UNIVERSITY OF CALIFORNIA

Santa Barbara

Towards a Jump-Glider Optimization Tool

A Thesis submitted in partial satisfaction of the
requirements for the degree Master of Science
in Mechanical Engineering

by

Andrew James Boyles

Committee in charge:

Professor Elliot Hawkes, Chair

Professor Paolo Luzzatto-Fegiz

Professor Igor Mezic

June 2019

The thesis of Andrew James Boyles is approved.

_____

Igor Mezic

_____

Paolo Luzzatto-Fegiz

_____

Elliot Hawkes, Committee Chair

June 2019

Towards a Jump-Glider Optimization Tool

Andrew James Boyles

ACKNOWLEDGEMENTS

I would like to thank all of the friends I have made during my time at this University. I would like to thank Rochi for helping me understand optimization. I would like to thank the members of Hawkes Lab: Nick, Charlie, David, Bryan, Aaron, and Sicheng. And, most of all, I would like to thank Dr. Elliot Hawkes for directing and assisting my research for my time here, and Kenny Chui for helping me with the more menial tasks this research required and for always being there to talk to about my next steps.

ABSTRACT

Towards a Jump-Glider Optimization Tool

by

Andrew James Boyles

In recent years, interest has been shown in jump-gliding devices. A few devices have been made that can successfully outperform a ballistic projectile using low Reynolds Number glides. This thesis presents the attempts to develop a high Reynolds Number jump-glider, using both prototyping means and modeling means. The descriptions of these processes include discussions on testing methods, computational trajectory models, and optimization fundamentals. This research has culminated in the framework of a tool that can be used to design optimal jump-gliders.

Contents

List of Figures

Chapter 1

Introduction

Jump-gliders are devices that combine the self-contained jumping abilities of jumping robots with glider constructions in order to travel distances that jumping robots alone could not travel. In [1], a jump-glider was created that was able to glide large distances using a rotation mechanism on its wing to set the wing in place. In [2], a jump-glider was manufactured that could steer itself using an adjustable rudder, but it needed to be launched from an initial height of $2\ m$ in order to surpass the distance traveled by an equivalent ballistic mass. In [3], a jump-glider was designed that could surpass a ballistic projectile while launching itself from the ground.

While these jump-gliders are great strides in the development of efficient methods of locomotion, the papers do not discuss the design of the glider mechanisms in detail. The placement of the wings (in this paper, 'wings' refers to every wing element on the glider, including the wing and the elevator), the sizes of the wings, and the angles they are mounted at, relative to the fuselage. These parameters have significant effects on the glider's performance.

In this thesis, I will be discussing the motivation and the design of a jump-glider optimization tool. First, Chapter 2 talks about my attempts to create a jump-glider by hand. Those results motivate the mathematical model of the jump-glider derived in Chapter 3. The model is used in Chapter 4 to optimize the parameters outlined above. From there, Chapter 5 looks to the future of the tool and the design of my jump-glider, and Chapter 6 concludes the thesis.

Chapter 2

Building a Jump-Glider

2.1 Jumper

During his work at Stanford, Dr. Elliot Hawkes developed a jumping robot (Figure 2.1). It has two servo motors which wind the rubber bands by spinning a gear along a track. The two flag-like protrusions guide a string that pulls the gear system off of its track. Once it's off the track, the rubber bands release their energy and launch the robot. It is estimated that the robot pulls the rubber bands about $25\ cm$ to input a potential energy of about $19\ J$, and it is launched at around $24.7\ m/s$.

2.2 Converting to a Jump-Glider

Using Hawkes' jumper as the input energy source, it could be converted from a jumper to a jump-glider by putting it within a glider shell. The jumper would jump from within the shell, and, if it is constructed well, the glider shell would carry the jumper a further distance than if it had jumped alone.

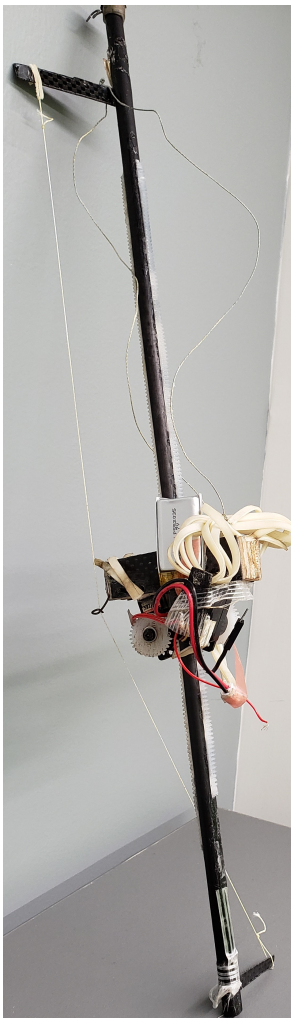

Figure 2.1: Jumper

2.3 Glider Materials

2.3.1 Fuselage

The jumping robot was to be converted into a jump-glider by replacing the carbon-fiber rod with a lightweight, tube-like fuselage that would house the jumping components and have wings attached to the exterior. Initially, carbon-fiber was considered for the fuselage, due to its relatively low density and its rigidity, but there are other materials with lower densities that can also be used. There are foam materials with extremely low densities that can be put into tube shapes, but the wall thicknesses of those tubes are much thicker than carbon-fiber tubes,

so they would greatly increase the drag on the fuselage. Apogee sells thin cardboard tubes that have lower densities than carbon-fiber, and with thinner wall thicknesses. These tubes had a small enough thickness to be lighter than the carbon-fiber tube used for the jumping robot, while being resilient enough to be launched more than once.

2.3.2 Wings, Elevators, and Rudders

Figure 2.2: Rip-Stop Nylon Wing Glider

There were multiple wing designs for the glider. The first glider design used rip-stop nylon for the wing and fiberglass for the elevator and the rudder (Figure 2.2). It was initially guessed that the Reynolds number of the glide would be around 50,000, and around that Reynolds Number, rectangular wings with an Aspect Ratio (AR) of 6 are just about as aerodynamically viable as wings with higher ARs, making a wing with an AR of 6 ideal due to its lower weight [1]. The wing was given a semicircular shape so that its frame could be constructed using a thin carbon-fiber rod and string, and was designed with an aspect ratio of 6 to achieve a close performance to a rectangular wing at low Reynolds Numbers.

Later, for reasons discussed below, the wing, elevator, and rudder were replaced with balsa wood (Figure 2.3). The balsa wood is more dense per unit area than the fabric, so the wingspan had to be reduced in order for the balsa wood wings to be able to have an aspect ratio of 6 while keeping the weight of the glider about the same.

Figure 2.3: Balsa wood wings

2.3.3 Other Materials

The hex key is used as a hook for the launcher used to test the glider's performance. It added the benefit of adding mass to the front of the glider, however, it was not enough to move the center-of-mass near the front of the glider. To fix this, a 10 x 25 mm hex bolt was taped inside the fuselage, at the front of the glider. Note that the completed jump-glider would not have the hex key, since it would launch itself using a motor and rubber tubing.

2.3.4 Estimated Reynolds Number of Glide

The Reynolds Number is a factor that influences the design of gliders. Higher Reynolds Numbers usually correspond to more turbulent flows [4], which affect the aerodynamic forces exerted on the glider. Because of this, different wings are suited to different levels of turbulence [5]. Thus, it is an effective design tool to base designs around the estimated Reynolds Number

of the glide.

The Reynolds Number is calculated with the equation

$$Re = \frac{\rho u L}{\mu}, \tag{2.1}$$

where ρ is the density of the fluid, u is the flow velocity, L is the characteristic length, and μ is the dynamic viscosity of the fluid [4]. Because this Reynolds Number is being used to quickly characterize the entire glide, approximate values were used to determine a 'ballpark' estimate of the highest order of magnitude the Reynolds Number would achieve during the glide. At atmospheric pressure and room temperature, air's density is around $1.204\ kg/m^3$ [6], and the dynamic viscosity is around $18.13 * 10^{-6}\ Ns/m^2$ [7]. The characteristic length of a glider is the chord length of its wing. The rip-stop nylon wing had a larger chord length ($9.6\ cm$) than the balsa wood wing, so that was used for the Reynolds Number estimate. The maximum flow velocity that the glider experiences is the launching speed of the jumping robot, which is $24.7\ m/s$. Plugging these parameters into equation (2.1), the Reynolds Number is calculated as 160,000. Despite the Reynolds Number being higher than the category discussed above (50,000), the wing designs were kept in order to focus on achieving a glide, rather than figuring out the optimal glide from the start.

