Optimal Flight with a Glider

MS-E2132 Laboratory Assignments in Operations Research II

18.10.2024

Patience Anipa - 101541866

Contents

1 Objective

The objective of this project is to optimize the flight of a glider by finding the best control that maximizes the glider's horizontal distance for each unit of lost altitude. We aim to solve the discrete time dynamic optimization problem using the direct solution method.

2 Background

Gliders are aircraft that fly without an engine, relying on external forces such as gravity and lift to navigate. In the context of optimal control, the primary forces acting on a glider include gravity, lift, and drag. The lift is generated by the aerodynamic properties of the glider's wings, which vary with the angle of attack and velocity. The goal is to control the lift coefficient (C_L) to optimize the glider's trajectory, maximizing the distance traveled horizontally per unit of altitude lost.

The flight dynamics of the glider are captured through state-space representation, which involves simplifying the motion to a vertical plane. The state variables include the x-coordinate (horizontal position), h (altitude), v (velocity), and γ (flight path angle). The control variable is the lift coefficient C_L. The glider's behavior is described by a system of first-order differential equations, derived using principles of aerodynamics.

Two distinct flight scenarios are considered:

- Windless conditions, where the glider is only subject to gravity and aerodynamic forces.
- Thermal conditions, where an upward airflow (thermal) affects the glider's flight path.

The dynamic optimization problem is solved using Sequential Quadratic Programming (SQP) in MATLAB.

3 Modelling for Optimization

A six degrees of freedom model is typically needed to describe the movement and rotation of an aircraft, requiring the moment of inertia and external forces. However, since rotational dynamics are much faster than translational ones, the aircraft can be simplified as a point mass, ignoring rotations. The flight is examined in a vertical plane with state variables: the x-coordinate, altitude h, velocity v, and flight path angle γ. The lift coefficient C_L serves as the control variable, and air density is assumed constant due to small changes in altitude.

In this section, a free body diagram of the glider is drawn to show the magnitude and direction of the forces acting upon it. The diagram is used to derive the state equations of the model using the state variables. Finally, we investigate the model by simulating it using numerical integration and observing the trajectories of the glider. MATLAB routines 'ode45' is used for simulation.

3.1 Model of the Glider

Below is a drawing of a free body diagram of the glider, that includes the magnitude and direction of forces acting upon it.

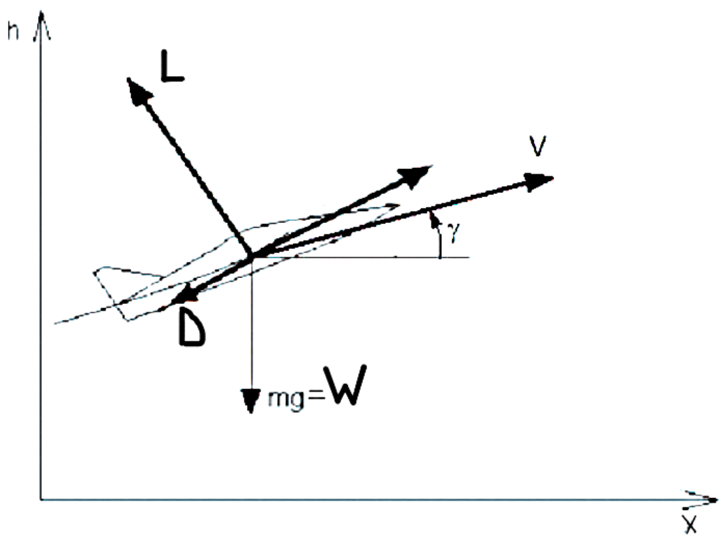


Figure 1: Forces acting on the glider: L, the lift, W, the force of gravity (W = mg), and D, the drag. γ is the flight path angle, which is the angle formed between aircraft's velocity vector V (tangential direction to the trajectory) and the horizontal.

The force of lift, L, created by the airflow around the wings is perpendicular to the trajectory, and the force of drag, D, is parallel to the trajectory.

