Airframe Design Write Up

Weston Pace

November 2022

1 Optimization

Optimization is not an uncommon idea. As humans, we want to get the best version of something that we possibly can; what is the purpose of using something if it can be better? Inevitably, the importance of optimization translates directly to engineering principles.

Optimization can be relatively simple or extremely complicated. Perhaps we are optimizing the area of a fenced off space that contains only two variables, or we are designing an aircraft with hundreds of variables. Regardless of the scale or complexity of the problem, the foundational principles of optimization remain constant. As Ning and Martins explain in their book "Engineering Design Optimization", optimization consists of a few parts: an objective function, design variables, and constraints. The design variables are the things that are being changed in order to get an optimal solution. The amount of design variables can vary from being a few to a few hundred or even more. The objective function is the value that we are trying to optimize; this can range from the lift coefficient on an aircraft to the mileage on a car. The objective function is made up of the design variables and uses them to compute the value that we are trying to optimize. Finally, the constraints are used to put restrictions on what our variables can be. This can range from a simple domain of values that a variable can rest inside or a structural coefficient that must be met in order for the solution to be legitimate.

On a quick note, optimization processes come in a lot of different forms. They can range from gradient based solvers to methods that evaluate a lot of different possible solutions at the same time and brute force their way to a solution. The method that is used for optimization is dependent on the situation at hand, and it takes a bit of intuition to find the best method for a specific problem.

2 Goals for Assignment Three

For assignment three, design an aircraft that:
- Increases my objective value that I determine
- Can lift 0.5 kilograms and doesn't exceed a wing span of 1.5 meters

- Is statically and dynamically stable

3 Which Optimizer to Use

Julia has a decent selection of optimizer programs to run for a problem like this. I originally was using Optim.jl, and overall, it was good to use. The set up for it was easy, and it could get good solutions out. However, I ran into problems using Optim.jl regarding constraints and optimization efficiency. Due to this, I switched to SNOW.jl, which was put together by Professor Ning. This package ran very well, and met the requirements that I needed for this assignment.

4 Design Variables

VortexLattice gives a range of aerodynamic variables to adjust for the wing, horizontal stabilizer, and vertical stabilizer. These include span, sweep, dihedral, twist, chord length, and positioning along the airframe. Each of these variables uniquely affects the stability and efficiency of the aircraft. For example, adjusting the geometry of the wing can increase the lift of the aircraft; dihedral can stabilize roll; and the stabilizers' positions and geometry can stabilize pitching and yaw motion. I was feeling ambitious, so made almost all of the components for the stabilizers and wing a design variable. This led me to have around 18 variables to work through. I soon realized that I was not ready for an optimization of this size, so I needed to downgrade. After talking with my mentor, I realized that having twist as a design variable was not really necessary because I can just initially pick a good twist angle myself without delegitimizing my optimization. Also, VortexLattice uses Lifting Line theory, which is an inviscid method. This means that the optimizer is just going to crank angle of attack to the max because there is no stall to worry about. In short, this would have not been a very helpful design variable. Furthermore, I chose to just assign the chord lengths a value myself because I wanted to have realistic chord lengths that the optimizer would have to work around instead of the optimizer picking weird values for them. Also, a lot of the effects that chord length has on a plane can be created through changing the span of the wing, so I decided this was a fair adjustment to make. After these changes were made, I was left with the stabilizer and wing sweep, dihedral, and span as design variables. I also had the horizontal and vertical stabilizers' positions as a variable, and I forced them to share the same position instead of moving independently of one another.

5 Conditions for My Optimization

Since the assignment wants a plane that is statically and dynamically stable and can lift a load of 0.5 kilograms, I went to work to find some values that I could make constraints in order to satisfy these needs.

Firstly, I know that a mass of 0.5 kilograms applies a force of 4.9 Newtons, so the lift generated by my plane must be able to create a force of at least 4.9 Newtons. To find this force, I used the equation:

$$L = \frac{1}{2}\rho V_\infty^2 A C_L$$

Using this equation would allow me to force my optimization to be able to lift the necessary mass.

I also wanted my plane to be dynamically and statically stable, so I began to look at what modes of flight are common in an aircraft. Some of these modes include phugoid motion; short-period oscillation; dutch roll; and a slow spiral. Each of these have corresponding stability derivatives that tell whether the aircraft can stabilize itself if it falls into one of these undesirable modes of flight or whether the plane will lose control and inevitably crash. I figured out whether these specific derivatives that correspond to these modes were supposed to be positive or negative from Professor Ning's book "Flight Vehicle Design".

To combat a Phugoid and Short-Period Oscillation flight mode, we want the $C_{m\alpha}$ stability derivative to be negative. This means that as the plane begins to pitch up or down, the aircraft will naturally create a pitching moment that will bring the plane back to a stable state.

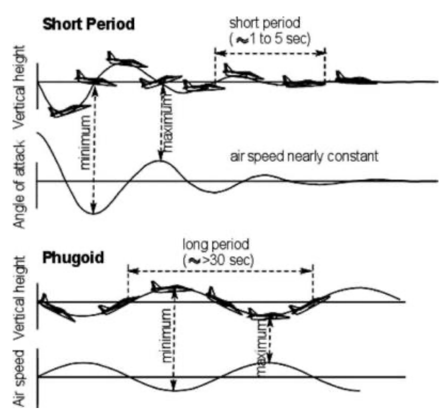

Figure 1: Illustration of Phugoid and Short-Period Oscillation modes (courtesy of Studfiles.net)

For the dutch roll, we want the C_{lb} stability derivative to be negative because as the plane begins to roll, we want a negative moment in order to bring the plane back to a resting state.

