Assignment 1: Optimal flight with a glider

Johannes Mäkinen, Viljami Uusihärkälä
Aalto University School of Science, Department of Mathematics and Systems Analysis
{*johannes.makinen, viljami.uusiharkala*}*@aalto.fi*

1 Introduction

Finding the optimal trajectory of a gliding flight with no additional thrust seems like a simple task. However, when trying to model the physical aspects of the flight, using flight mechanics, dynamic optimization and simulation, we can see that it becomes quite a complex system. Fortunately, it is known that the dynamics of rotation are significantly more rapid than those of movement, we can simplify the aircraft as a point mass and ignore the rotations of the glider. The forces that affect the glider, the dynamics of the glider and the surrounding environment can all be taken into account by starting from the basic free body diagrams of a moving body in the air. By creating the state equations of such a point mass body, we can simulate and more importantly determine the optimal controls in different scenarios. The scenarios that we are going to go over in this project are gliding flight with and without thermal upward air flow.

Our goal in this project is to simulate and optimize the flight of a glider using dynamic optimization methods. The objective of the simulation is to learn how the mechanics and controls can be used to vary the state of the glider in the air. Afterward, the objective of the optimization is to find a flight path, e.g. the controls that produce the maximal distance covered by the glider in the air. Matlab will be used for both the simulation and the optimization.

2 Modeling for optimization

First, we have to formulate the state equations of the point mass glider in the air. Figure 1 shows the fundamental forces that affect the glider during flight.

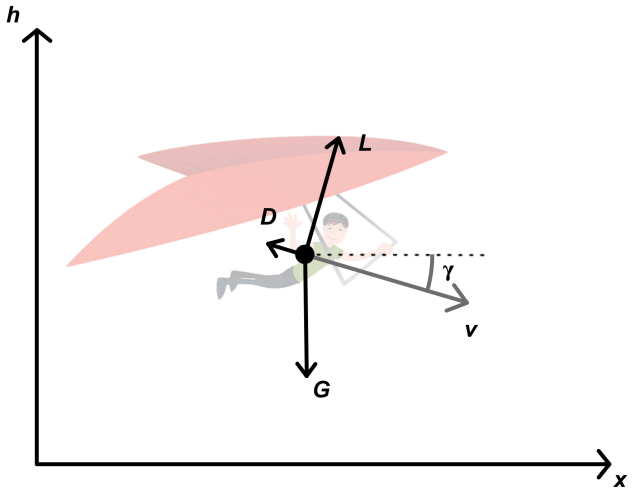

Figure 1: Free body diagram of forces acting upon the glider.

In Figure 1, L, D, G, v and γ are used to describe the lift, drag, gravitational force, velocity and the flight path angle, respectively. Lift force L can be defined as $L(\alpha, h, v) = C_L(\alpha)Sg(h, v)$, where $C_L(\alpha)$ is the lift coefficient which usually depends on the angle of attack α. Though, at small velocities (under 200m/s) and angles close to 0, we can approximate the lift coefficient as $C_L(\alpha) = C_{L_\alpha}\alpha$. From now on, we will be denoting the lift coefficient as C_L which is used as the control variable. S is the reference area (usually the wing area) of the glider and the kinetic energy density is $g(h, v) = 0.5\rho(h)v^2$. As the changes in altitude h of the glider are small, we will assume a constant air density ($\rho(h) =$ constant).

The drag force can be defined $D = (C_{D_0} + KC_L^2)Sq(h, v)$, where $C_{D_0} + KC_L^2$ is the total drag coefficient. C_{D_0} is the zero-drag coefficient, which is caused by the geometry of the aircraft and the flow of air around it. $KC_L^2Sq(h, v)$ is an approximation of the induced drag, which is caused by a portion of the aircraft's kinetic energy being used to maintain the whirls on the tip of the wings due to pressure differences. Both C_{D_0} and K are estimated based on wind tunnel or flights tests.

$G = mg$ is the gravitational force that pulls the glider towards the center of Earth. Velocity v differs from the horizontal plane by the flight path angle γ. Table shows the given estimated values for each parameter, and the limit for the control C_L.

Parameter	Symbol	value
Combined mass of the glider and the pilot	m	100 kg
Reference area of the glider	S	14 m^2
Zero drag coefficient of the glider	C_{D_0}	0.034
Induced drag coefficient of the glider	K	0.07
Gravitational acceleration	g	9.81 kgm/s^2
Air density	ρ	1.13 kg/m^3
Allowed lift coefficients	C_L	$[-1.4, 1.4]$

Table 1: Estimated values of parameters and controls.

Now, using the fundamental forces shown in Figure 1, we can derive the state equations of the model. The changes in position x and altitude h can be seen as time derivatives of those parameters. The two other equations we solve with Newton's 2nd law $\sum F = ma$, where the forces along the x-axis are taken as is. We also want to get a formulation for the state variable γ, and thus we take into account the forces perpendicular of the velocity, and determine the sum of the forces to create the centripetal acceleration $a_{cm} = \frac{v^2}{r} = v\dot{\gamma}$.

$$\begin{cases} v_x = \frac{\Delta x}{\Delta t} \implies \dot{x} = v_x \\ v_y = \frac{\Delta h}{\Delta t} \implies \dot{h} = v_y \\ \sum F_x = ma_x = m\dot{v} \\ \sum F_y = ma_{cm} = m\frac{v^2}{r} \end{cases} \implies \begin{cases} \dot{x} = v\cos(\gamma) \\ \dot{h} = v\sin(\gamma) \\ -D - G\sin(\gamma) = m\dot{v} \\ L - G\cos(\gamma) = mv\dot{\gamma} \end{cases} \tag{1}$$

We can present the formulation of 1 in a form that i usually used in dynamic optimization problems.

$$\begin{cases} \dot{x} = v\cos(\gamma) \\ \dot{h} = v\sin(\gamma) \\ \dot{v} = \frac{-D-G\sin(\gamma)}{m} \\ \dot{\gamma} = \frac{L-G\cos(\gamma)}{mv} \end{cases} \tag{2}$$

2.1 Simulation of the glider

The state Equations 2 derived in the previous section can be used to simulate the behaviour of the glider. The simulations were performed using Matlab's numerical differential equation solver in order to analyze the model with different parameters and initial values. In the simulations, the glider was given some initial velocity and height from which the simulation was calculated using a predefined simulation period of 10 seconds. Thus it can be also examined how far the glider flew in the fixed time period with the given parameter values. Furthermore, the lift coefficient C_L was kept constant throughout the simulations at $C_L = 0.3$ to provide some upward lift.

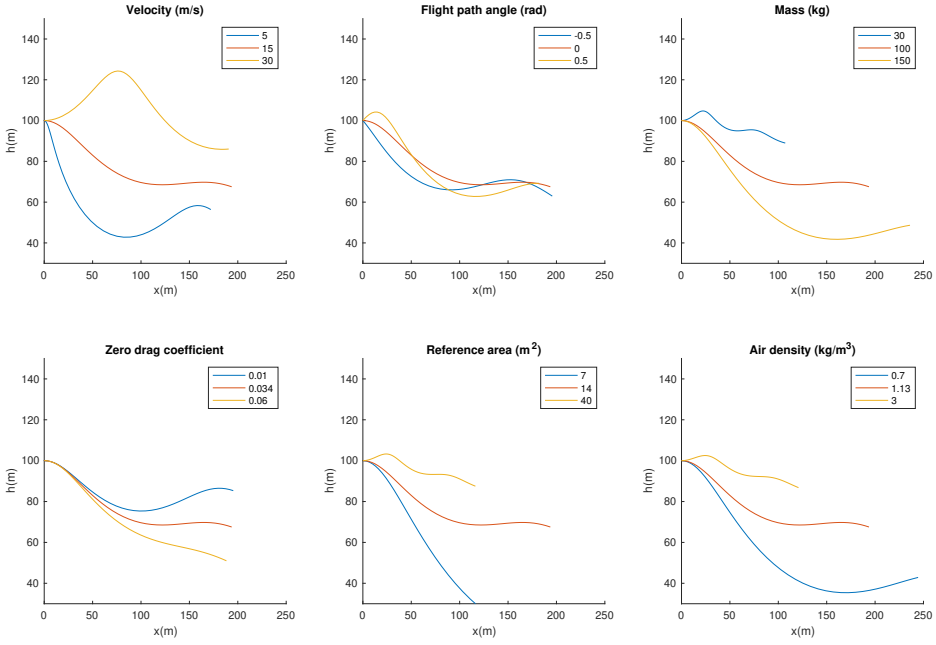

Figure 2: Glider's trajectories with varied parameters.