2.4 Launch Setup

Because many iterations of the two main glider designs were tested, and it would have been time-consuming to iterate the design of a self-launching jump-glider, an external launcher was constructed in order to replicate the launching conditions; specifically the input energy. Shown in Figure 2.4, the launcher used rubber tubing to replicate the energy input from the jumping robot ($19\ J$). The "spring constant" of the rubber tubing can be found using the elastic stress-strain relationship

$$\sigma = E\epsilon, \tag{2.2}$$

Figure 2.4: Glider Launcher

where σ is the elastic stress on the tubing, ϵ is the elastic strain the tubing experiences, and E is the tubing's Modulus of Elasticity. Right before launching the jump-glider, the tubing has no acceleration, so the force pulling the tubing is equivalent to its "spring force", F_k,

$$\sigma = \frac{F_k}{A_c}, \tag{2.3}$$

where A_c is the tubing's cross-sectional area. The engineering strain on the tubing depends on the change in its nominal length, l

$$\epsilon = \frac{\Delta l}{l}. \tag{2.4}$$

Plugging Equations 2.3 and 2.4 into Equation 2.2 and solving for the "spring force" gives a Hooke's Law equation

$$F_k = \frac{A_c E}{l}\Delta l = k\Delta l, \tag{2.5}$$

where k is the "spring constant" of the tubing. The potential energy stored by the rubber tubing, U, can be calculated using this constant

$$U = \frac{k(\Delta l)^2}{2}. \tag{2.6}$$

Instead of measuring the parameters in Equation (2.5) and solving Equation (2.6) for Δl, it was more convenient to calculate the potential energy by measuring the distance that the tubing needs to be stretched in order to produce $75N$ of force (chosen arbitrarily) and applying Hooke's Law to get the "spring constant". The distance, x, that the tubing needs to be stretched to store (approximately) the correct amount of energy is calculated with

$$x = \sqrt{\frac{2U}{k}}. \tag{2.7}$$

Because the equivalent spring constant changes over the tubing's lifespan, the amount of energy it gives to the glider is approximate, which was allowable for these proof-of-concept tests.

The carbon-fiber rods were used to guide the glider along the ramp by having the wing slide on them. The rubber tubing was stretched by pulling on a string looped around the front of the tubing. A loop at the end of the launcher allowed the string to be pulled vertically. The rods are smooth enough that the friction between the glider and the rods consumes a negligible amount of energy.

The glider is initially suspended off the ground by the launcher, but the amount of added potential energy is negligible since the glider's mass is relatively low (about $50\ g$).

2.5 Results

Neither glider was able to achieve a gliding distance that was visibly comparable to launching the glider without the wings (as a ballistic projectile), even after trying many different mounting

angles for the elevators for each glider. Various factors contributed to this result.

2.6 Discussion

At first, the rip-stop nylon glider was tested by throwing it by hand. When the glider was able to glide when hand-thrown, it was then tested using the launcher. These tests revealed that the difference in Reynolds Numbers between hand-throwing the glider and launching the glider was high enough to render the hand-throwing tests useless. Gliders that were able to glide when thrown by hand were unable to glide when launched at much higher speeds.

On top of this, many factors increased the time it took to test the gliders.

One downside of the cardboard tube fuselage was its fragility. If launched in a way that imposed an abundance of normal stress, the fuselage would begin to buckle, which likely added drag and lateral forces to the trajectory of the glider. The fuselage was able to work between twenty and thirty tests before needing to be replaced, meaning that varying the glider's parameters came at a higher time cost than assumed at first.

There were three major problems with the rip-stop nylon wing design. First, the rip-stop nylon would often tear, so it needed to be fixed often, or, even worse, it would throw off data for a few tests before it was fixed. Second, rip-stop nylon is difficult to adhere. It resists most glues and tapes; only SilPoxy seemed to work, and that was only for a small number of tests. I eventually sewed the wing onto the fuselage, which made the glider's fabrication much more time consuming than it needed to be. Lastly, as can be seen in Figure 2.5, the wing was very difficult to keep taut. Various string configurations were tested to try to pull the wing taut, but none of them worked well with the thin frame.

Additionally, when there was not a sufficient rudder, the gliders would occasionally veer in the lateral directions.

The elevator and rudder also caused problems because the fiberglass was too flimsy to easily attach to the fuselage. The rudder had to be covered in tape and the fuselage had to be cut open to make a slot for it, meaning that the fuselage would have to be replaced if it was cut

Figure 2.5: Attempts to make wing taut

wrong. For these reasons, the wing, the elevator, and the rudder were then made from balsa wood. Unfortunately, after many more tests, with similar variations in the glider's parameters, this glider also did not visibly compare to launching the fuselage.

2.7 Motivation for Mathematical Modeling

Because many tests had not lead to the design of a successful glider (one that was able to exceed the distance flown by the fuselage alone), and because the glider design would need to be changed after the jumping capabilities were added, which would have required more tests, it was decided that the glider should be designed with the assistance of mathematical modeling. Due to the relative mathematical simplicity of the balsa wood wings, when compared to the rip-stop nylon wing, it was also decided that the jump-glider would be modeled with balsa wood wings.

Chapter 3

Physics Model

3.1 Motivation

The first goal of the mathematical design of the glider was to model a configuration's trajectory. A 'configuration' is defined by its unique combination of parameters that were being varied in the design of the glider: the positions of the wings, the mounting angles of the wings, and the sizes of the wings. The trajectory of a glider can be determined with knowledge of the four forces acting on it: thrust - the force propelling the airplane forward, lift - the force on the airplane perpendicular to the direction of the ambient air flow, drag - the force in the direction of the ambient air flow, and the weight of the glider. For an airplane, thrust can be provided by a propulsion system, but a glider does not have one. The main thrust that the glider will utilize will be the lift force when the glider is moving downward; the lift force will have a vector component driving the glider forward. For a stable glide, lift will balance the weight of the glider, and the flow will point the lift force vertically in order to cancel the weight [8]. The optimal glider will have the wings placed in such a way to achieve this balance for the speeds it is intended to glide at.

3.2 Basis

The trajectory of a glider can be derived by integrating Newton's Second Law on the center-of-mass of the glider. Assuming that lateral forces and torques do not have a large effect on the trajectory, these equations take the form

$$M\ddot{p}_x = F_x \tag{3.1}$$

$$M\ddot{p}_y = F_y \tag{3.2}$$

$$I\ddot{\beta} = T_z, \tag{3.3}$$

where M is the mass of the glider, p is the position of the center-of-mass, F_x is the sum of the horizontal forces acting on the glider, F_y is the sum of the vertical forces acting on the glider, I is the glider's moment of inertia (in the pitching direction), β is the angle that the glider's elevator makes with the horizontal axis , and T_z is the sum of the pitching torques acting on the glider (Figure 3.1) [9].

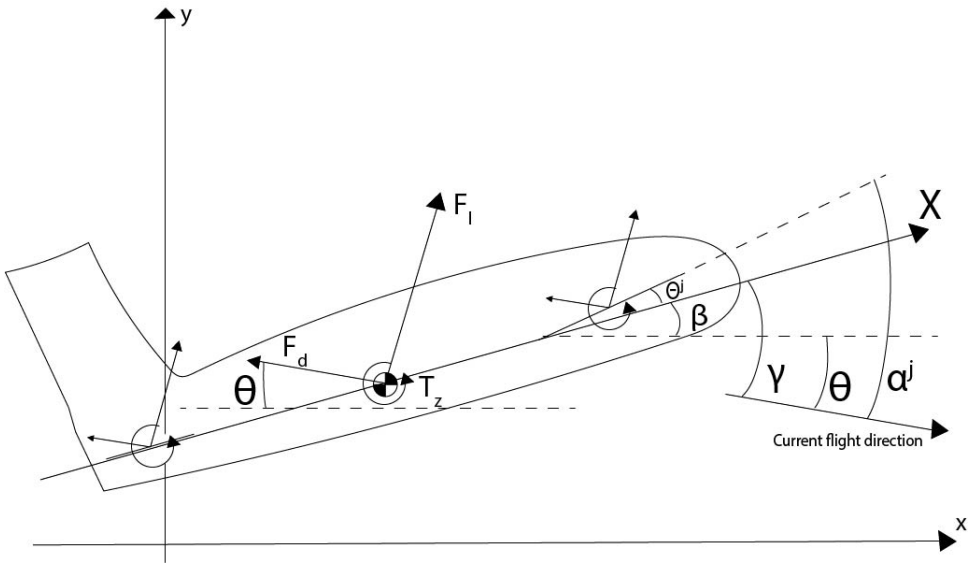

Figure 3.1: Glider angles (adapted from [9])