Lift and drag are proportional to the surface area of the wings, S, and the dynamic pressure, $\frac{1}{2}\rho V^2$, where ρ is the density of air, and V is the forward velocity of the aircraft. The equations for lift and drag are thus:

$$L = \frac{1}{2}\rho V^2 C_L S \tag{1}$$

$$D = \frac{1}{2}\rho V^2 C_D S \tag{2}$$

3.2 State Equations of the Glider

Recall the state variables; x-coordinate, the altitude h, the velocity V, and the flight path angle γ.

Equations of motion:

Newton's second law of motion applied to the motion tangential to the trajectory(aircraft's velocity vector):

$$m\frac{dV}{dt} = -mg\sin\gamma - D \tag{4}$$

where V is the speed of the glider, m is its mass, g is the acceleration due to gravity, and D is the drag force given by Eq. (2).

The negation of the first term on the right-hand side of Eq. (4) matches the intuition: when γ is negative, the nose of the glider is pointing down, and accelerates due to gravity. When $\gamma > 0$, the glider must fight against gravity.

In the normal direction, we have the centripetal force $\frac{mV^2}{r}$, where r is the instantaneous radius of curvature. After noticing that:

$$\frac{d\gamma}{dt} = \frac{V}{r},$$

this can be expressed as $V\frac{d\gamma}{dt}$, giving:

$$mV\frac{d\gamma}{dt} = -mg\cos\gamma + L \tag{5}$$

where L is the lift force given by Eq. (1).

Finally, rewriting the equations with the expressions for drag D and lift L, we get:

$$m\frac{dV}{dt} = -mg\sin\gamma - \frac{1}{2}\rho V^2 C_D S \tag{6}$$

$$mV\frac{d\gamma}{dt} = -mg\cos\gamma + \frac{1}{2}\rho V^2 C_L S \tag{7}$$

Glider Trajectory:

To obtain the flight trajectories predicted by this model, we integrate the spatial coordinates, which depend on both the forward velocity and the trajectory angle. Flight is examined in a vertical plane designated by coordinates (x, h) with respect to an inertial frame of reference, and these are obtained from:

$$\frac{dx}{dt} = V\cos(\gamma) \tag{8}$$

$$\frac{dh}{dt} = V\sin(\gamma) \tag{9}$$

State Equations of Glider:

Thus the state equations of the glider are given by a system of four first-order differential equations:

$$\dot{x} = V\cos(\gamma) \tag{a}$$

$$\dot{h} = V\sin(\gamma) \tag{b}$$

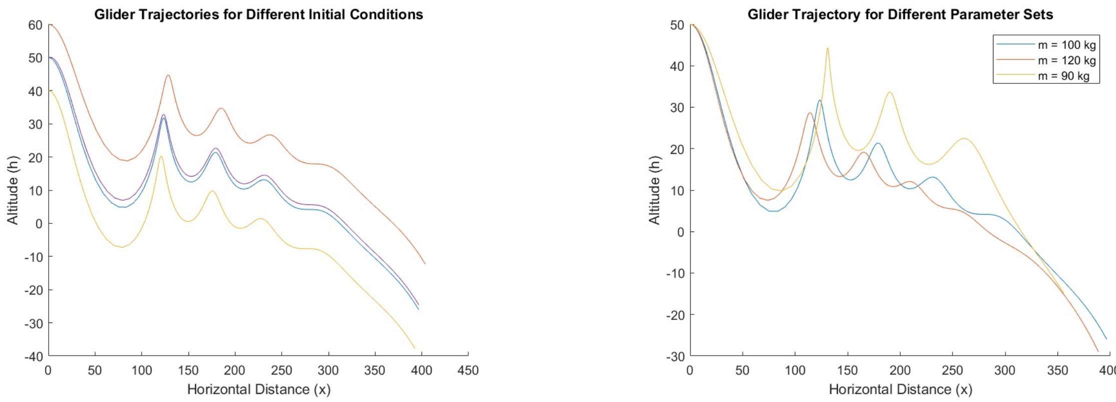
$$\dot{V} = -g \sin\gamma - \frac{D}{m} \tag{c}$$

$$\dot{\gamma} = \frac{-g\cos\gamma}{V} + \frac{L}{mV} \tag{d}$$

Where D and L are the drag and lift forces given by Equations (1) and (2).