The simulated trajectories are gathered in Figure 2. A total of six different initial values or parameters were varied independently of each other. From the figure it is obvious that the selected variables have a significant impact on the shape and length of the trajectories. For example, the initial velocity determines whether the glider will gain height or start to descent immediately. With the initial velocity $v_0 = 30$ m/s, the glider climbs over 20 meters before starting to descend. Also, the initial flight path angle greatly affects the initial behaviour of the glider by sharing similar characteristics as with the velocity simulation.

According to Figure 2 mass and air density play important roles when it comes to the speed of the glider. As can be seen from the figure, increasing the mass will increase the speed of the glider, which is why the glider can travel a greater distance in the fixed 10-second time frame. Also, decreasing the air density will result in a faster glider. Though, it can be also seen that the decreased air density will greatly decrease the lift force and causes the glider to descent rapidly. When the air density is increased, however, the glider will gain more lift, and the trajectory will be upwards in the beginning.

Finally, Figure 2 suggests that the features of the glider are also of great importance when it comes to the properties of the trajectory. That is, the zero drag coefficient C_{D_0} and the reference area S have a considerable impact on the lift. The figure agrees with the intuitive fact that giving the glider wider wings, will result in more lift and slower descent. This behaviour is obvious in Figure 2, where the glider with reference area $S = 40$ m^2 hardly descents in the simulation, whereas the glider with reference area 7 m^2 starts to rocket down immediately after the simulation has begun. The zero drag coefficient simulation also shows that the natural drag of the glider is correlated with the lift. Particularly, increasing the zero drag coefficient will decrease the overall lift. The

observations above suggest that the state equations are feasible and the model could be used in the optimization problem for determining the optimal lift controls.

2.2 Stalling speed

Using the state Equations at 2, we can compute the stalling speed when the glider is moving at a constant velocity and flight path angle ($\gamma \approx 0 \implies \sin(\gamma) \approx \gamma$ & $\cos(\gamma) = 1$). The stalling speed is the speed at which the greatest possible lift becomes less than the gravitational force. Thus, we try to find a speed at which $\max(L) < G$

$$L < G \tag{3}$$
$$\implies C_L S \cdot 0.5\rho v^2 < mg \tag{4}$$
$$\implies v < \sqrt{\frac{2mg}{C_L S \rho}} \tag{5}$$

The maximum lift at any speed is achieved when C_L is at the (positive) extreme point of its possible values. Therefore, we can solve Equation 5 by using $C_L = 1.4$ and the other parameter values from Table 1.

$$v < \sqrt{\frac{2mg}{C_L S \rho}} \tag{6}$$
$$\implies v < \sqrt{\frac{2 \cdot 100\text{kg} \cdot 9.81\text{m/s}^2}{1.4 \cdot 14\text{m}^2 \cdot 1.13\text{kg/m}^3}} \tag{7}$$
$$\implies v < 9.412\text{m/s} \tag{8}$$

Thus, the stalling speed of the glider is ~ 9.4 m/s. In practice, the stalling speed is different for each aircraft and is highly affected by the geometry of the wings. Next, we'll simulate the glider in such conditions that it should stall mid-air. That is, we'll simulate the behaviour of the glider with the maximum allowed lift coefficient $C_L = 1.4$ and the theoretical stalling speed just calculated. Furthermore, we will investigate the flight in horizontal initial flight ($\gamma_0 = 0$).

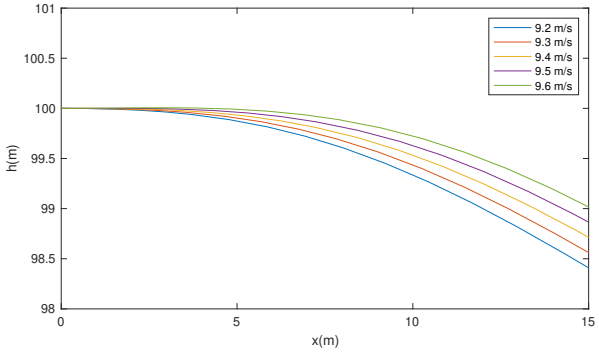

Figure 3: Glider's trajectory in the neighbourhood of the stalling speed.

Figure 3 shows the behaviour of our model in the neighbourhood of the theoretical stalling speed. According to the figure, there are no observable changes in the trajectory when the stalling speed is given to the aircraft. Contrarily, the behaviour is similar with initial velocities above and below the stalling speed. That is, the model described by the state equations in 2 is not valid with speeds less than the stalling speed (9.4 m/s), but the model behaves as if there was not anything special about the stalling speed, which is very significant in practice.

3 Optimization

3.1 Static time-invariant scenario

The objective of the flight is to glide as far along the x-axis as possible for each unit of lost altitude. We can now formulate a static, time-invariant optimization problem and try to solve the optimal ratio. We'll assume that the flight path angle is near 0 ($\sin\gamma \approx \gamma, \cos\gamma \approx 1$), and that the velocity and the flight path angle are constant ($\dot{v} = 0, \dot{\gamma} = 0$). We can simplify the Equations 2 to comply with our assumptions.

$$\begin{cases} \dot{x} = v \\ \dot{h} = v\gamma \\ 0 = \frac{-D-G\gamma}{m} \\ 0 = \frac{L-G}{mv} \end{cases} \implies \begin{cases} \dot{x} = v \\ \dot{h} = v\gamma \\ (C_{D_0} + KC_L^2)0.5S\rho v^2 = -mg\gamma \\ C_L 0.5S\rho v^2 = mg \end{cases} \tag{9}$$

Now, we want to maximize the ratio between distance traveled along the x-axis and the lost altitude $\frac{\Delta x}{-\Delta h}$. This can be further formulated as

$$\frac{\Delta x}{-\Delta h}\frac{\Delta t}{\Delta t} = \frac{\dot{x}}{-\dot{h}}, \tag{10}$$

and using first two equalities of the simplified Equations 9 we get

$$\frac{\dot{x}}{-\dot{h}} = \frac{v}{-v\gamma} = -\gamma^{-1}. \tag{11}$$

Using the two latter equalities of the Equations 9 to find another formula of γ.

$$\frac{(C_{D_0} + KC_L^2)0.5S\rho v^2}{C_L 0.5S\rho v^2} = \frac{-mg\gamma}{mg} = -\gamma \tag{12}$$

Combining our intermediate results we can provide a single expression that we want to maximize. From Equations 10 and 11 we see that we want to maximise $-\gamma^{-1}$, which can also be stated as minimising $-\gamma$. Simplifying Equation 12, we can see that we want to minimize the expression

$$\frac{C_{D_0} + KC_L^2}{C_L}. \tag{13}$$

Expression 13 is a simple polynomial which has its minimum value when its derivative is zero, on the possible interval $C_L \in [-1.4, 1.4]$.

$$\frac{d}{dC_L}\left(\frac{C_{D_0} + KC_L^2}{C_L}\right) = K - \frac{C_{D_0}}{C_L^2} = 0 \implies C_L^* = \pm\sqrt{\frac{C_{D_0}}{K}} \approx \pm 0.6969 \tag{14}$$

The optimal control $C_L = 0.6969$ gives the maximum distance for each unit of altitude lost in our static, time-invariant scenario. The same result could have been concluded by analyzing Figure 4, which shows the (C_D, C_L) -curve and its tangent passing through the origin. The tangent that passes through the origin has a contact point with the curve when $C_L = \pm\sqrt{\frac{C_{D_0}}{K}} \approx \pm 0.6969$