The forces and torques come from the weight of the glider, and the drag and lift forces acting on the glider, which are dependent on the glider's angle-of-attack, the angle between the glider's elevator and the flow acting on the glider. For complicated wing shapes, wings can be broken up into "wing elements", small rectangular sections of the wing, each with different drag and lift forces and pitching torques (Figure 3.2). The forces and torque on each of these elements is [9]

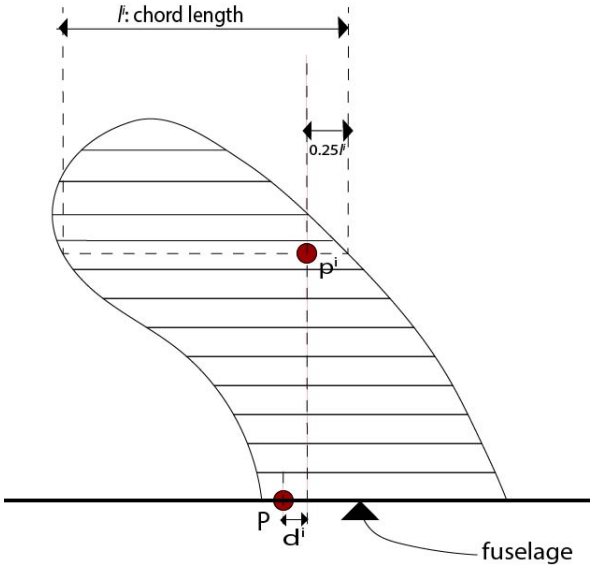

Figure 3.2: Discrete wing elements (adapted from [9])

$$f_d^i = \frac{c_d \rho V^2 a^i}{2} \tag{3.4}$$

$$f_l^i = \frac{c_l \rho V^2 a^i}{2} \tag{3.5}$$

$$\tau_z^i = \frac{(c_m l^i + c_d d^i sin(\alpha) + c_l d^i cos(\alpha))\rho V^2 a^i}{2}, \tag{3.6}$$

where f_d^i is the drag force on the wing element, i, c_d is the coefficient of drag on the wing, f_l^i is the lift force on the wing element, c_l is the coefficient of lift on the wing, τ_z^i is the pitching torque on the wing element, c_m is the pitching moment coefficient on the wing, l^i is the chord length of the wing element, α is the wing's angle-of-attack (this is slightly different

than the glider's angle-of-attack; the glider's angle-of-attack is the angle between the elevator and the flow velocity, while the wing's angle-of-attack is the angle between the wing and the flow velocity), a^i is the area of the wing element and d^i is how far p^i, the aerodynamic center of the wing element, is from the wing's mounting point, P, in the upwind direction [9]. The pitching torque equation is the result of the aerodynamic center being a correction for a moving center-of-pressure on the wing. The center-of-pressure is the actual point on the wing where the resultant drag and lift forces act. However, at different angles-of-attack, the center-of-pressure moves along the wing, making it difficult to track the center-of-pressure of the wing for different angles-of-attack, so the pitching torque equation allows the drag and lift to be calculated as if they were actually acting on the aerodynamic center of the wing. Instead of evaluating the location of the center-of-pressure for each different angle-of-attack, the necessary pitching moment coefficient is used to produce the correct equivalent pitching torque. The aerodynamic center of the wing is used because it is the point on the wing for which the pitching moment needed to make up this difference does not vary greatly with the angle-of-attack [10]. Because this analysis is being done on the rectangular balsa wood wings, the pitching moments on them are zero, due to the aerodynamic centers of the wings being in the same location as the centers-of-pressure for angles-of-attack well below stall [11], and a^i is the same for every wing element, meaning that the forces on the wing are

$$f_d^j = \frac{c_d^j \rho V^2 A^j}{2} \tag{3.7}$$

$$f_l^j = \frac{c_l^j \rho V^2 A^j}{2}, \tag{3.8}$$

where A^j is the area of each wing.

The coefficients of drag and lift on a wing depend on its angle-of-attack, which changes during flight. For this reason, it is necessary to reference polars (graphs of the coefficients as functions of the angle-of-attack) to accurately calculate the drag and lift forces on the wings during flight. I used polars for a Clark Y airfoil with a Reynolds Number of 50,000 and an N_{crit} of 9 from [12] based on advice from one of the authors of [9], Nobuyuki Umetani.

Because the glider has a large fuselage intended to house the jumping robot, the drag force on the fuselage cannot be neglected. The drag on the fuselage depends on the angle between the vector normal to the cylindrical body of the fuselage and the flow velocity (Figure 3.3)

$$\phi = \frac{\pi}{2} - \gamma + \psi. \tag{3.9}$$

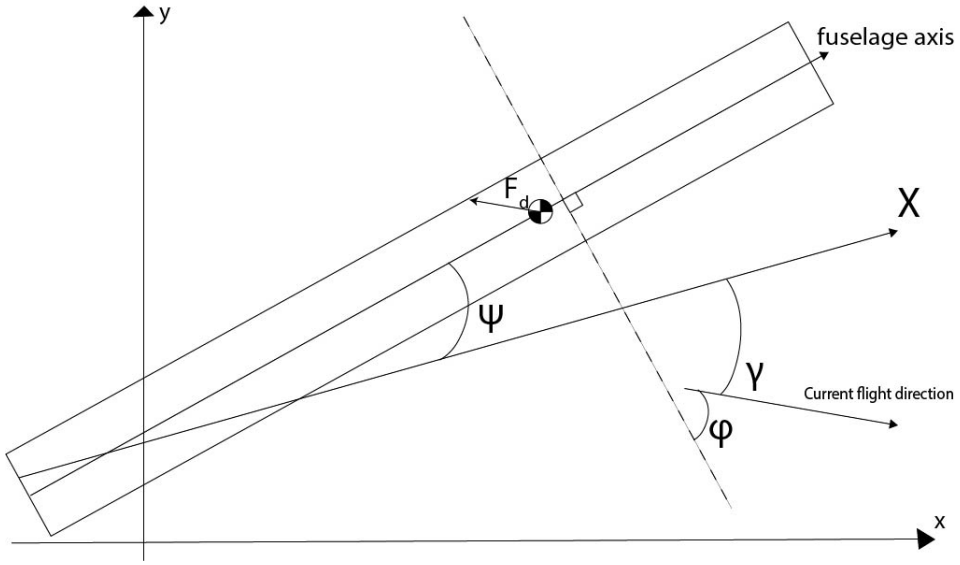

Figure 3.3: Fuselage angles (adapted from [9])

This angle determines the surface area of the fuselage that the flow velocity 'sees' and is able to exert drag on

$$A^f = \frac{\pi(d^f)^2 sin(\phi)}{4} + \pi d^f L cos(\phi)), \tag{3.10}$$

where the 'f' superscript denotes the fuselage. The drag force on the fuselage is

$$f_d^f = \frac{c_d^f \rho A^f V^2}{2}. \tag{3.11}$$

The total drag and lift forces on the glider are calculated with sums [9]

$$F_d = \sum_{j=1}^{\#\ elements} f_d^j \tag{3.12}$$

$$F_l = \sum_{j=1}^{\#\ elements} f_l^j. \tag{3.13}$$

In order to calculate the glider's acceleration in the x and y directions, the forces must be broken into their x and y components. These components are dependent on the glider's 'defending' angle, the angle between the global x axis and the flow velocity (Figure 3.1) [9]

$$F_x = -F_d cos(\theta) + F_l sin(\theta) \tag{3.14}$$

$$F_y = F_d sin(\theta) + F_l cos(\theta). \tag{3.15}$$

Because the aerodynamic centers of the wings and the center of the fuselage are not at the center-of-mass of the glider, the drag and lift forces produce a pitching torque. The "moment arms" for this pitching torque are the perpendicular distances between the center-of-mass and the aerodynamic centers of the wings and the perpendicular distances between the center-of-mass and the center of the fuselage. These distances are dependent on the angle-of-attack, the angle between the glider's elevator and the flow velocity (Figure 3.4) [9]

$$d_h^j = x_{ac}^j cos(\gamma) - y_{ac}^j sin(\gamma) \tag{3.16}$$

$$d_v^j = x_{ac}^j sin(\gamma) + y_{ac}^j cos(\gamma), \tag{3.17}$$

where x_{ac}^j is the distance along the axis of the glider between the wing's (or fuselage's) aerodynamic center and the center-of-mass, and y_{ac}^j is the distance along the axis normal to the glider between the wing's aerodynamic center and the center-of-mass. x_{ac}^j is always negative because the aerodynamic centers of the wings are behind the center-of-mass of the glider, and y_{ac}^j is always positive because the aerodynamic centers of the wings are always higher on the glider than its center-of-mass (unless the glider turned upside-down, which is not a desired trajectory).The pitching torque on the glider is then [9]

