Towards Low-Noise Design of a Proprotor with the Vortex Lattice Method and Gradient-Based Optimization

Daniel Ingraham [1] David Kinney [2]

[1]NASA Glenn Research Center

[2]NASA Ames Research Center

AIAA SciTech Conference
January 8th– 12th, 2024

This work is supported by the Transformational Tools and Technologies project, part of the Transformative Aeronautics Concepts Program.

Motivation: Multidisciplinary design optimization of UAM/AAM concepts

Previous Work[1]: BEMT+Compact F1A+Optimization=Quiet Propeller

Combined

- a propeller aerodynamic code, implementation of blade element momentum theory (BEMT)
  - CCBlade.jl, A. Ning, BYU
- a propeller acoustic code, implementation of the compact form of Farrasat's 1A acoustic analogy
  - AcousticAnalogies.jl, using L. Lopes compact approximation to the monopole term
- gradient-based optimizer

to design an optimally efficient propeller subject to thrust and acoustic constraints.

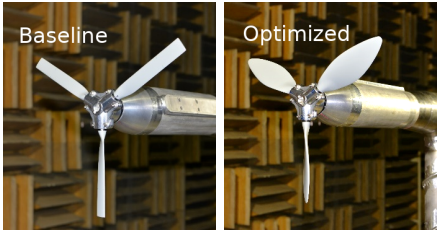


Image credit: N. Zawodny, L. Lopes, NASA LaRC.

[1] See `https://ntrs.nasa.gov/citations/20220015637` for more.

Current Goal: Replace BEMT with VLM

- Blade Element Momentum Theory is great
  - fast
  - robust
  - accurate for "simple" cases (isolated propeller, on axis flow)

  but **limited in applicability** (multiple rotors, installation effects, off-axis flow, etc.)
- Computational Fluid Dynamics (CFD) too slow for highly multi-disciplinary optimizations

Goal

Replace the BEMT aerodynamic model used in previous work with the vortex lattice method (VLM)

Many tools. . .

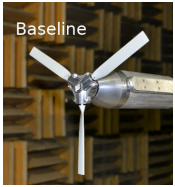
- Geometry parameterization
  - OpenVSP called via pygeo
- Aerodynamics
  - VSPAERO, used in unsteady VLM mode
  - CCBlade.jl
- Acoustics
  - AcousticAnalogies.jl
- Optimization framework
  - NASA's OpenMDAO
  - SNOPT called via pyOptSparse (`https://github.com/mdolab/pyoptsparse`, Univ. of Michigan MDO lab)

(See last slide for attribution and URLs.)

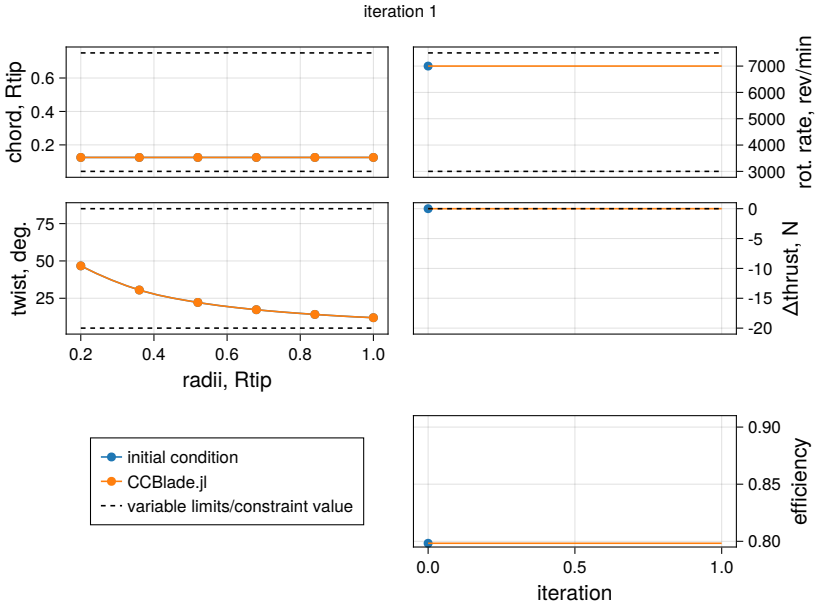
Problem setup: maximize propeller efficiency, **with and without downstream wing**