3.3 Model Simulation

3.3.1 Simulations at different initial conditions and alternative parameter values



(a) (b)

Figure 2: On the left, (a), a simulation of the glider path with different initial conditions in order of higher altitude and velocity, predefined initial conditions, a non-zero flight path angle and lower altitude and velocity. On the right, (b), a simulation with alternative parameter values for one of the initial conditions. Glider with higher mass has higher corresponding parameter values compared to the other gliders.

In glider model, it can be observed how the parameters affect the glider's performance, particularly its trajectory. A heavier glider will generally lose altitude more quickly because more force is needed to counteract gravity. However, it may also maintain momentum better, traveling farther horizontally if enough lift is generated. Higher Zero Lift Drag Coefficient values increase drag, reducing the glider's horizontal distance and potentially causing it to lose altitude faster. The induced drag coefficient accounts for drag generated by lift. Higher values of k increase drag at higher lift coefficients (CL), leading to greater energy loss as lift is generated. The reference area determines how much lift and drag the glider can generate. A larger wing area increases both lift and drag, which may allow the glider to maintain altitude but also increases drag, slowing it down. Higher air density increases both lift and drag forces because the dynamic pressure is directly proportional to the air density. In denser air, the glider can generate more lift but also faces greater drag. Gravitational acceleration influences the glider's descent rate. A higher gravitational acceleration would cause the glider to descend faster, while a lower gravitational acceleration slows its descent.

3.3.2 Stalling speed

The stalling speed of an aircraft is the minimum speed at which it can maintain level flight without losing altitude. If the aircraft's speed drops below this point, it will enter a stall, which is a sudden loss of lift causing the aircraft to descend.

The following calculations are performed to find the flight speed in steady state and from it the stall speed can be calculated.

As established earlier the lift force L is given by:

$$L = \frac{1}{2}\rho V^2 C_L S$$

For horizontal flight ($\gamma = 0$), the lift force must balance the gravitational force W:

$$L = W = mg$$

Substituting the lift equation:

$$L = \frac{1}{2}\rho V^2 C_L S = mg \tag{1}$$

Solving for V to find the flight speed:

$$V = \sqrt{\frac{2mg}{\rho S C_L}} \tag{10}$$

Equation (10) is the Aircraft Speed Equation for steady level flight.

The minimum possible flight speed occurs at $C_{L_{max}}$ - just before stall. Where $C_{L_{max}}$

is the maximum lift coefficient. The stall speed may be determined if $C_{L_{max}}$ is known.

From the parameters given for the range C_L = [-1.4, 1.4], $C_{L_{max}}$ is 1.4.

We substitute the value of $C_{L_{max}}$ into Equation (10).

The stalling speed is 9.4115 m/s.