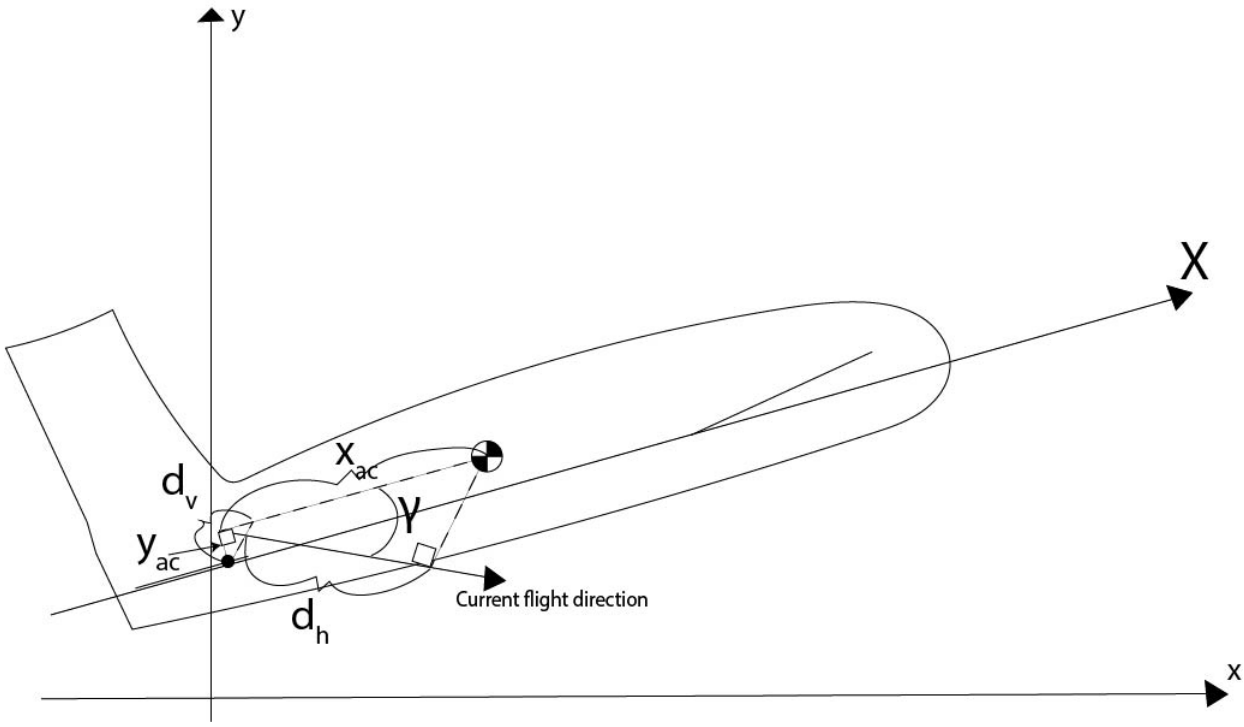

Figure 3.4: Pitching torque moment arms (adapted from [9])

$$T_z = \sum_{j=1}^{\#\ elements} d_h^j f_l^j + d_v^j f_d^j. \tag{3.18}$$

Now that the forces and torques on the center-of-mass are known, equations 3.1 - 3.3 can be integrated to calculate the acceleration of the glider [9], which is done using Verlet Integration.

3.3 Verlet Integration

Verlet integration is a technique used to numerically integrate equations. For the first time step, the second position of the glider is estimated as [13]

$$\overrightarrow{p}_1 = \overrightarrow{p}_0 + \overrightarrow{p}_0 \Delta t + \frac{\overrightarrow{\ddot{p}}_0 \Delta t^2}{2} \tag{3.19}$$

$$\beta_{n+1} = \theta + \overrightarrow{\dot{p}}_n \Delta t + \frac{\overrightarrow{\ddot{p}}_n \Delta t^2}{2}, \tag{3.20}$$

where t denotes time. For every other time step, the position can be estimated as

$$\overrightarrow{p}_{n+1} = 2\overrightarrow{p}_n - \overrightarrow{p}_{n-1} + \overrightarrow{\ddot{p}}_n \Delta t^2 \tag{3.21}$$

$$\beta_{n+1} = 2\beta_n - \beta_{n-1} + \ddot{\beta}_n \Delta t^2, \tag{3.22}$$

where n denotes the current time step [13].

The velocity is calculated using a first order Euler approximation

$$\overrightarrow{\dot{p}}_{n+1} = \frac{\overrightarrow{p}_{n+1} - \overrightarrow{p}_n}{\Delta t}, \tag{3.23}$$

and this process is iterated for a set amount of time steps.

3.4 Verification

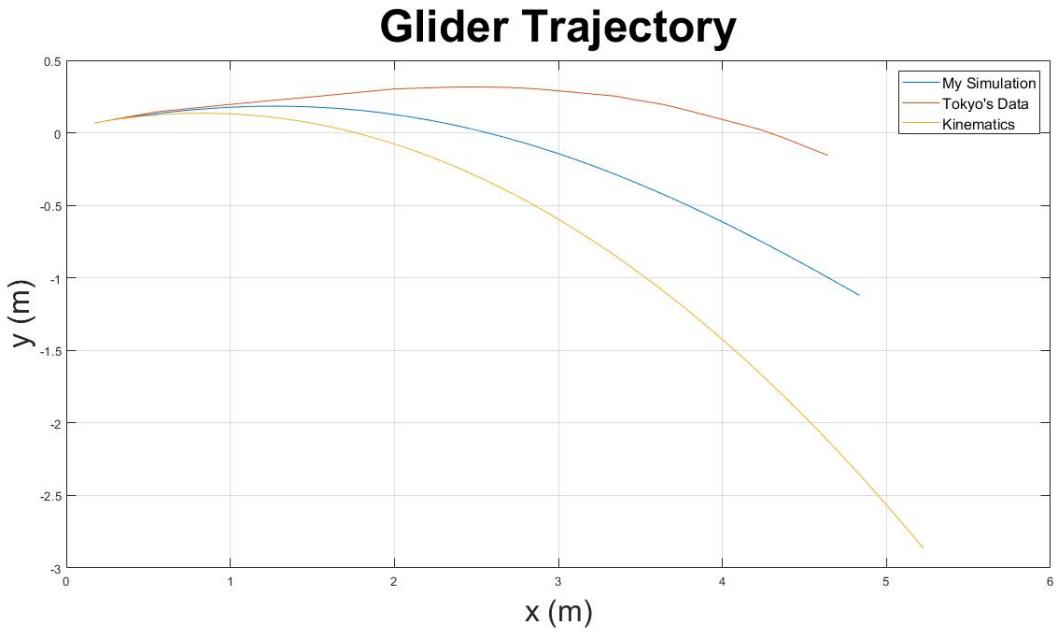

Figure 3.5: Simulation comparison to data from the University of Tokyo

The University of Tokyo provided data of the measured and simulated trajectories shown in [9] for the 2014 SIGGRAPH conference in Vancouver. Figure 3.5 is a comparison of their

measured trajectory of one of their gliders to my simulation of that glider; it also includes the kinematic trajectory for a particle launched at the same angle and speed as the glider. In this comparison, the height of the ground is not considered in order to compare the aerodynamic trajectories of the three data sets for the full length of time. The simulation was compared to an actual measured trajectory to ensure that my simulation can accurately simulate real trajectories.

3.4.1 Error Analysis

The error of my simulation was evaluated by recording the maximum difference between the norm of University of Tokyo's data and my simulation's data. The maximum error happened at the end of the simulation, about 7%. The average error along the simulation was about 3%. This result was deemed accurate enough to be used to determine the glider's optimal configuration. The majority of this error likely comes from the coefficients of drag and lift, since the Clark Y airfoil's polars can only approximate the aerodynamic behavior of this glider.

Chapter 4

Glider Optimization

4.1 Brute-Force Optimization

A simple way to optimize a function is to plug in many parameters and save the best result; this method was my first attempt at optimization. This was accomplished in MATLAB with a sequence of nested for loops that calculated the trajectory of the glider for many different configurations. The parameters that were optimized were: the locations of the wings, the mounting angles of the wings, the launching angle, and the sizes of the wings. Despite the estimated Reynolds Number being above 50,000, meaning that a rectangular wing with an AR of 6 may not perform as well as specificed in [1], the AR of 6 was kept while varying the sizes of the wing and the elevator in order to not increase the number of parameters to optimize. The strategy for the optimization was to keep the number of iterations of each for loop small in order to quicken the code's run-time. A coarse search was done first in order to find a region of interest, and then it was zoomed-in on until the maximum horizontal distance the glider travels before hitting the ground increased negligibly. If, after zooming in, the optimization found that the optimal parameter was at one of the bounds of its range, that bound was expanded by an arbitrary amount to allow the actual optimal in that region to be found.