- Objective: maximize efficiency at cruise
- Design variables:
  - Rotation rate
  - Chord distribution via 6 spline control points
  - Twist distribution via 6 spline control points
- Constraints:
  - average thrust equal to baseline design's value
  - Overall sound pressure level 4 dB less than baseline
  - chord and twist curvature constraints (stay tuned)

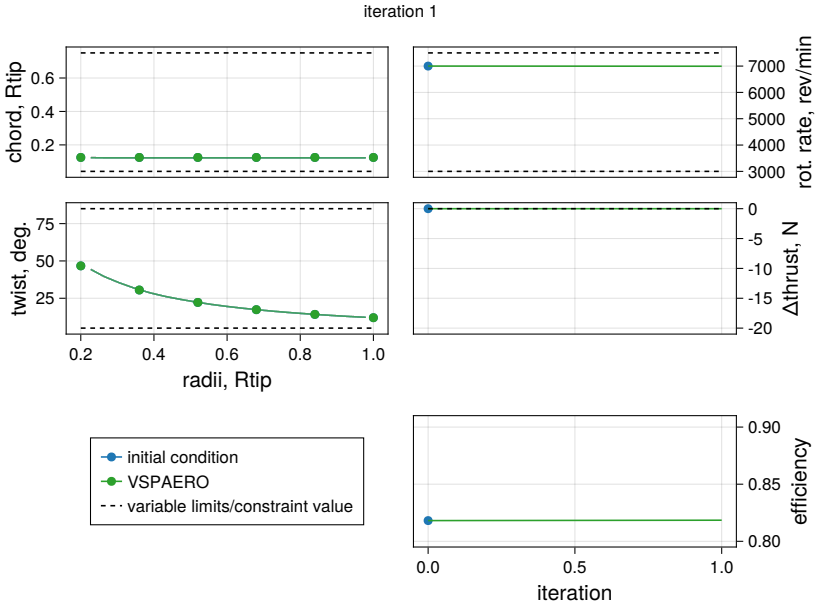
- Baseline design: C24ND from N. Zawodny, L. Lopes, NASA LaRC



CCBlade.jl optimizations for an isolated propeller seem reasonable



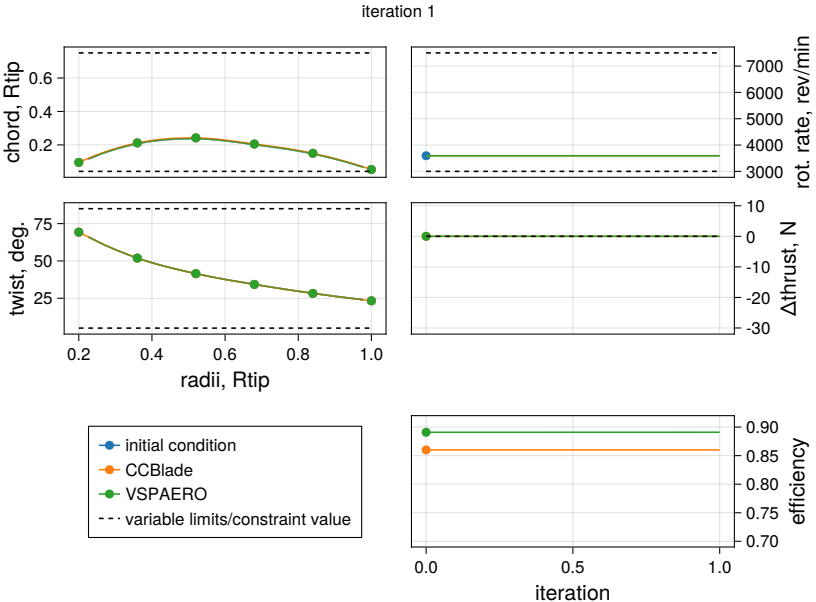
But VSPAERO optimizations for an isolated propeller are a bit wonky