3.3.3 Simulations of horizontal flights at speeds lower than the stalling speed

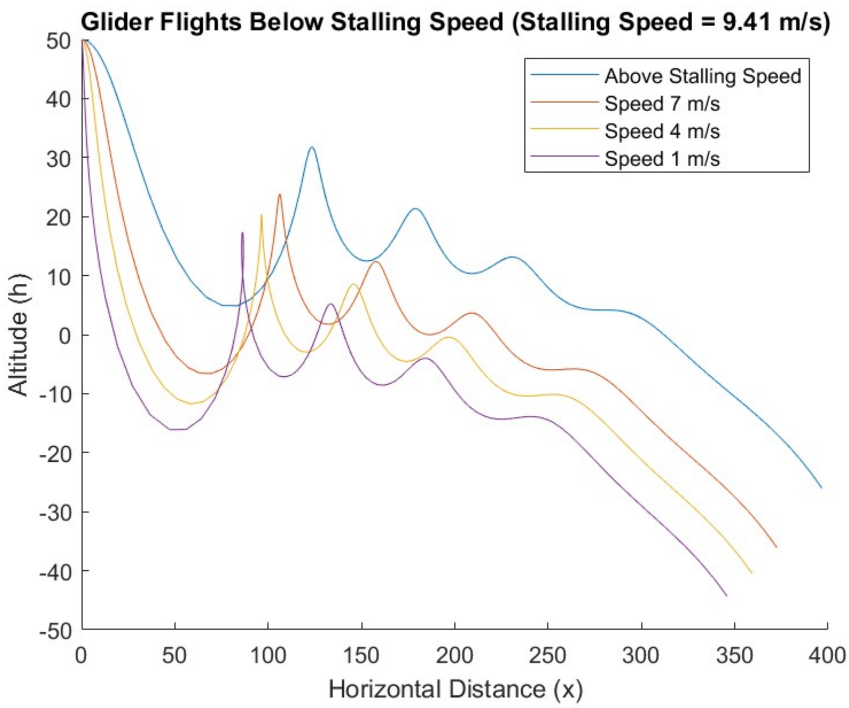


Figure 3: Below the stalling speed, the glider loses altitude rapidly because the lift force is insufficient to counterbalance gravity. This is apparent from the steep descent in the flights.

The model assumes valid lift and drag relationships, but under the conditions of very low speeds, the assumptions about aerodynamic forces break down. For example, airflow over the wings could become turbulent, causing the glider to stall. This is observed in a significant loss of altitude when simulating flight at speeds below the stalling speed, which reflects the real-world behavior of stalling. Therefore the model is valid in this situation.

4 Optimization

The objective of the flight is to glide as far along the x-axis as possible for each unit of lost altitude. The theory of aerodynamics reveals that this ratio is maximized when the ratio between the total drag coefficient and the lift coefficient is as small as possible.

In this section, we optimize the problem according to the specification given. We first consider the static time-invariant optimization problem, where the flight path angle is close to zero and the velocity and flight path angle are constant. The dynamic formulation of the problem is solved subsequently.

4.1 Static Optimization Problem

4.1.1 Static time-invariant optimization problem

Modifying the state equations on the assumptions that the flight path angle is close to 0 ($\sin\gamma \approx \gamma, \cos\gamma \approx 1$) and that the velocity and the flight path angle are constant, we get the following state equations:

$$x' = V \tag{e}$$

$$h' = V\gamma \tag{f}$$

$$V' = -g\gamma - \frac{D}{m} = 0 \tag{g}$$

$$\gamma' = \frac{-g}{V} + \frac{L}{mV} = 0 \tag{h}$$

Since the objective is to glide as far along the x-axis as possible for each unit of lost altitude, the objective function is as follows:

$$\text{Maximize} \quad \frac{x'}{-h'}$$

Which can be re-written as:

$$\text{Maximize} \quad \frac{V}{-V\gamma} = \frac{1}{-\gamma}$$

To find the optimal control C_L that maximizes the objective function, we need to express γ as a function of C_L.

To determine the equation for γ in terms of C_L, we use the following equations:

$$V' = -g\gamma - \frac{D}{m} = 0 \tag{g}$$

$$\gamma' = \frac{-g}{V} + \frac{L}{mV} = 0 \tag{h}$$

Express Equation (h) in terms of V^2:

$$V^2 = \frac{2mg}{\rho S C_L}$$

Express Equation (g) in terms of γ and substitute V^2 to get the following equation for γ:

$$\gamma = -\frac{C_D}{C_L} = -\frac{C_{D_0} + KC_L^2}{C_L}$$

Then we substitute γ into the objective function and proceed as follows:

$$\text{Maximize} \quad \frac{1}{-\gamma} = \frac{C_L}{C_{D_0} + KC_L^2}$$

subject to:

$$C_{D_0} = 0.034$$

$$K = 0.07$$

$$C_L \in [-1.4, 1.4]$$

The maximum lift-to-drag ratio can be found by minimizing the drag-to-lift ratio $\frac{D}{L}$. Now, to minimize $\frac{D}{L}$, we first express it as:

$$\frac{D}{L} = \frac{C_{D0} + KC_L^2}{C_L}$$

Next, differentiate $\frac{D}{L}$ with respect to C_L and equate to zero:

$$\frac{d}{dC_L}\left(\frac{C_{D0} + KC_L^2}{C_L}\right) = 0$$

Simplifying the derivative:

$$\frac{d}{dC_L}\left(\frac{C_{D0}}{C_L} + KC_L\right) = -\frac{C_{D0}}{C_L^2} + K = 0$$

Solving for C_L:

$$C_L = \sqrt{\frac{C_{D0}}{K}}$$

Solving this equation gives the value $C_L = 0.696932054$.