4.1.1 Changing Coordinate Systems

For the physics model, the center-of-mass and the moment of inertia in the pitching direction were found using a SolidWorks model of the glider. However, the optimization ran through a large list of arbitrary parameters, so it would have been tedious to find the new centers-of-mass and the new moments of inertia with the same method.

Instead, to account for the changes in the location of the center-of-mass, the origin of the coordinate system was shifted from the center-of-mass to the front of the glider. More specifically, the new origin of the coordinate system was shifted to the front of the hex key that attaches the glider to the launcher. This was done to avoid confusion in the results of the optimization. If the optimal locations of the wing and the elevator were expressed as being relative to the center-of-mass of the glider, it would become unclear where on the glider the wing and elevator should be located, since moving the wing and the elevator shifts the location of the center-of-mass on the glider. For each iteration of the optimization, the center-of-mass and moment of inertia were calculated based on their "nominal" values, the center-of-mass and moment of inertia from the SolidWorks model. The new center-of-mass, $\overrightarrow{COM_i}$, was calculated by subtracting the mass contributions of the nominal locations of the wing from the nominal center-of-mass and adding the contributions of the new wing and elevator locations

$$\overrightarrow{COM_i} = \overrightarrow{COM_0} - \frac{m_{w,0}\overrightarrow{r_{c,w,0}} + m_{e,0}\overrightarrow{r_{c,e,0}}}{m_0} + \frac{m_{w,i}\overrightarrow{r_{c,w,i}} + m_{e,i}\overrightarrow{r_{c,e,i}}}{m_i}, \tag{4.1}$$

where: $\overrightarrow{COM_0}$ is the location of the nominal center-of-mass in the glider's local coordinates, $m_{w,0}$ is the wing's nominal mass, $\overrightarrow{r_{c,w,0}}$ is the nominal position of the center of the wing, $m_{e,0}$ is the elevator's nominal mass, $\overrightarrow{r_{c,e,0}}$ is the nominal position of the center of the elevator, $m_{w,i}$ is the wing's new mass, $\overrightarrow{r_{c,w,i}}$ is the new position of the center of the wing, $m_{e,i}$ is the elevator's new mass, and $\overrightarrow{r_{c,e,i}}$ is the new position of the center of the elevator. The masses of the wing and the elevator were calculated using their dimensions and the measured density of the balsa wood, which is $93.8 \pm 3.7\ kg/m^3$. The pitching moment of inertia about the center-of-mass, I_i, is calculated in a similar way.

$$
\begin{aligned}
I_i \;=\; & I_0 - m_{w,0}((r_{c,w,0,x} - COM_{0,x})^2 + (r_{c,w,0,y} - COM_{0,y})^2) \\
& -m_{e,0}((r_{c,e,0,x} - COM_{0,x})^2 + (r_{c,e,0,y} - COM_{0,y})^2) \\
& +m_{w,i}((r_{c,w,i,x} - COM_{0,x})^2 + (r_{c,w,i,y} - COM_{0,y})^2) \\
& +m_{e,i}((r_{c,e,i,x} - COM_{0,x})^2 + (r_{c,e,i,y} - COM_{0,y})^2), \qquad (4.2)
\end{aligned}
$$

where I_0 is the nominal pitching moment of inertia about the center-of-mass and the positions of the centers of the wing elements and the positions of the center-of-mass are being used in their x and y-components. The coordinates of the center-of-mass were subtracted from the coordinates of the centers of the wing elements because the origin of the coordinate system was no longer at the center-of-mass, but the moment of inertia was about the center-of-mass.

The optimization varied the parameters and evaluated which combination had the furthest flight. From there, the parameters were plugged into the physics model to isolate that configuration's trajectory.

4.1.2 Results

The optimization was unable to zoom-in to an optimal configuration. Attempting to zoom-in every parameter after each search would cause some of the parameters to try to exceed their new boundaries, and optimizing one parameter at a time produced similar results; the previously optimized parameter would un-optimize while another parameter was being optimized, even to the point of exceeding the parameter's maximum boundaries. A visual example of a function that could lead to some of these difficulties is shown in Figure 4.1, the Rosenbrock Function. Finding the global minimum of this function would be difficult with this brute-force method because the valley does not follow one parameter [14].

If the cost function were two or three-dimensional, it might have been an option to graph the cost function and visualize where the global minimum is, but with six dimensions, it is much more difficult to visualize where the minima are.

This optimization method took a relatively long time to find results. Listing only five options

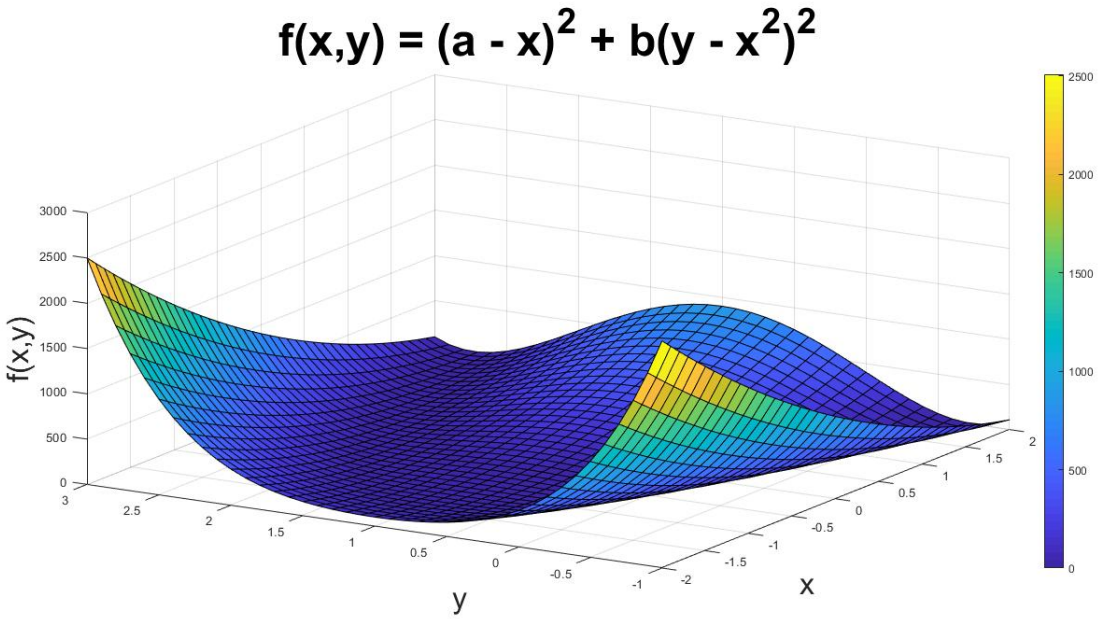

Figure 4.1: The Rosenbrock Function

for each parameter would have the optimization take at least an hour, which became a time-sink when progress was lost when trying to zoom-in. This inability to search with a fine resolution, along with the long amount of time it took for the optimization to run, suggested that a more efficient optimization would be needed to find any desirable results, such as the Gradient Descent Method, the optimization method used in [9].

4.2 Gradient Descent

The Gradient Descent Method finds the minimum of a function by starting at a guess for the minimum and then following the negative gradient in steps to the minimum. Referring to the vector of the function's inputs as x and the function as F, each set of inputs to the function can be found with [15]

$$x_{n+1} = x_n - \gamma_n \nabla F(x_n),\ n \geq 0. \tag{4.3}$$

To find the gradients numerically, the cost functions were calculated in the neighborhood of

each guess. MATLAB's gradient function then gave the correct gradient of the current guess, since it had the values in the neighborhood to compare the guess' cost function to.