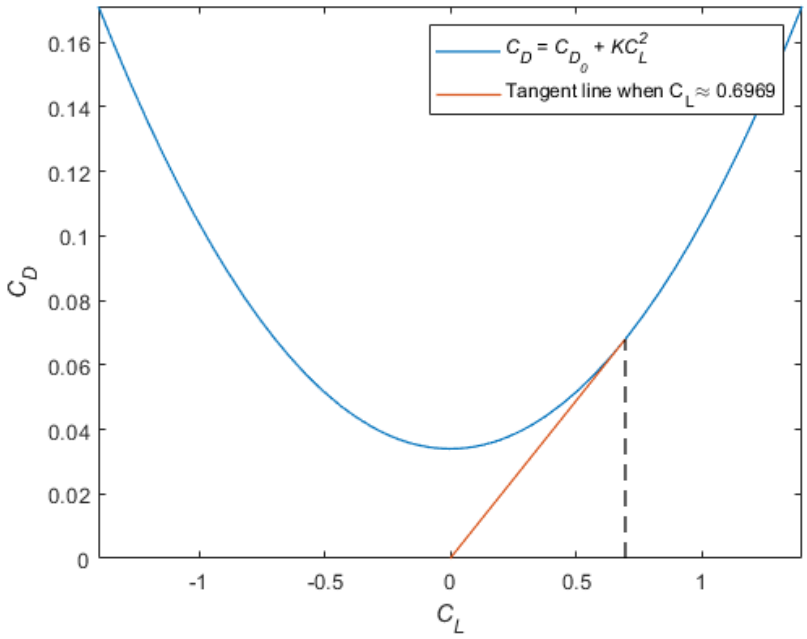

Figure 4: (C_D,C_L)-curve.

The resulting optimal static control derived in Equation 14 can be now used in the simulation used earlier to compare the results of the simulation and the static optimal control approximation. This comparison is conducted by comparing the distances traveled in each approach.

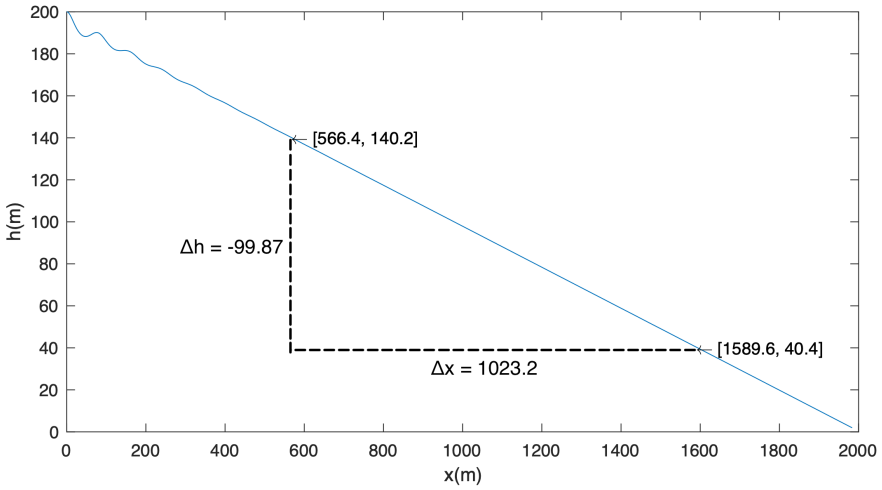

Figure 5: Simulation of static flight.

Figure 5 shows the simulated path of the glider after the flight path has settled to stable descent using the static optimal control $C_L^* \approx 0.6969$. It can be read from the figure, that as the glider descents $\Delta h = -99.87$ meters, the glider will travel $\Delta x = 1023.2$ meters horizontally. Using the same horizontal drop Δh, we can calculate the distance travelled suggested by the static optimization model. Using Equations 10 and 12 we get that

$$\Delta x_s = -\frac{C_L^* \Delta h}{C_{D_0} + K{C_L^*}^2} = -\frac{0.6969 \cdot (-99.87\text{m})}{0.034 + 0.07 \cdot 0.6969^2} \approx 1023.5\text{m} \tag{15}$$

That is, both approaches yielded nearly identical results. This is expected as the flight path angle is relatively small in the stable flight ($\gamma \approx$ -0.097) and thus it can be concluded that the sine and cosine approximations do not result in an excessive amount of error.

3.2 Formulating the dynamic optimization problem

The dynamic optimization of the glider can be done using basic Dynamic optimization methods. We assume free final time such that the objective functional is to maximize the x-coordinate at the end of the flight. We need to fulfill the state equations and the control restrictions. The altitude at the end is $h(t_f) = h_f$ and the velocity at the end is $v(t_f) = v_f$.

As our objective is to maximise the distance traveled in the x-direction max. $x(t_f) = x_f$, but with the first equality in Equations 2 we can rewrite it as max. $\int_0^{t_f} v\cos(\gamma)dt$. Now, collecting our objective functional, the state equations in 2 and the initial and end values we can formulate the whole dynamic optimization problem.

$$\text{max.} \quad \int_0^{t_f} v\cos(\gamma)dt \tag{16}$$

$$
\begin{aligned}
\text{s.t.} \quad & \dot{x} = v\cos(\gamma) && (17)\\
& \dot{h} = v\sin(\gamma) && (18)\\
& \dot{v} = \frac{-D - G\sin(\gamma)}{m} && (19)\\
& \dot{\gamma} = \frac{L - G\cos(\gamma)}{mv} && (20)\\
& C_L \in [-1.4, 1.4] && (21)\\
& h(t_f) = h_f && (22)\\
& v(t_f) = v_f && (23)\\
& x(0) = 0 && (24)\\
& h(0) = h_0 && (25)\\
& v(0) = v_0 && (26)\\
& \gamma(0) = \gamma_0, && (27)
\end{aligned}
$$

where $L(h, v) = C_L 0.5 S\rho v^2$ and $D = (C_{D_0} + KC_L^2)0.5S\rho v^2$.

Using the problem stated in 16-27, we form the Hamiltonian function $H(X, p, C_L)$, where the states are $X = [x, h, v, \gamma]$ and the costates are $p = [p_x, p_h, p_v, p_\gamma]$.

$$
\begin{aligned}
H(X, p, C_L) &= v\cos(\gamma) + \dot{x}p_x + \dot{h}p_h + \dot{v}p_v + \dot{\gamma}p_\gamma && (28)\\
H(X, p, C_L) &= v\cos(\gamma) + v\cos(\gamma)p_x + v\sin(\gamma)p_h \\
&\quad + \frac{-D - G\sin(\gamma)}{m}p_v + \frac{L - G\cos(\gamma)}{mv}p_\gamma && (29)
\end{aligned}
$$

The extreme value of C_L can be formulated as a function of the states and costates. This is done by taking the derivative of the Hamiltonian with respect to the control and finding its zero points.

$$
\begin{aligned}
& \frac{\partial}{\partial C_L} H(X, p, C_L) && = 0 && (30)\\
\Rightarrow & \frac{\partial}{\partial C_L}\left(-\frac{D}{m}p_v + \frac{L}{mv}p_\gamma\right) && = 0 && (31)\\
\Rightarrow & \frac{\partial}{\partial C_L}\left(-\frac{p_v}{m}(D_0 + KC_L^2)\cdot 0.5 \cdot S\rho v^2 + \frac{p_v}{mv}C_L \cdot 0.5 \cdot S\rho v^2\right) && = 0 && (32)\\
\Rightarrow & -p_v K C_L S\rho v^2 + 0.5 \cdot S\rho v p_\gamma && = 0 && (33)\\
\Rightarrow & C_L = \frac{p_\gamma}{2p_v K v} && && (34)
\end{aligned}
$$

We also know that the control is bounded on the interval $|C_L| \leq 1.4$, and thus $|\frac{p_\gamma}{2p_v K v}| \leq 1.4$. Now, taking the second order derivative, we can inspect the extreme point even more.

$$
\frac{\partial^2}{\partial C_L^2} H(X, p, C_L) = -\frac{p_v K S\rho v^2}{m} \tag{35}
$$

The second-order derivative of the Hamiltonian with respect to the control C_L is always negative, and thus the extreme point in 34 is the maximum point.