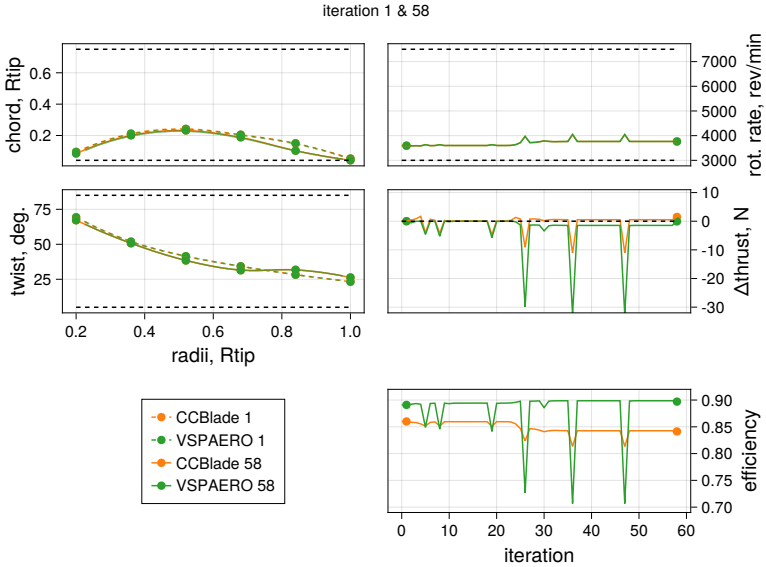
Two Ideas

1. Starting the VSPAERO optimization from a more "optimal" design may coax it into something more reasonable
2. Taking VSPAERO's convergence history and applying it to CCBlade.jl may reveal differences between the two codes

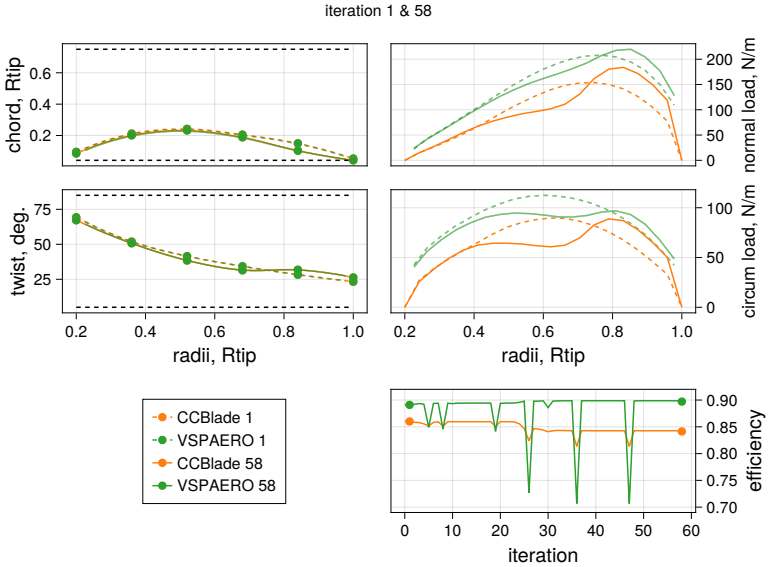
Wonky results persist, even when starting from CCBlade.jl-optimal result



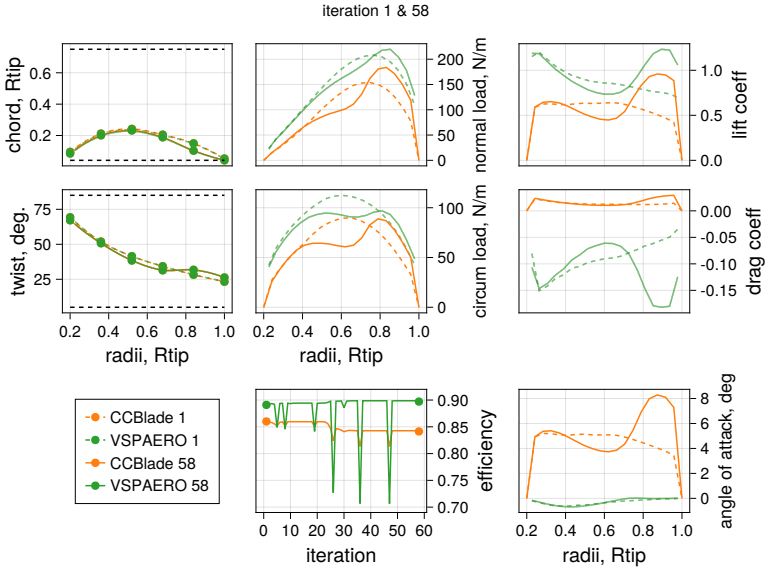
Iteration 1 and 58 show significant differences in efficiency—why?