Thus, the optimal control C_L which maximizes the flight distance for the static time-invariant flight, is 0.696932054.

4.1.2 The (C_L,C_D) curve

The relationship between lift and drag is given through the aircraft drag polar - often plotted as C_L vs C_D. The drag polar shows the aerodynamic efficiency of a given aircraft - that is, it represents the lift-to-drag ratio.

The time independent optimal control of the above static time-invariant optimization problem can be obtained graphically by analyzing the (C_L,C_D) curve.

The maximum lift to drag ratio is the tangent of the line drawn from the origin to the curve. At this tangent, we can find the optimal control C_L.

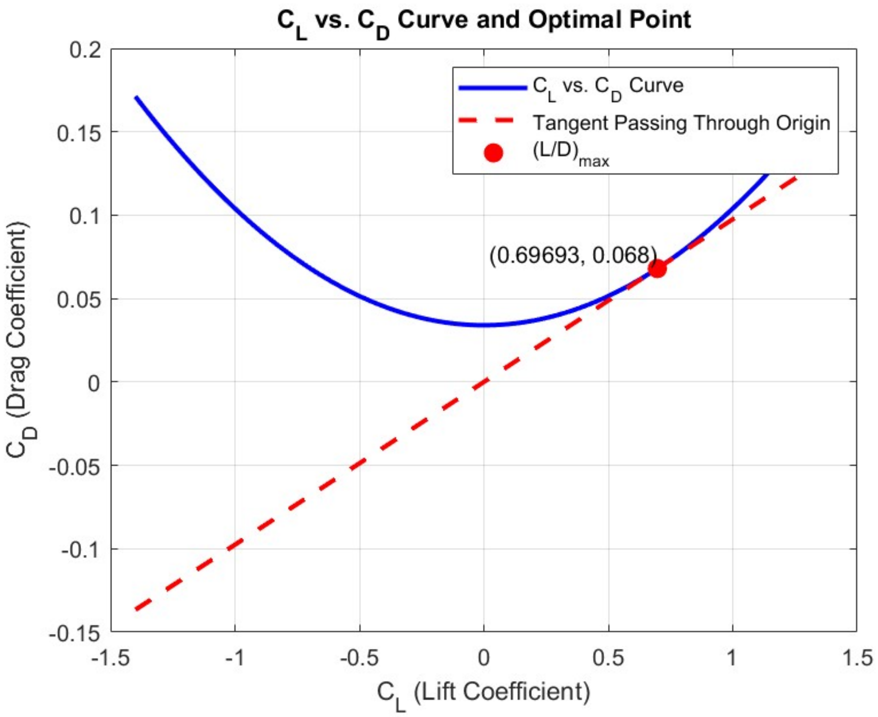


Figure 4: The (C_L,C_D) curve and optimal control at the point of tangency.

4.1.3 Glide distance

We compare the distances travelled by the simulation of the point mass model with constant optimal control and the static optimization problem.

Glide range of static flight:

The glide range x can be expressed in terms of the initial and final altitudes and the glide angle γ:

$$x = \frac{h_0 - h_f}{\tan(\gamma)} \tag{1}$$

Here, γ can be derived from the relationship between lift L and drag D as follows:

$$\tan(\gamma) = \frac{D}{L} \tag{2}$$

$\tan(\gamma)$ can also be expressed in terms of the lift coefficient C_L and drag coefficient C_D.

By substituting the expression for $\tan(\gamma)$ into the glide range equation, we can calculate the maximum glide distance based on the optimal lift coefficient derived from the static optimization problem.

Solving this with an initial altitude of 50 and final altitude of 40, the maximum distance traveled is **102.49 meters**.