Now we can form the costate equations

$$\begin{cases} -\dot{p}_x = H_x = 0 \\ -\dot{p}_h = H_h = 0 \\ -\dot{p}_v = H_v = \frac{p_\gamma \rho S C_L}{2m} - \frac{p_v \rho S (C_{D_0} + K C_L^2) v}{m} + \cos(\gamma) + p_x \cos(\gamma) + p_h \sin(\gamma) + \frac{p_\gamma g \cos(\gamma)}{v^2} \\ -\dot{p}_\gamma = H_\gamma = (p_h v - g p_v)\cos(\gamma) - v\sin(\gamma) - p_x v \sin(\gamma) + \frac{g p_\gamma \sin(\gamma)}{v} \end{cases} \tag{36}$$

Initial and terminal conditions for the states and costates can be derived from the general transversality condition:

$$[h_x(x^*(t_f), t_f) - p^*(t_f)]^T \delta x_f + [H(x^*(t_f), u^*(t_f), p^*(t_f), t_f) + h_t(x^*(t_f), t_f)]\delta t_f = 0 \tag{37}$$

$$x(0) = 0, \qquad x(t_f) = free \tag{38}$$
$$v(0) = v_0, \qquad v(t_f) = v_f \tag{39}$$
$$h(0) = h_0, \qquad h(t_f) = h_f \tag{40}$$
$$\gamma(0) = \gamma_0, \qquad \gamma(t_f) = free \tag{41}$$
$$p_x(t_f) = 0 \tag{42}$$
$$p_\gamma(t_f) = 0 \tag{43}$$

The transversality conditions 37 imply the boundary conditions 42 & 43 on the costate variables which do not have an endpoint constraint of the state variables. The rest of the initial and terminal conditions are specified as given except for the final distance and the flight path angle at the end of the flight.

The boundary and transversality conditions derived above could be used to determine the free end time explicitly, though, as a function of the control C_L, since the end time is heavily affected by the control. One would need to first solve the expressions for the states via solving the system of differential equations in 2 and then solving the unknown integration constants, together with the final time t_f, using the boundary and transversality conditions in 38 through 43. In this report, however, the end time is not solved as described above, but the problem is solved numerically using two different approaches which will be presented in the following section.

Finally, before moving on to solving the control problem, we will check the existence of singular control solutions. We start by examining the Hamiltonian 29 in order to investigate whether there are circumstances under which the minimum principle fails to provide sufficient information about the relationship between the optimal states X^*, costates p^* and the control C_L^*. Consider the minimum principle

$$H(X^*, p^*, C_L^*) \leq H(X^*, p^*, C_L) \tag{44}$$

$$\Rightarrow -D^* p_v^* + L^* \frac{p_\gamma^*}{v^*} \leq -D p_v^* + L \frac{p_\gamma^*}{v^*} \tag{45}$$

$$\Rightarrow -(C_{D_0} + K C_L^{*^2}) v^* p_v^* + C_L^* p_\gamma^* \leq -(C_{D_0} + K C_L^2) v^* p_v^* + C_L p_\gamma^* \tag{46}$$

$$\Rightarrow C_L^*(p_\gamma^* - K C_L^* v^* p_v^*) \leq C_L(p_\gamma^* - K C_L v^* p_v^*) \tag{47}$$

From the above expression it can be concluded that the information of the relationships between the variables will be lost if $p_v^* = 0$ and $p_\gamma^* = 0$ for a finite interval $[t_1, t_2]$. Next we will determine whether this situation can occur. We substitute $p_v^* = 0$ and $p_\gamma^* = 0$ to the Hamiltonian in 29.

$$\begin{aligned} H(X, p, C_L) &= v\cos(\gamma) + v\cos(\gamma) p_x + v\sin(\gamma) p_h \\ &\quad + \frac{-D - G\sin(\gamma)}{m} \cdot 0 + \frac{L - G\cos(\gamma)}{mv} \cdot 0 \qquad (48) \\ &= v\cos(\gamma) + v\cos(\gamma) p_x + v\sin(\gamma) p_h \qquad (49) \end{aligned}$$

Furthermore, we can apply additional information about costate p_x. That is, according to Equation 36 the costate p_x is constant, since it's time derivative is zero. Moreover, the costate p_x has fixed final value of 0, as is stated in Equation 42. Combining this information, we can state that $p_x = 0$ at all times. Thus Hamiltonian becomes

$$H(X, p, C_L) = v\cos(\gamma) + v\sin(\gamma) p_h. \tag{50}$$

One of the necessary conditions states that Hamiltonian must be zero at all times. With the above form of Hamiltonian, it seems improbable that Hamiltonian would be zero at all values of γ and the constant p_h, which would contradict the condition of $p_v = 0$ and $p_\gamma = 0$ and would thus contradict the existence of singular intervals. We can confirm this result by setting $p_x = 0$, $p_v = 0$ and $p_\gamma = 0$ in Equation 36 and noting that derivatives of the costates become zero as well. We then have

$$\begin{cases} -\dot{p}_v(p_x, p_v, p_\gamma = 0) = \cos(\gamma) + p_h \sin(\gamma) = 0 \\ -\dot{p}_\gamma(p_x, p_v, p_\gamma = 0) = p_h v\cos(\gamma) - v\sin(\gamma) = 0 \end{cases} \tag{51}$$

$$\Rightarrow -p_h^2 \sin(\gamma) - \sin(\gamma) = 0 \tag{52}$$

$$\Rightarrow -p_h^2 = 1. \tag{53}$$

That is, by setting $p_x = 0$, $p_v = 0$ and $p_\gamma = 0$, we get a non-real solution for p_h, which again contradicts the existence of singular intervals. Now that we have shown that there cannot exist singular control solutions we can proceed to implementing the optimization model and solving the optimal control problem in the following sections.

3.3 Direct and indirect solution methods

There are two main categories of solution methods that are used when solving dynamic optimization problems numerically. First, the direct method is a way of producing the optimal solutions by not solving the necessary optimality conditions, but rather discretizing the dynamic optimization problem in time and solving the resulting nonlinear optimization problem with nonlinear programming. A direct method does not require an analytic expression for the necessary conditions, which can be a deciding factor when dealing with complex nonlinear dynamic equations.

The second way, the indirect solution method, relies on solving the necessary optimality conditions of the problem numerically. The necessary optimality conditions can be derived by setting the first variation of the Hamiltonian function to zero. The indirect approach usually requires the solution of a nonlinear multi-point boundary value problem. This is also a significant drawback of the method, as the derivation of analytic expressions for the necessary optimality conditions can be quite hard and time-consuming.

The indirect multiple shooting method is a powerful tool for solving trajectory optimization problems. The main point of the method is discretizing the control policy as a piecewise linear control, and then discretizing the states at the interval points of the piecewise control policy. We then integrate from the start of the interval to the end of the interval and enforce the condition that the states at the end of the interval are the initial states of the next interval. This way we avoid integrating the whole interval $t \in [0, t_f]$ at once, which could lead to unstable states and cumulative errors. Though, integrating between the smaller intervals can again be time-consuming as we are numerically integrating the whole interval, just in smaller steps. This can be alleviated with direct collocation.

Direct collocation is a nonlinearisation method in which we approximate the control policy as piecewise linear, and the states as third-degree piecewise polynomials. The key way of reducing the integration effort of indirect multiple shooting methods is the fact that we only enforce the constraints on the dynamics on intermediate points called collocation points. This implies that as we discretize the time interval to smaller intervals, we then enforce the constraints of the controls and states in between the intervals only in the middle point of the interval. This is much easier computationally as the integrations of the indirect multiple shooting method are converted into simpler expressions, which can be solved significantly faster. Though, with direct collocation, the number of decision variables increases, but in our problem, it shouldn't have a significant drawback.