CCBlade.jl cares about twist more than VSPAERO



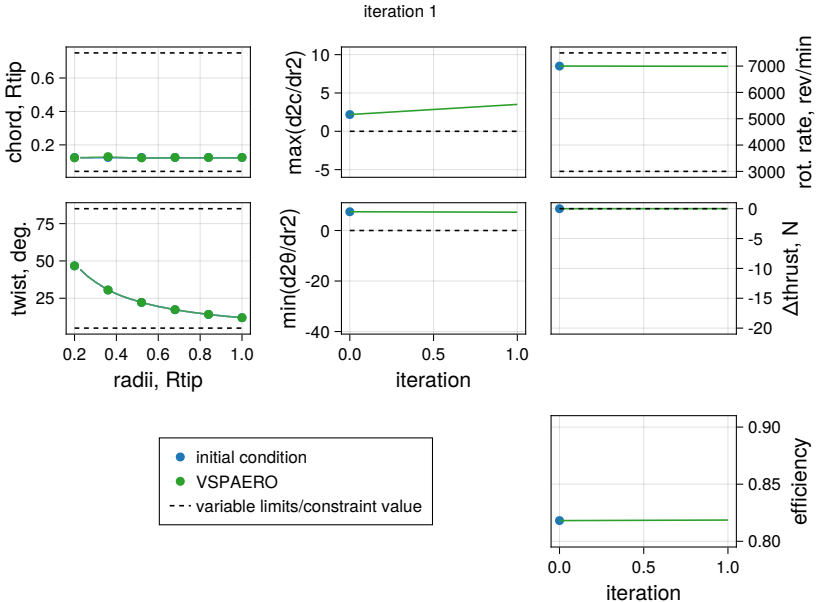
Effects of twist show up in CCBlade.jl's angle of attack



Concavity constraints

- Observation: propellers/proprotors have a typical chord and twist distribution
  - chord distribution: concave down (second derivative wrt radius < 0)
  - twist distribution: concave up (second derivative wrt radius > 0)
- Will adding constraints on the chord and twist concavity help the optimizer produce a more reasonable design?

Adding the chord and twist concavity constraints help



Theories/Ideas

- VLM and BEMT handle drag much differently
  - BEMT: directly coupled as part of the BEMT calculation
  - VLM: ad hoc viscous loading model added to inviscid VLM solution
- BEMT is much less "coupled" than VLM
  - BEMT: solution at each radial location completely independent of all others
  - VLM: solution at given point mathematically affects everything everywhere (though drops off quickly with increasing distance)
- How is VSPAERO able to create smooth loading distributions from not-so-smooth chord and twist?
  - Visualizing the solution in a more detailed way may help
- Is the finite difference method killing us?
  - Used for geometry parameterization + aerodynamic `Group` derivatives
  - Possible cause of convergence troubles, and anomalous isolated, acoustically constrained results (see paper)

Thanks!

Special thank you to:

- Rob McDonald, OpenVSP developer
- Tim Brooks, NASA Glenn/BQMI
- OpenMDAO development team
- NASA Transformational Tools & Technologies Project

Tools:

- OpenVSP/VSPAERO: `openvsp.org`
- pygeo: `github.com/mdolab/pygeo`, Univ. of Michigan MDO lab
- CCBlade.jl: `github.com/byuflowlab/CCBlade.jl`, BYU FLOWLab
- AcousticAnalogies.jl: `github.com/OpenMDAO/AcousticAnalogies.jl`, NASA
- OpenMDAO: `openmdao.org`, NASA
- SNOPT: `ccom.ucsd.edu/~optimizers/solvers/snopt`
- pyOptSparse: `github.com/mdolab/pyoptsparse`, Univ. of Michigan MDO lab