4.2.1 Verification

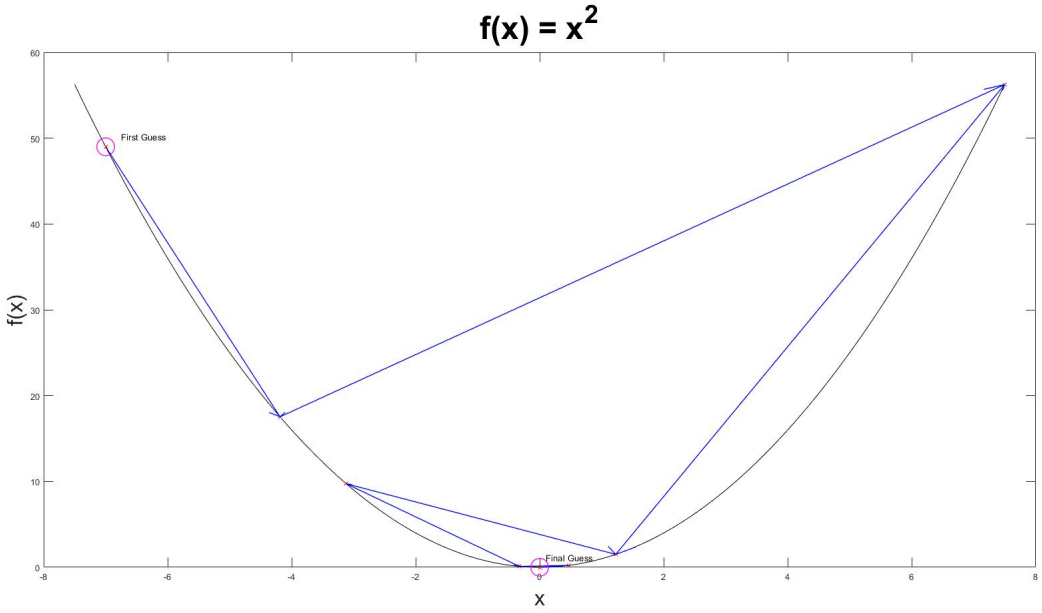

Figure 4.2: A verification of the gradient descent method

The gradient descent code was verified by testing it with simple functions, such as $f(x) = x^2$. Figure 4.2 shows tests of various initial guesses and the path that the gradient descent took to successfully find the minimum.

4.2.2 Cost Functions

The function that was used was a cost-based function based on three costs associated with gliding. The first cost function stabilized the pitch by prioritizing positive pitching torque when the angle-of-attack was negative and negative pitching torque when the angle-of-attack was positive [9]

$$C_1 = \int_{V_{min}}^{V_{max}} T_z^-(V, \gamma_{min}) + T_z^+(V, \gamma_{max}) dV. \tag{4.4}$$

When a glider has stable flight, it will be nearly horizontal, which means the vertical force lifting the glider will almost exclusively come from the lift force it experiences. This lift force must balance the weight of the glider in order to sustain a glide. For this reason, the second cost function was [9]

$$C_2 = \int_{V_{min}}^{V_{max}} ||F_l(V, \gamma) - Mg||^2 dV, \ where\ T_z(V, \gamma) = 0. \tag{4.5}$$

An aerodynamically optimized glider will reduce the drag it experiences in order to glide as far as it can, so the third cost function was [9]

$$C_3 = \int_{V_{min}}^{V_{max}} ||F_d(V, \gamma)||^2 dV, \ where\ T_z(V, \gamma) = 0. \tag{4.6}$$

These three cost functions combined into a weighted cost function that greatly prioritizes the pitch stability [9]

$$C(\mathcal{F}) = 100C_1 + C_2 + C_3, \tag{4.7}$$

where $\mathcal{F} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ is the nonlinear function "which maps the glider's speed V and angle of attack γ to (F_d, F_l, T_z)" [9].

4.2.3 Integration

Due to the complexity of the cost functions, the integrations were performed numerically. Each integral was approximated with the formula

$$\int_{V_{min}}^{V_{max}} f(V) dV = \Delta V \sum_{i=1}^{N-1} f(i), \tag{4.8}$$

where $f(V)$ represents one of three functions that are integrated to calculate the total cost, N is the number of numerical points used to represent $f(V)$, and ΔV is the velocity difference

between two points. Each function (torques, forces, etc.) were calculated numerically using the equations above (3.1 - 3.3).

4.2.4 Computational Methods for the Second and Third Cost Functions

A major complication in calculating the second and third cost functions is the line integral. The line is the combination velocities and angles-of-attack such that the pitching torque is zero. This can be found in simulation by finding a symbolic equation for the torque as a function of the speed and angle-of-attack. However, this is difficult because the pitching torque is a function of the coefficients of drag and lift, which are referenced with polars. Obtaining a symbolic equation for torque requires approximating the polars as functions of the angle-of-attack. This was done using MATLAB's polyfit function to fit high-order polynomials to the polars in the region of angles-of-attack that the glider is expected to have (the region of speeds and angles-of-attack that the glider was expected to be in is called the Ω region in [9]). The polynomial fits are shown in Figures 4.3 and 4.4, with maximum errors of 15% for the drag polynomial and 9.3% for the lift polynomial, both happening around 12 degrees. Once the angles-of-attack that make the pitching torque zero were found, the drag and lift forces were calculated using the methods in section 3.2 found for various speeds within the Ω region to be used in the integrals for C_2 and C_3. To increase the speed of the optimization, the forces and torques were calculated using this method for only whole-number speeds, and then the forces and torques for other speeds were found using linear interpolation [9].

4.2.5 Testing for Non-Convexity

Finding the minimum of a function is much simpler when the function is convex, because, within the region in which it is defined, it has one minimum [16]. If a function is not-convex, optimization methods run the risk of finding a local minimum [17] or a saddle point in high-dimensional problems [18], rather than the global minimum, which, for the cost function, represents the optimal configuration for the glider. A function, f, is convex when [16]