3.4 Solving the optimization problem

3.4.1 Optimization method

Sequential quadratic programming (SQP) is an iterative method of solving a constrained optimization problem. Specifically, such problems where the objective and the constraints are twice continuously differentiable. The method works by solving quadratic model subject to constraints that are linearized. At each iteration, the subproblem is solved using common techniques, like the interior point method or augmenting the objective function with Lagrange multipliers. Iteratively solving the subproblems we finally find the optimum of the original problem. In our trajectory optimization, we will be using the augmented Lagrange method for solving the subproblems.

3.4.2 Implementation of SQP

Implementing the SQP method using Matlab can be done using the following scripts and functions. First, flight_main.m is the main script, which utilizes SQP and iteratively optimizes the glider's flight. We first give the options for the discretization, initial and terminal conditions, and then the initial guesses for the states. Afterward, iterating over the whole timeframe in smaller intervals using direct collocation, we optimize the controls for maximum distance traveled along the x-axis. Our Matlab implementation solves the problem first with a few discretization points. Then the number of points is increased, the previous solution is then interpolated to provide a good initial solution, and the new problem is then solved. This allows for a faster computation of the problem than just starting directly with a larger number of points would.

Collcon.m function is used to inform the main script of the values of the constraints at the given interval points. In the function, we compute $\dot{x}$ at the discretization point using another function dy.m. Then we compute the piecewise polynomial and its derivative. Then again, using dy.m we compute the state equations of the system at the center of the intervals. Lastly, we obtain the collocation constraints. The function objfun.m is used to simply return the value of a function to be maximized at a given point, in our case the x-coordinate at the end. Function dy.m represents the parameters, state variables and the state equations of the system.

Scaling by the variables of an optimization problem converts the variables from units that reflect the physical nature of the problem, to units that display certain desirable properties during the optimization process. The scaling should be done so that the variables of the scaled problem should be of similar magnitude in the region of interest. A common way of performing the scaling is to scale all the variables close to an interval of $[0, 1]$ or $[-1, 1]$. The scaling can affect the optimization in various ways. First, the optimization methods are usually based on some convergence criteria and tolerances. This means that some algorithms can have difficulties with "small" and "large" values of variables when trying to determine a stopping criteria for the optimal solution. Second, by scaling the variables, we are making sure that each variable has relatively similar importance in the optimization. This can also be thought of as making the weights of each variable similar. Third, as we are numerically optimizing using Matlab, the finite-precision calculations can be affected significantly depending on the scaling. The reasoning for scaling the objective function is quite similar to that of the variables. The magnitude of the objective function should be close to the order of unity in the region of interest. This can be thought of as scaling a vector by its norm to achieve a unit vector. This usually helps with the convergence and tolerance checks.

3.4.3 Results

Now we can solve the optimal trajectory and control of the glider using the SQP method and our implementation of direct collocation using Matlab. We assume that the glider is initially flying horizontally ($\gamma_0 = 0$), at an altitude $h_0 = 50$m. We analyze and solve the problem of the greatest increase along the x-axis for a drop of 10m in altitude ($h_f = 40$m). We also assume that the terminal velocity is $v_f = 10$m/s, to prevent the glider for stalling as we determined in equation 8 .

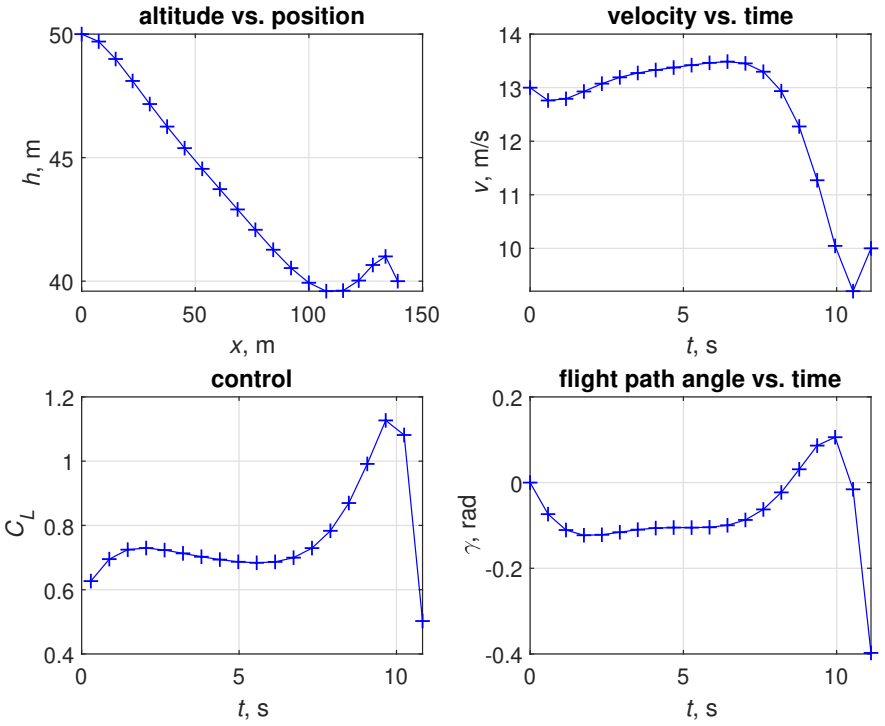

Figure 6: Results of the optimal control problem.

Figure 6 shows the resulting optimal trajectory and states after the optimization is conducted. Particularly, the altitude vs. position plot provides an overall view of the flight. We notice immediately, that the optimal distance travelled is around 140 meters, which is achieved by performing a final jump at the end of the flight. By analyzing the velocity vs. time plot it can be seen that the optimal distance is accomplished by accumulating velocity towards the end of the flight and finally converting the resulting high velocity into lift by increasing the control, as can be seen from the control plot in Figure 6. The flight path angle vs. time plot confirms the aforementioned optimal behaviour of the glider. The optimal trajectory is somewhat affected by the requirement of having final velocity of 10 m/s, since after the jump, velocity must be acquired enough to reach the required 10 m/s. Omitting this requirement could improve the optimal distance, but the requirement ensures that the real-life stalling speeds cannot be achieved at the end. Also, the requirement guarantees, that not all velocity is converted into lift and that the glider does not fall at end of the optimization altitude of 40m. Furthermore, it is worth mentioning, that the velocity drops slightly below the stalling speed at around 10.5s, but this is not problematic, since the control drops simultaneously to lower levels, which allows for smaller velocities.

Number of discretization points	20
Distance travelled $(x(t_f) - x(0) = x(t_f))$	139.08 m
Terminal velocity $(v(t_f))$	10.00 m/s
Terminal time (t_f)	11.11 s
Optimization time	1.83 s

Table 2: Characteristics of the optimal flight.

Some key properties about the flight and the optimization procedure are collected in Table 2. Throughout the optimization, the number of discretization points was increased by starting from 5 points and increasing them by three at a time eventually arriving at 20 discretization points. Also, Table 2 presents the final optimal distance travelled and the elapsed time of the flight.

Comparing the results of the optimal control problem to the static problem from Section 3.1 we can see significant differences. The most remarkable difference is that with the non-static control the glider can achieve 40 % longer distances compared to the optimal static control flight. By comparing the optimal trajectories in Figures 5 and 6 one can conclude that the most noticeable difference is the jump at end of the non-static control flight. That is, the trajectories are relatively similar elsewhere, but the non-static glider achieves the additional 40 meters by performing the bounce. This can be confirmed by the control plot in Figure 6, which states that the decent in the first half of the optimal non-static flight is performed using approximately the same constant control ($C_L \approx 0.7$) that was calculated in Section 3.1. To summarize, the ability to vary the control throughout the flight has significant effects on the optimal distance travelled as well as the optimal trajectory, by allowing to use the additional accumulated velocity towards the end of the flight in the form of the jump.