Simulation of the point mass model with the constant optimal control:

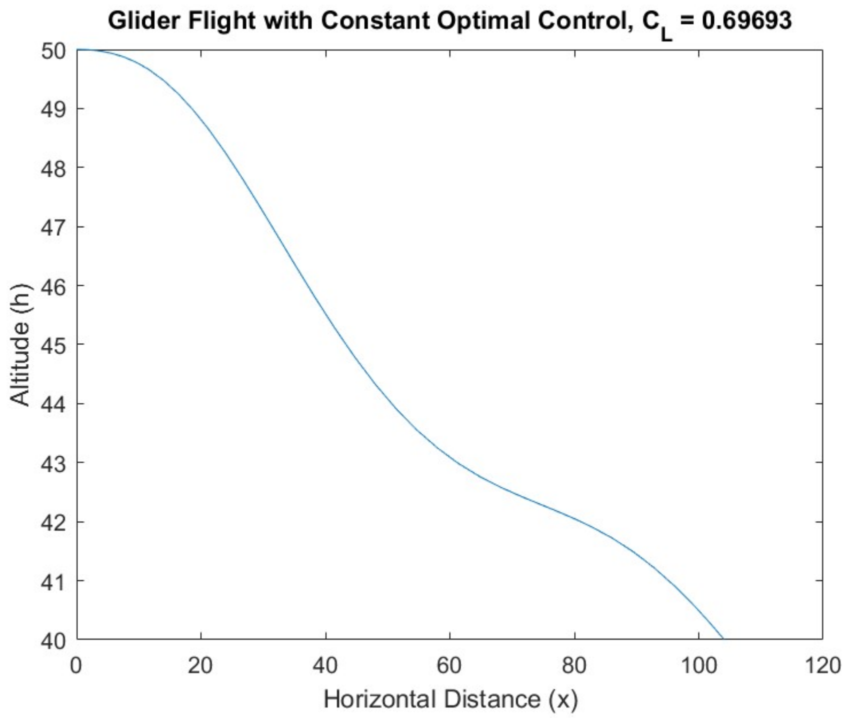


Figure 5: Maximum horizontal distance traveled is 104.0965 meters.

When the point mass model is simulated with the constant optimal control, the glider travels farther. The distance traveled in the static optimization scenario results in a shorter distance since it does not dynamically adjust for changing conditions.

4.2 Dynamic Optimization Problem

The objective of the optimization problem is to maximize the x-coordinate at the end of the flight.

Maximize:

$$J = \int_{t_0}^{t_f} x \; dt = x(t_f)$$

Subject to:

$$
\begin{aligned}
\dot{x} &= V\cos(\gamma) \\
\dot{h} &= V\sin(\gamma) \\
\dot{V} &= -g\sin\gamma - \frac{D}{m} \\
\dot{\gamma} &= \frac{-g\cos\gamma}{V} + \frac{L}{mV}
\end{aligned}
$$

Boundary Conditions:

$$
\begin{aligned}
x(0) &= 0 \\
h(0) &= h_0 \\
V(0) &= V_0 \\
\gamma(0) &= \gamma_0 \\
h(t_f) &= h_{t_f} \\
V(t_f) &> V_{t_f} \\
\gamma(t_f) &= free \\
C_L(t) &\in [-1.4, 1.4]
\end{aligned}
$$

4.3 Optimality Conditions

To find the optimal trajectory and control inputs, we apply the necessary conditions of optimality by employing optimal control theory techniques.

The Pontryagin's Maximum Principle uses the Hamiltonian function to define the necessary conditions for an optimal control. Pontryagin's Maximum Principle states that it is necessary for any optimal control along with the optimal state trajectory to solve the so-called Hamiltonian system, which is a two-point boundary value problem, plus a maximum condition of the control Hamiltonian.

4.3.1 The Hamiltonian function

The Hamiltonian H combines the state variables, costate variables, and the control variable (in this case, the lift coefficient C_L). It is typically structured as:

$$H(X, p, C_L) = L(X, C_L) + p^T \cdot f(X, C_L)$$

Where:

$L(X, C_L)$ is the objective functional (which is $x(t_f)$ in this case),

p^T is the transpose of the costate vector p,

$f(X, C_L)$ contains the state equations in vector form.