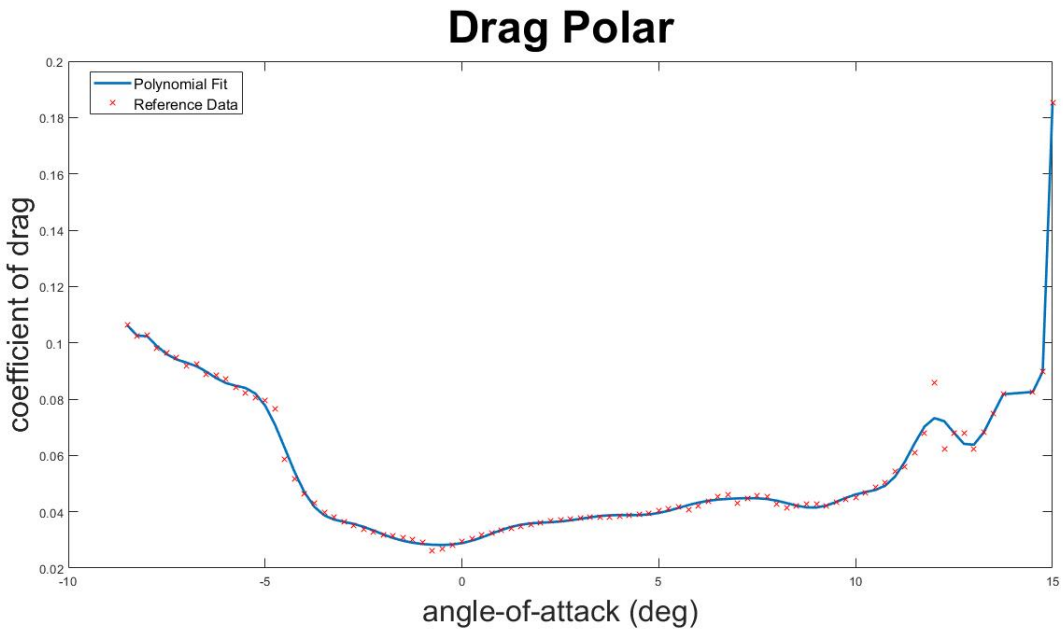

Figure 4.3: Polynomial approximation of a Clark-Y drag-coefficient polar

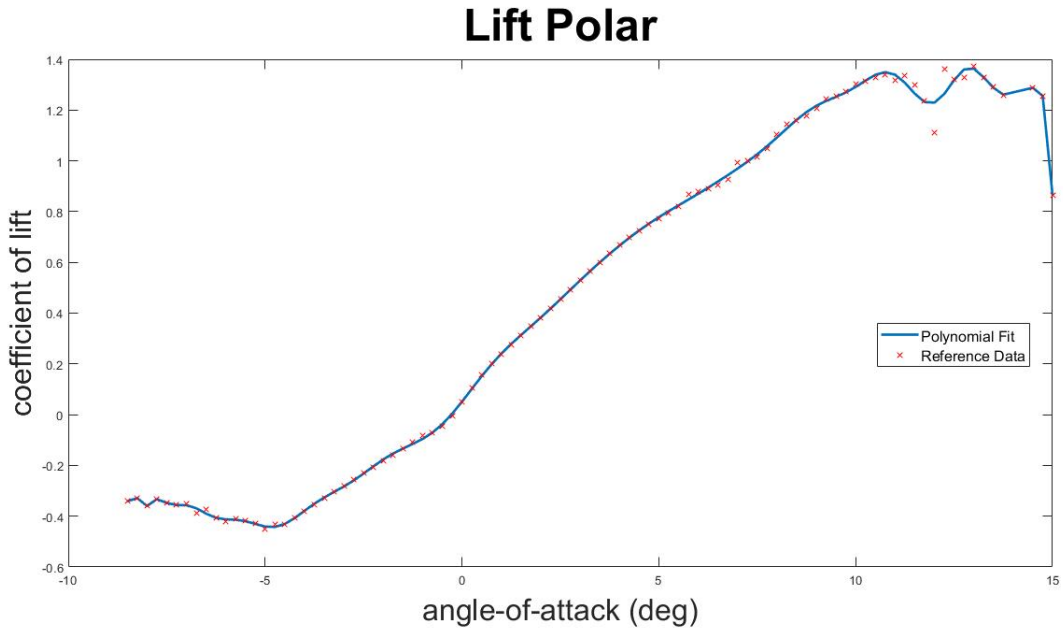

Figure 4.4: Polynomial approximation of a Clark-Y lift-coefficient polar

$$\forall x_1, x_2 \in X, \forall t \in [0,1] : \ f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2). \tag{4.9}$$

An intuitive example of this property is the function, $f(x) = x^2$, which was used to test the gradient descent method in Figure 4.2. Non-convex functions do not follow the conditions set in equation (4.9), meaning that they have more than one minimum in the region in which they are defined.

The cost function's convexity was tested using a simple MATLAB script. The script picked two random points, x_1, x_2, that consist of randomly selected glider parameters, such as wing span and elevator mounting angle, and a $t \in [0, 1]$ and tests the convexity condition. If the condition was satisfied, the script started the process over, otherwise it output a variable that indicated that the function is not convex. Putting the cost function in this script quickly revealed that it is not convex, meaning that there is more than one minimum within the region in which it is defined [16].

4.2.6 Accuracy Test

To verify the optimization's ability to find optimal configurations, it was tested on the glider used to obtain the results displayed in Figure 3.5. The goal was to prove that the optimization could improve the performance of that glider, before it was used to find the optimal configuration of the jump-glider.

Note that the aspect ratios of the wing and elevator that the glider used by the University of Tokyo had were preserved during the optimization, rather than giving them both an aspect ratio of 6. This decision was made to allow the original configuration of the glider to be a possible solution for the optimization.

4.2.7 Results

Figure 4.5 shows the results of applying the optimization to the test glider. The simulated trajectory from the physics model is compared to the simulated trajectory after the optimization, both trajectories being run for the same amount of time. As can be seen, the optimized glider falls short in distance and height when compared to the non-optimized glider.

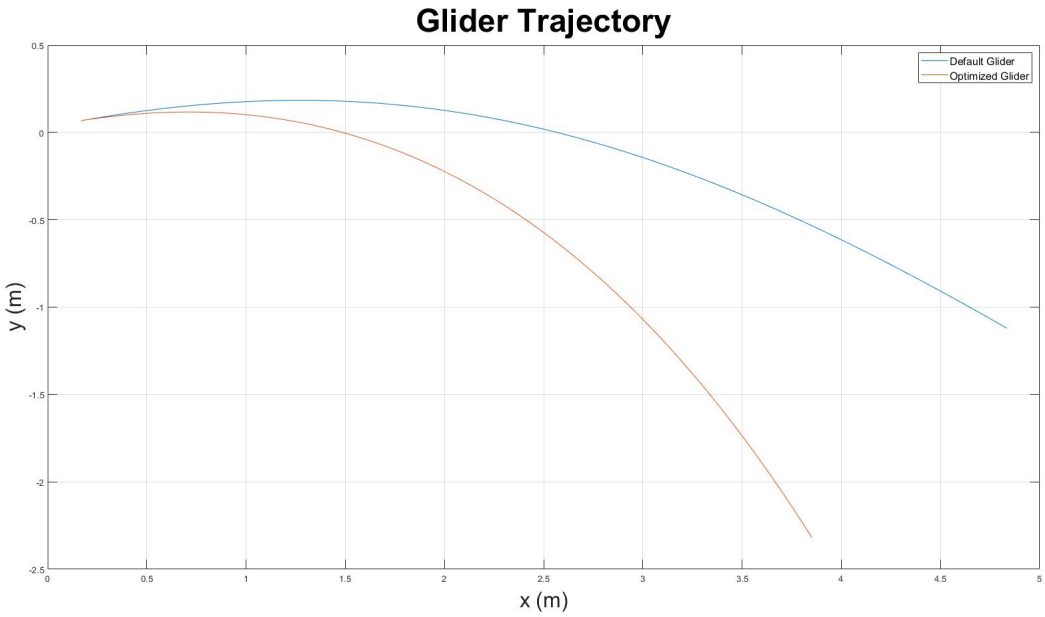

Figure 4.5: Attempt to optimize one of the gliders in [9]

Figure 4.6 shows the optimization of the cost function for one parameter, the wing's mounting angle. As can be seen in the Figure, the optimization does not seem to find a minimum for the function.

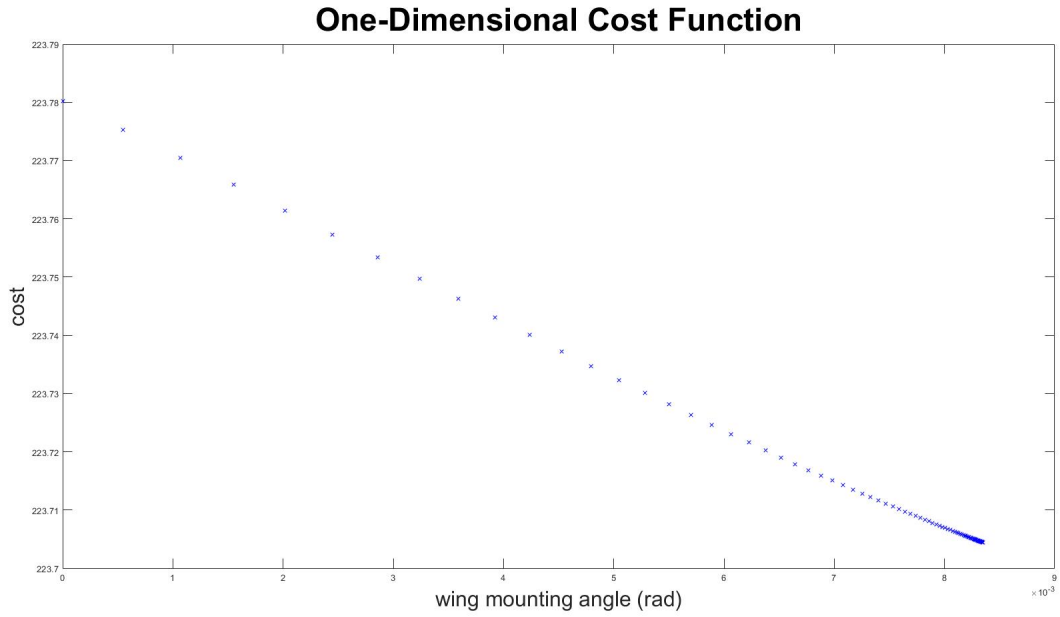

Figure 4.6: One dimensional cost function

4.2.8 Discussion

The cost function only changes by a small amount, despite the wing's mounting angle having a range of about 3 degrees during the optimization. But, when the angle-of-attack is varied by about 3 degrees, the coefficient of lift (shown in Figure 4.4) varies considerably, which should affect the cost function. However, Figure 4.7 shows that the cost function precipitously drops as the wing's mounting angle increases beyond the range that was tested in Figure 4.6. The drop is much more significant than how much it should vary for differing coefficients of drag and lift. Both of these anomalies suggest that the cost function is being calculated incorrectly.