3.4.4 On scaling of the variables

Our Matlab implementation also returns the Lagrange multipliers for each iteration. We know that after reversing the effects of scaling, we could multiply the multipliers with $\frac{3}{2\Delta t}$, and the results would approximate the costates at the centers of the discretization segments. Here Δt is the length of the discretization segment. Using this information, we can determine how we could reverse the effect of scaling, as we have the numerical values of the Lagrange multipliers and the timesteps at hand.

Let us define λ_i^k and $\tilde{\lambda}_i^k$ as the nonscaled and scaled Lagrange multipliers of state variable i at discretization point k; and Δt and $\Delta\tilde{t}$ as the nonscaled and scaled timesteps. Now, we know that if we reverse the scaling by multiplying the scaled Lagrange multiplier with some scaling factor s^k, we should get the same approximations of the costates at the centers of the discretization segments.

$$\lambda_i^k \cdot 1 \cdot \frac{3}{2\Delta t} \approx p_i^{mid_k} \qquad \forall i \in \{x, h, v, \gamma\} \tag{54}$$

$$\tilde{\lambda}_i^k s_i^k \cdot \frac{3}{2\Delta\tilde{t}} \approx p_i^{mid_k} \qquad \forall i \in \{x, h, v, \gamma\} \tag{55}$$

$$\implies \frac{\lambda_i^k/\Delta t}{\tilde{\lambda}_i^k s_i^k/\Delta\tilde{t}} \approx 1 \qquad \forall i \in \{x, h, v, \gamma\} \tag{56}$$

$$\implies s_i^k = \frac{\lambda_i^k \Delta\tilde{t}}{\tilde{\lambda}_i^k \Delta t} \qquad \forall i \in \{x, h, v, \gamma\} \tag{57}$$

Using Equations 54-57 and our numerical values obtained from Matlab, we can visualize the reversing of the scaling.

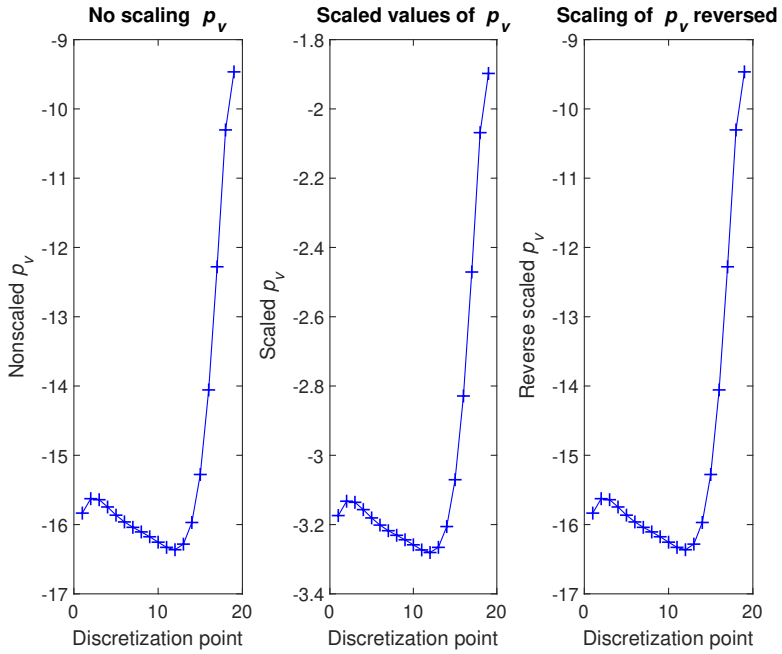

Figure 7: Effect of scaling and reversing the scaling of costate p_v.

In Figure 7, we see first the costate values of velocity p_v when we optimize without any scaling. The middle plot shows what the values are if we would scale them. The last plot on the right shows the values when the scaling is reversed using Equation 57, e.g. $p_v^{mid_k} \approx \tilde{\lambda}_v^k s_v^k \cdot \frac{3}{2\Delta\tilde{t}}$.

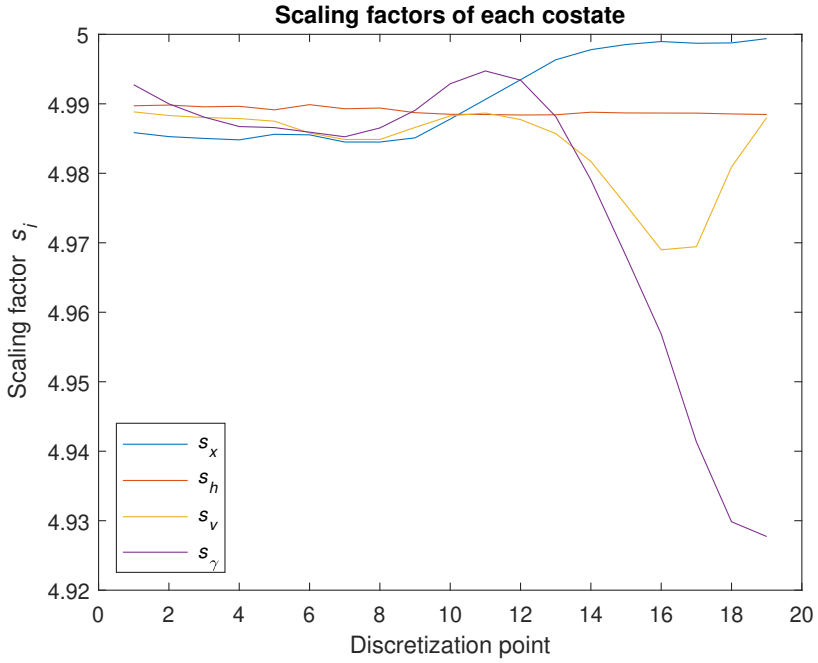

Figure 8: Scaling factors of the costates.

Figure 8 presents the scaling factors of each costate in each discretization point. Here we can see that the variations are quite small, and thus it would almost be sensible to just take the average of them and scale all the costates with just one numerical factor. Although, the scaling factor of p_γ has a larger variation at the last few discretization points and we do not see where it evolves if we would have had more discretization points.

Figure 9 shows the costate values of velocity when using the indirect multiple shooting method for the same problem. As we can see, compared to Figure 7, the values are quite similar with both methods.

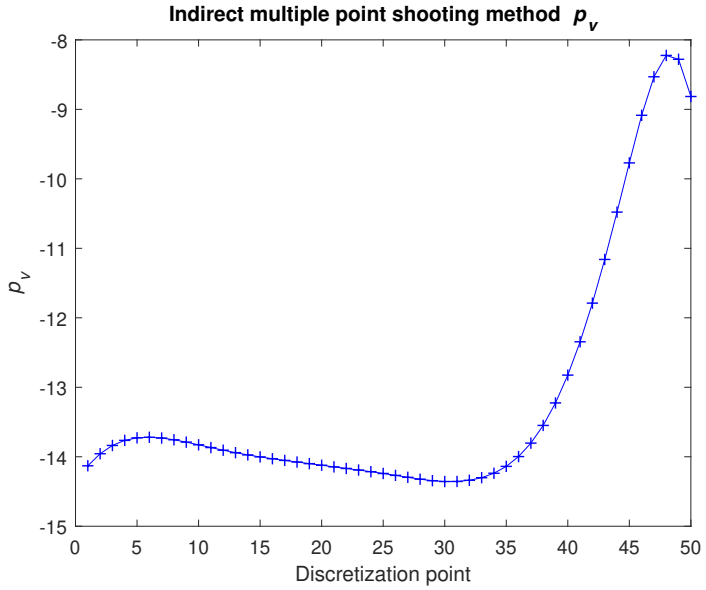

Figure 9: Costate p_v provided by the indirect method.

We can now also compute the optimal control for the indirect multiple shooting method using the equation 34 for each discretization point.