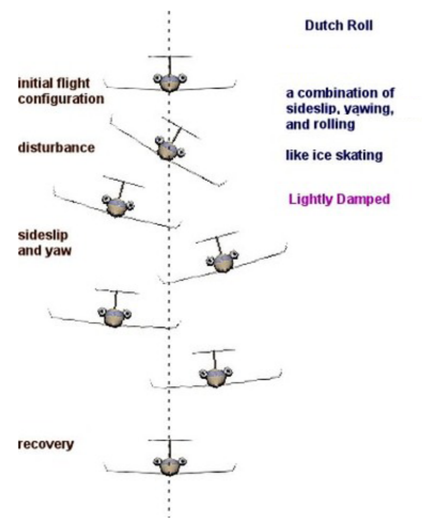

Figure 2: Illustration of the Dutch Roll mode (courtesy of Leeham News and Analysis)

And for the slow spiral, C_{nb} must be positive since a positive yawing moment is desirable when a plane begins to spiral out of control. The reason that this derivative is positive and not negative like the others is because under standard sign convention, a positive yaw moment correlates to a negative beta angle. This is different from the fact that for $C_{m\alpha}$, a positive pitching moment corresponds to a positive α value.

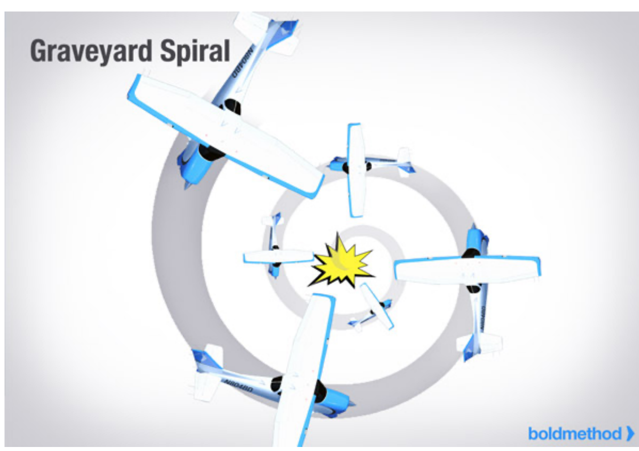

Figure 3: Illustration of the Slow Spiral mode (courtesy of Boldmethod)

6 Objective Value

Picking a good objective value is one of the most important parts of an optimization problem, and it is also one of the more tricky parts. It's easy to choose an objective value that should be a constraint and vice-versa. After considering what I would want out of my aircraft, I chose to make the lift-to-drag ratio my objective value. A good plane should have a high lift to drag ratio, so this seemed like a fitting value to optimize.

7 Final Results

All of these values are in units of meters, and each constraint was satisfied through the optimizer.

Additionally, I assumed that my plane was flying at $22\frac{m}{s}$ and at an angle of attack of 2^o.

Wing	Value	Horizontal Stabilizer	Value	Vertical Stabilizer	Value
xle	0.1	xle	0.2	xle	0.1
yle	0.5	yle	0.3	yle	0.0
zle	0.01	zle	0.001	zle	0.1
chord	[0.3, 0.2]	chord	[0.2, 0.2]	chord	[0.2, 0.1]
theta	0.0	theta	0.0	theta	0.0
phi	0.0	phi	0.0	phi	0.0
position	0.0	position	2.0	position	2.0

7.0.1 Final Design

Wing	Value	Horizontal Stabilizer	Value	Vertical Stabilizer	Value
xle	0.5	xle	0.5	xle	0.3
yle	0.75	yle	0.3	yle	0.0
zle	0.2	zle	0.0	zle	0.3
chord	[0.3, 0.2]	chord	[0.2, 0.2]	chord	[0.2, 0.1]
theta	0.0	theta	0.0	theta	0.0
phi	0.0	phi	0.0	phi	0.0
position	0.0	position	1.7	position	1.7

Each of the data in the table refers to half of the wing and horizontal stabilizer. The other half of these components are just mirrored when analyzed by VortexLattice. Additionally, xle refers to a length going parallel with the aircraft's body; yle refers to a length going perpendicular and away from the airframe (like the wing span); and zle refers to a length going perpendicular and away from the yle's axis (like the span of vertical stabilizer). The chord vectors show the length of the root chord and the tip chord respectively. Theta is the twist of each aerodynamic component, and phi is the twist of the airfoils. Additionally, the position of each of these airframe components is with respect to the center of mass of the aircraft.

7.0.2 Changes In Design

There were some obvious changes from the initial wing design to the final. The sweep increased from 0.1 to 0.5; the half span increased from 0.5 to 0.75; and the dihedral went from 0.01 to 0.2. For the horizontal stabilizer, the only two things that really changed were the sweep, which went from 0.2 to 0.5, and the position, which went from 2.0 to 1.7. Finally, the vertical stabilizer changed in its position from 2.0 to 1.7; the sweep went from 0.1 to 0.3; and the span changed from 0.1 to 0.3.

Overall, there were pretty significant adjustments through the optimization process, but there is a chance that some of these values could have changed even more if I adjusted the constraint domain on my variables.

7.0.3 Comparison Between Initial and Final Design

Final Design	Values	Initial Design	Values
Lift to Drag Ratio	124.249	Lift to Drag Ratio	83.06
Lift Force	17.484 N	Lift Force	11.5783 N
C_{ma}	-0.0384	C_{ma}	1.4
C_{nb}	0.282	C_{nb}	0.117
C_{lb}	-0.0199	C_{lb}	0.0007

8 Takeaways

A major thing that I learned from this assignment is that it is best to take optimization one slow step at a time. I was very quick to jump into an advanced problem, and when problems arose, I didn't know where to start fixing them from. It was only when I broke the problem down into smaller pieces that I was able to get the results that I needed.