chord Length,	100% length	50%	200%
Twist 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/ <i>m</i> ³		
Diameter	0.254 m (10 in.)		
Velocity	12 m/s (26.84 mph)		
Advance Ratio	0.472		

III. Optimization

The objective function, equation (??), was optimized using SNOW.jl. The code was designed 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 page 10 of Engineering Design Optimization [?].

IV. Results

Although the chord thickness changed during the optimization for all blade counts, both the rotor identification and the NACA number stayed the same throughout the optimization. Changing the pitching angle only rotated the rotor blade, and changing the chord length magnified the size of its entire profile uniformly.

A. Results Plot

Figure (??) 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. These decreases in other rotor properties may not be desired. This illustrates one problem with optimizers: if created correctly, they will do exactly what they are programmed to do and may sacrifice some desirable properties to further optimize the objective function.

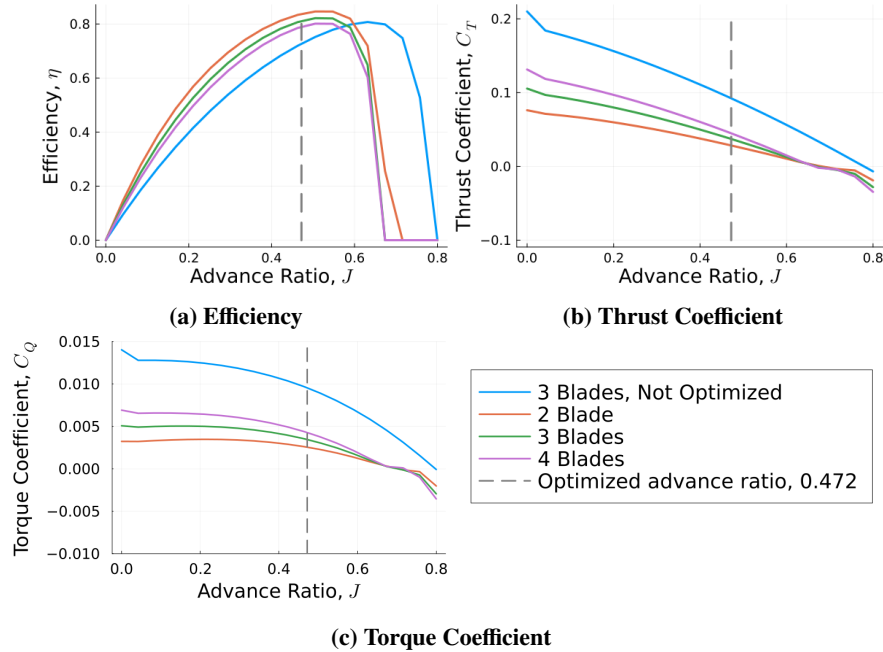


Fig. 1 Efficiency, Thrust Coefficients, and Torque Coefficients Compared at Different Advance Ratios

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 thickness that would provide

the maximum efficiency while still maintaining some required thrust. The optimizer might not reduce the rotor to its minimum possible thickness in every case as it did in this optimization.

B. Results Table

Table (??) shows the optimal twist angles and chord magnifications for each different rotor blade count. At the conditions listed in table (??), 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 Thickness Multiplication	Twist Angle
3 (Default)	1.0	0°
2	0.50	-2.70°
3	0.50	-2.88°
4	0.50	-2.94°

V. 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 [?].

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 can be found 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 [?] using the relation below, equation (??).

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

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 (??) 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 (??). These negative angles of rotation were surprising at first, because angles near 0° were expected to be the most efficient. Figure (??), 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.

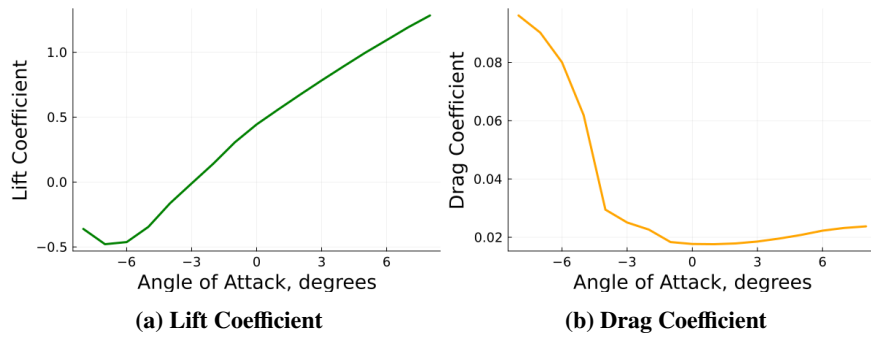


Fig. 2 Lift and Drag Experienced by NACA 4412 Airfoils
These plots show how lift and drag change for an airfoil at different angles of attack.

VI. 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.

VII. 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.