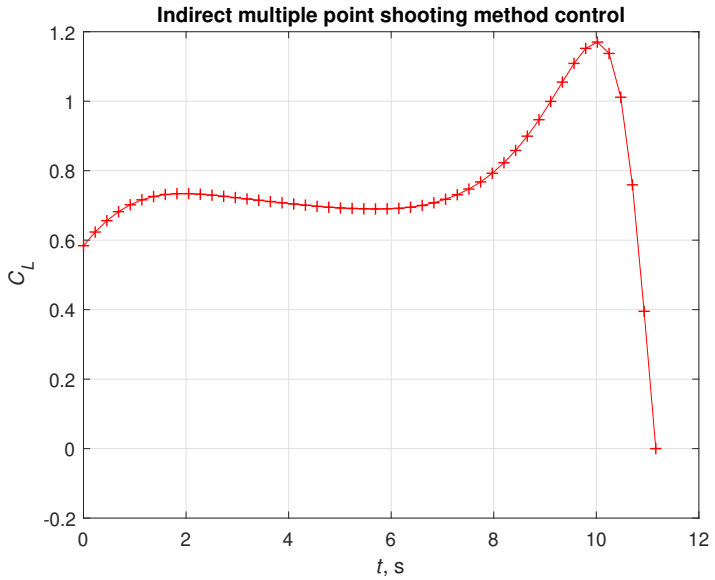

Figure 10: Optimal control provided by the indirect method.

Again, comparing the optimal control of the indirect multiple shooting methods in Figure 10 to the optimal control from our direct collocation in Figure 6, we can see that the controls are similar. From Figures 9 and 10 we can state that the solution obtained from the indirect multiple shooting methods is practically the same as with our SQP method and direct collocation.

3.5 Glider optimization in thermal

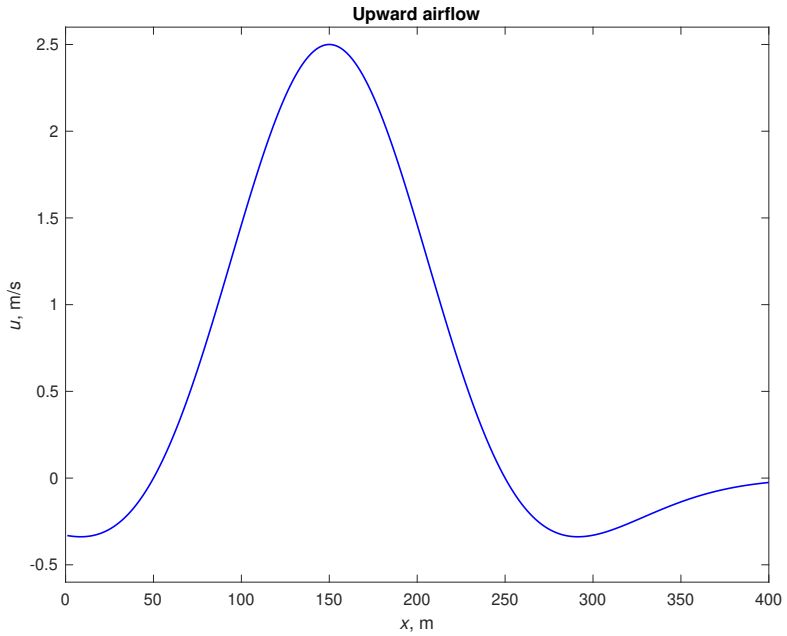

Figure 11: Upward air flow as a function of horizontal distance.

Finally, we shall extend our optimization model to simulate changing air flow conditions. Particularly, we will introduce upward air flow (or thermal) conditions as a function of x-coordinate. The upward air flow $u(x)$ can be characterized as follows

$$u(x) = 2.5\exp - (x - x_{A0})^2/R^2(1 - (x - x_{A0})^2/R^2), \tag{58}$$

where the parameters x_{A0} and R are set to 150m and 100m, respectively. The intensity of the upward air flow is shown in Figure 11. In order to model the glider in such conditions, the state equations in 2 need to be varied. That is, we will continue to use the state variables x and h as previously, but we will replace state variables v and γ with vertical and horizontal velocities v_x and v_h. These new state variables have the following state equations

$$\begin{cases} \dot{v_x} = (-L\sin(\eta) - D\cos(\eta))/m \\ \dot{v_h} = (L\cos(\eta) - D\sin(\eta) - mg)/m, \end{cases} \tag{59}$$

where $\eta = \arctan((v_h - u(x))/v_x)$ and the velocity relative to the air is $v_r = \sqrt{v_x^2 + (v_h - u(x))^2}$.

Before proceeding to solving the optimization problem in thermal setting, we will confirm the feasibility of the above state equations. This is conducted by setting the air flow to zero, $u(x) = 0$, and verifying that the state equations become equivalent to those stated in 2. First we notice that when $u(x) = 0$, then η becomes γ, since $\eta = \arctan((v_h - 0)/v_x) = \gamma$. Now we can derive the original state equations from $\dot{v_x}$ and $\dot{v_h}$. Applying the identity $\cos^2(\gamma) + \sin^2(\gamma) = 1$, we get the state equation for v as follows

$$\dot{v} = \frac{d}{dt}\sqrt{v_x^2 + v_h^2} = \frac{1}{2}(v_x^2 + v_h^2)^{-\frac{1}{2}}(2v_x\dot{v_x} + 2v_h\dot{v_h}) \tag{60}$$

$$= \frac{1}{2v}\left(2v\cos(\gamma)\dot{v_x} + 2v\sin(\gamma)\dot{v_h}\right) \tag{61}$$

$$= \frac{1}{2v}\left(\frac{2v}{m}\cos(\gamma)(-L\sin(\gamma) - D\cos(\gamma)) + \frac{2v}{m}\sin(\gamma)(L\cos(\gamma) - D\sin(\gamma) - G)\right) \tag{62}$$

$$= \frac{1}{m}\left(-L\cos(\gamma)\sin(\gamma) - D\cos^2(\gamma) + L\sin(\gamma)\cos(\gamma) - D\sin^2(\gamma) - G\sin(\gamma)\right) \tag{63}$$

$$= \frac{1}{m}(-D(\cos^2(\gamma) + \sin^2(\gamma)) - G\sin(\gamma)) \tag{64}$$

$$= \frac{-D - G\cos(\gamma)}{m}. \tag{65}$$

Similarly, we can derive a formula for $\dot{\gamma}$

$$\dot{\gamma} = \frac{d}{dt}\arctan\left(\frac{v_h}{v_x}\right) = \frac{1}{\left(\frac{v_h}{v_x}\right)^2 + 1}\frac{d}{dt}\left(\frac{v_h}{v_x}\right) \tag{66}$$

$$= \frac{1}{\left(\frac{v_h}{v_x}\right)^2 + 1}\left(\frac{\dot{v_h}v_x - v_h\dot{v_x}}{v_x^2}\right) \tag{67}$$

$$= \frac{\dot{v_h}v_x - v_h\dot{v_x}}{v_h^2 + v_x^2} \tag{68}$$

$$= \frac{1}{v^2}\left(\dot{v_h}v\cos(\gamma) - \dot{v_x}v\sin(\gamma)\right) \tag{69}$$

$$= \frac{1}{mv}\left(\cos(\gamma)(L\cos(\gamma) - D\sin(\gamma) - G) - \sin(\gamma)(-L\sin(\gamma) - D\cos(\gamma))\right) \tag{70}$$

$$= \frac{1}{mv}(L\cos^2(\gamma) - D\sin(\gamma)\cos(\gamma) - G\cos(\gamma) + L\sin^2(\gamma) + D\cos(\gamma)\sin(\gamma)) \tag{71}$$

$$= \frac{1}{mv}(L(\cos^2(\gamma) + \sin^2(\gamma)) - G\cos(\gamma)) \tag{72}$$

$$= \frac{L - G\cos(\gamma)}{mv}. \tag{73}$$

From the derivations above we can conclude that by setting the upward air flow $u(x)$ to zero we recover the original state equations as is desired. In the following section, we will implement the new state equations to the model and solve the problem similarly using Matlab and interpret the results.