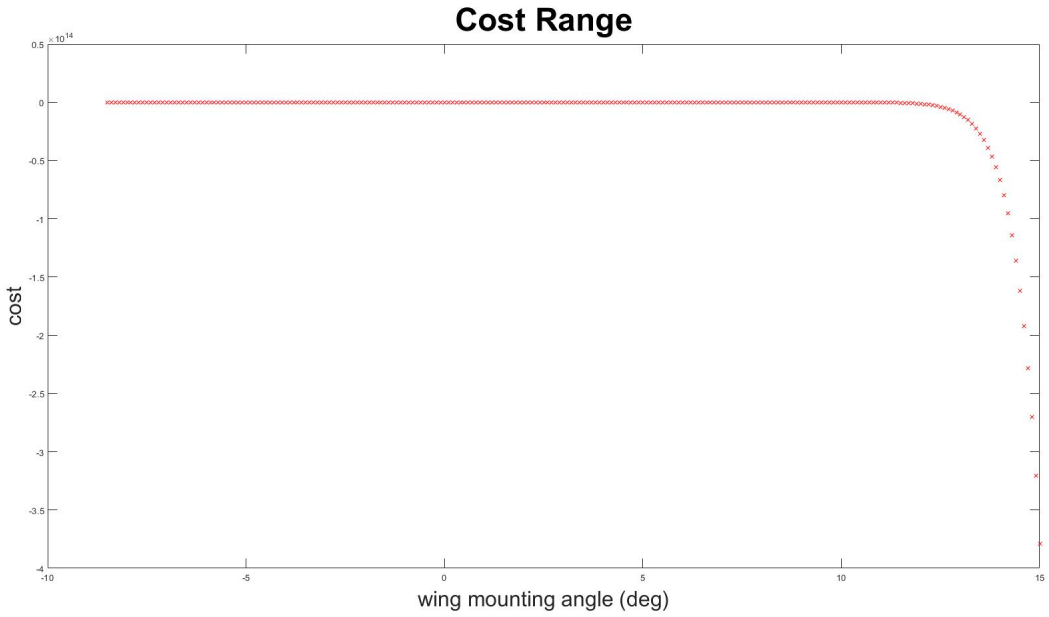

Figure 4.7: Full Range of Costs

Chapter 5

Future Work

Once the cost function is fixed, the next step is to find the global minimum of the cost function in its defined domain. This step will depend on problems that the optimization may run into. If the optimization runs into saddle points, a method such as the one presented in [18] may be applicable. If the optimization is just running into local minima, a simple gradient descent method will likely only be able to find a local minimum, and there are too many parameters to be able to easily find the global minimum with a lucky choice of initial conditions [17]. Luckily, there are ways to tailor the Gradient Descent Method to find the global minimum [19] of a function.

In terms of time efficiency, the Gradient Descent Method has been known to perform slowly (as in, needing a large number of steps) [17], but there are other optimization methods that are quicker. [20] from the University of California, San Diego, has developed an efficient optimization method that can be applied to both convex and non-convex functions (however, it should be noted that fixing my optimization may reveal that the cost function is actually convex). This method would reduce the amount of times that the cost function needs to be calculated, saving computation time.

It may also be worthwhile to consider the cost of the length of the fuselage of the glider against the extra pitching torque produced by placing an elevator further back behind the center-of-mass of the glider. However, because the pitching torque needs to be balanced, rather than

maximized, placing the elevator further back on the glider may require it to be smaller to keep that balance, negating any advantage of lengthening the fuselage.

Another optimization parameter is the shapes of the wing and the elevator. They were made rectangular due to the ease of manufacturing rectangular wings and the simplification of the integration of the wing elements used in 3.5, but, because the Reynolds Number is higher than 50,000, the optimal shape of the wing and the elevator may not be rectangular, and the optimal aspect ratio may not be 6.

The largest gap in the model of this glider is the assumption that lateral forces will not affect the trajectory. As mentioned above, various gliders that were built turned too much to glide forward a great distance. It could be worthwhile to account for lateral forces in order to correctly size a rudder.

Finally, a stability analysis could be useful to aid in the glider's design. It has not been tested how the glider responds to disturbances, which could be great depending on the intended application(s) of the jump-glider.

Chapter 6

Conclusion

This thesis has presented the framework of a tool that can be used to design the gliding components of jump-gliders. Once the problems with the optimization tool have been resolved, it can be applied to the jump-glider designed in Chapter 2, other jump-gliders, or even gliders in general. It will ensure that the design of the jump-gliders will not be impeded by a large number of tests that tediously tweak the parameters that the optimization tool can instead tune.

Bibliography

[1] A. L. Desbiens, M. T. Pope, D. L. Christensen, E. W. Hawkes, and M. R. Cutkosky, "Design principles for efficient, repeated jumpgliding", *Bioinspiration & Biomimetics*, vol. 9, no. 2, p. 025 009, May 2014. DOI: 10.1088/1748-3182/9/2/025009. [Online]. Available: https://doi.org/10.1088%2F1748-3182%2F9%2F2%2F025009.

[2] A. Vidyasagar, J.-C. Zufferey, D. Floreano, and M. Kovač, "Performance analysis of jump-gliding locomotion for miniature robotics", *Bioinspiration & Biomimetics*, vol. 10, no. 2, p. 025 006, Mar. 2015. DOI: 10.1088/1748-3190/10/2/025006. [Online]. Available: https://doi.org/10.1088%2F1748-3190%2F10%2F2%2F025006.

[3] A. L. Desbiens, M. Pope, F. Berg, Z. E. Teoh, J. Lee, and M. Cutkosky, "Efficient jumpgliding: Theory and design considerations", in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 4451–4458. DOI: 10.1109/ICRA.2013.6631209.

[4] P. Pritchard and J. Mitchell, *Fox and McDonald's Introduction to Fluid Mechanics, 9th Edition*. Wiley, 2015, ISBN: 9781119034582. [Online]. Available: https://books.google.com/books?id=vXu5BwAAQBAJ.

[5] F. A. Administration, *Pilot's Handbook of Aeronautical Knowledge*. United States Department of Transportation, 2016, ISBN: 978161954734. [Online]. Available: https://

www.faa.gov/regulations_policies/handbooks_manuals/aviation/phak/.

[6] E. Toolbox. (2019). Air - density, specific weight and thermal expansion coefficient at varying temperature and constant pressures, [Online]. Available: https://www.engineeringtoolbox.com/air-density-specific-weight-d_600.html (visited on 04/30/2019).

[7] ——, (2019). Air - dynamic and kinematic viscosity, [Online]. Available: https://www.engineeringtoolbox.com/air-absolute-kinematic-viscosity-d_601.html (visited on 04/30/2019).

[8] J. D. Anderson, *Aircraft performance and design*. McGraw-Hill Science/Engineering/Math, 1999.

[9] N. Umetani, Y. Koyama, R. Schmidt, and T. Igarashi, "Pteromys: Interactive design and optimization of free-formed free-flight model airplanes", *ACM Trans. Graph.*, vol. 33, no. 4, 65:1–65:10, Jul. 2014, ISSN: 0730-0301. DOI: 10.1145/2601097.2601129. [Online]. Available: http://doi.acm.org/10.1145/2601097.2601129.

[10] *Aerodynamic center - ac*, https://www.grc.nasa.gov/WWW/K-12/airplane/ac.html, Accessed: 2019-05-20.

[11] L. J. Clancy, *Aerodynamics*, eng. New York: Wiley, 1975, ISBN: 0470158379.

[12] A. Tools. (2019). Lift drag polars, [Online]. Available: http://airfoiltools.com/airfoil/details?r=polar/index/#xfoil (visited on 02/15/2019).

[13] E. Hairer, C. Lubich, and G. Wanner, "Geometric numerical integration illustrated by the störmer–verlet method", *Acta Numerica*, vol. 12, p. 400, Jul. 2003. [Online]. Available: https://doi.org/10.1017/S0962492902000144.

[14] L. C. W. Dixon and D. J. Mills, "Effect of rounding errors on the variable metric method", *Journal of Optimization Theory and Applications*, vol. 80, no. 1, pp. 175–

179, Jan. 1994, ISSN: 1573-2878. DOI: 10.1007/BF02196600. [Online]. Available: https://doi.org/10.1007/BF02196600.

[15] A. Cauchy, "Méthode générale pour la résolution des systemes d'équations simultanées", *Comp. Rend. Sci. Paris*, vol. 25, no. 1847, pp. 536–538, 1847.

[16] J. Borwein and A. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*, ser. CMS Books in Mathematics. Springer New York, 2010, ISBN: 9780387312569. [Online]. Available: https://books.google.com/books?id=eeL%5C_Cdmowv8C.

[17] S. Ruder, "An overview of gradient descent optimization algorithms", *arXiv preprint arXiv:1609.04747*, 2016.

[18] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization", in *Advances in neural information processing systems*, 2014, pp. 2933–2941.

[19] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient descent only converges to minimizers", in *Conference on learning theory*, 2016, pp. 1246–1257.

[20] P. Beyhaghi, "Delaunay-based global optimization algorithms and their applications", PhD thesis, UC San Diego, 2016.