3.5.1 Results of the thermal flight

The results are calculated with Matlab as previously using the same initial and terminal conditions. That is, with the new state variables, the initial velocity and angle can be expressed as $v_x(0) = 13$ m/s and $v_h(0) = 0$ m/s. Additionally, the final velocity requirement of 10 m/s was implemented by adding end boundary conditions for both v_x and v_h. Since constraining both velocity components simultaneously also determines the flight path angle γ, the final velocites were chosen such that the final flight path angle would be optimal. This optimal γ was determined by first running the results without final velocity requirement, and then using that angle to choose the components v_x and v_h, thus resulting in values $v_x(t_f) = 9.2$ m/s and $v_h(t_f) =$ -3.9 m/s.

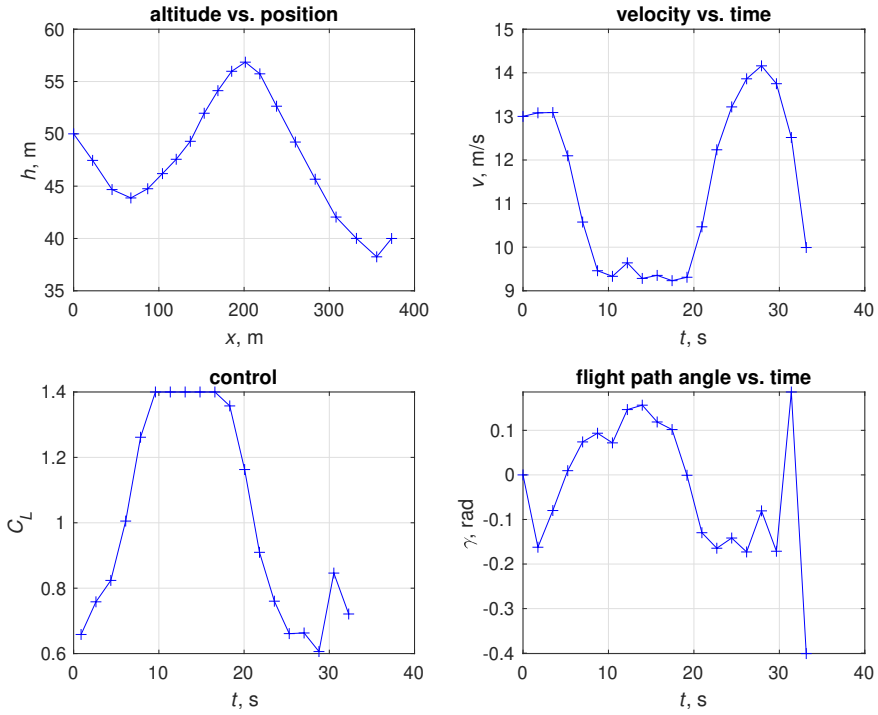

Figure 12: Results of the optimal control problem in thermal.

The results of the optimal flight in thermal conditions are presented in Figure 12. As can be observed from the figure, the optimal trajectory of the glider is noticeably different when the upward air flow is introduced. Particularly, the shape of the trajectory correlates heavily with the shape of the air flow presented in Figure 11. The optimal trajectory suggests that when the air flow increases the glider gains altitude and will start to decent again when the air flow has reached its maximum value. During the ascent, the velocity of the glider decreases and starts to grow considerably after the large descent begins. According to the control plot in Figure 12, the control has again an important role in maximising the travelled distance. During the high upward air flow, the control is increased and is constrained by the limit in order to exploit all the potential of the additional lift provided by the air flow. Also, during the descent, the control is not set to the level of optimal constant control ($C_L \approx 0.7$) but is gradually decreased in accordance with the decreasing air flow, while keeping the control above 0.7 to make use of the additional lift. As with the model without thermal, the trajectory includes a jump at the end of the flight to take advantage of the remaining velocity for achieving additional meters. Again, the jump can be noted from the plots of the trajectory and control, however, being smaller compared to the previous model. Finally, it is worth noting that the velocity vs. time plot reveals that during the high upward air flow, the glider's velocity is in the neighbourhood of the stalling speed while applying the maximum control. According to the calculations in Section 2.2, the glider should stall in these conditions. However, the calculation does not apply here, since the thermal provides additional lift which consequently decreases the stalling speed, so the model is valid.

Number of discretization points	20
Distance travelled ($x(t_f) - x(0) = x(t_f)$)	373.10 m
Terminal velocity ($v(t_f)$)	10.00 m/s
Terminal time (t_f)	33.15 s
Optimization time	2.49 s

Table 3: Characteristics of the optimal flight in thermal conditions.

Table 3 collects some key aspects of the flight and the optimization process. Again the final number of discretization points was set to 20. The table, together with the Figure 12, reveals the biggest difference between the original and the thermal optimization models. Particularly, the glider glides over 200 meters longer when the upward air flow is introduced, so the thermal increase the travelled distance by over 2.5 times. One can thus conclude, that upward lift is in a key role in maximizing the travelled distance with a glider. Also, the time taken to the flight was three times higher with the thermal flight, in accordance with the flight distance. Finally, it can be concluded that the introduction of the new state equations in the thermal model increased the optimization time by some 40 percent, which is to be expected as the state equations became more complex due to the upward air flow expression.

4 Summary and improvements

In this project, we have formulated and solved a dynamic optimization problem for maximizing the distance travelled by a glider. The task consists of multiple stages for finally arriving at an optimal control policy for the glider. The formulation started with considering the glider as a point mass in air and deriving the state equations from the fundamental laws of mechanics. In this step, some approximations were used, for example, omitting the moment forces when approximating the glider as a point mass and assuming the angle of attack α small to use angle invariant controls. These approximations were still justifiable to maintain the derivations and calculations relatively simple. After deriving the state equations, the model was tested by simulating the glider with different values of parameters in order to investigate the legitimacy of the model. From the simulations we concluded that the derived state equations describe the behaviour of a glider reasonably well and changing the parameter values yields behaviour that is expected from real life.

After accepting the model for the glider, a static optimization model was solved in order to provide context for the actual dynamic optimization model. The dynamic problem was derived by stating the objective function and formulating the necessary conditions and costates using the theory of dynamic optimization. Once the problem was set, we also confirmed that the problem does not have singular control solutions. After we discussed the aspects of different dynamic optimization numerical solution methods, we were finally ready to implement the problem using Matlab. The method of choice was the sequential quadratic programming method using direct collocation. This procedure for solving the optimal control and trajectory proved to be an extremely powerful way of coping with this type of dynamic problem. That is, the solution times were fast even when using 20 discretization points yielding sufficiently smooth trajectories and controls. Furthermore, we also concluded that the method of choice provided similar solutions to those given by the other

alternative, the indirect multiple shooting method. The optimal solutions to the problem turned out to be reasonable and the trajectory had some justifiable characteristics to it. Particularly, the optimal angle of the descent and the use of the velocity to make the last jump at the end of the flight. Finally, we modified the model by introducing the upward air flow, which was incorporated into the state equations of the glider. The solution to the extended problem was solved using the same procedure as previously. Again, the solutions to the thermal problem were rational and the control employed the additional lift provided by the thermal.

Finally, we will provide some discussion and suggestions about the assignment itself. The assignment in its diversity was a challenging task to complete. However, due to the variety of the assignment, we considered it quite interesting. Despite the challenging tasks of dynamic programming, state equations and numerical methods, for example, it helped us that the substance was somewhat familiar, so stating the conclusions and finding errors was easier. Also, it helped quite extensively that the majority of the Matlab code was provided, hence we could concentrate on the method itself, rather than syntax and Matlab functions. Furthermore, we appreciated the instructions in the sense that they were relatively detailed and provided some background as well. However, in some parts of the instructions, we were not quite sure in which level of detail some parts should be reported. Particularly, in some parts, we were wondering whether we should show the result strictly mathematically or describe the content more verbally. One example of this was when we had to show how could we determine the unknown final time. Also, we could have appreciated it if there had been some more teaching on the contents of the assignment. Especially, even though we had taken the course on dynamic optimization, the course did not provide examples that were in the scale of complexity of this assignment. Hence we could have benefited from the extra teaching on the topic. Lastly, we took advantage of the weekly reception hours quite extensively, so it might be that some students could benefit if there were a few more of those sessions, to proceed with the assignment at greater pace.