Thus, we have:

$$H(X,p,C_L) = x(t_f) + p_x(v\cos(\gamma)) + p_h(v\sin(\gamma)) + p_v\left(-\frac{D}{m} - g\sin(\gamma)\right) + p_\gamma\left(\frac{L}{mv} - \frac{g}{v}\cos(\gamma)\right)$$

Substituting the equations of the Drag and Lift Forces:

$$H(X,p,C_L) = x(t_f) + p_x(v\cos(\gamma)) + p_h(v\sin(\gamma)) + p_v\left(-\frac{(C_{D_0} + KC_L^2)\cdot\frac{1}{2}\rho v^2 S}{m} - g\sin(\gamma)\right)$$

$$+p_\gamma\left(\frac{C_L\cdot\frac{1}{2}\rho v^2 S}{mv} - \frac{g}{v}\cos(\gamma)\right)$$

Final Form of the Hamiltonian:

$$H(X,p,C_L) = x(t_f) + p_x(v\cos(\gamma)) + p_h(v\sin(\gamma)) - p_v\frac{1}{m}\left(\left(C_{D_0} + KC_L^2\right)\cdot\frac{1}{2}\rho v^2 S\right) - p_v g\sin(\gamma)$$

$$+p_\gamma\frac{C_L\cdot\frac{1}{2}\rho v^2 S}{mv} - p_\gamma\frac{g}{v}\cos(\gamma)$$

4.3.2 Extreme value of C_L

To find the extreme value of C_L as a function of states and costates, we take the derivative of the Hamiltonian with respect to C_L and set it to zero.

The relevant terms in the Hamiltonian with respect to C_L is:

$$H_{C_L} = -\frac{p_v}{2m} K C_L^2 \rho v^2 S + \frac{p_\gamma}{2mv} C_L \rho v^2 S$$

Taking the derivative of the Hamiltonian with respect to C_L:

$$\frac{\partial H}{\partial C_L} = -\frac{p_v}{m} K C_L \rho v^2 S + \frac{p_\gamma}{2mv} \rho v^2 S = 0$$

Rearranging gives:

$$\frac{p_\gamma}{2mv} = \frac{p_v}{m} K C_L$$

Thus, solving for C_L:

$$C_L = \frac{p_\gamma}{2 p_v v K}$$

Taking the boundaries of C_L into consideration,

$$-1.4 \leq C_L = \frac{p_\gamma}{2 p_v v K} \leq 1.4$$

C_L is calculated using the resulting values from the optimization problem at the discretization point for the final condition.

$C_L = 0.35082$, which is within the bounds $C_L \in [-1.4, 1.4]$.

To verify the nature of the extreme value (maximum or minimum), we compute the second derivative of the Hamiltonian with respect to C_L.

If $\frac{\partial^2 H}{\partial C_L^2} < 0$, it indicates a local maximum, whereas if $\frac{\partial^2 H}{\partial C_L^2} > 0$, it indicates a local minimum.

Taking the second derivative:

$$\frac{\partial^2 H}{\partial C_L^2} = -\frac{p_v}{m} K \rho v^2 S$$

Calculating this second derivative using the resulting values from the optimization problem gives a non-negative value, thus $\frac{\partial^2 H}{\partial C_L^2} > 0$ and C_L is a local minimum.

4.3.3 Costate equations

The costate equations, which define the necessary conditions for optimality, are derived from the Hamiltonian using the following relationship:

$$\dot{P}*_i(t) = -\frac{\partial H}{\partial X_i}$$

For our state variables x, h, v, γ, the costate equations are computed as follows:

$$\dot{P}*_x = -\frac{\partial H}{\partial x} = -1$$

$$\dot{P}*_h = -\frac{\partial H}{\partial h} = 0$$

$$\dot{P}*_v = -\frac{\partial H}{\partial v} = -p_x \cos(\gamma) - p_h \sin(\gamma) + \frac{p_v}{m} \left(\rho v S (C_{D0} + K C_L^2) \right) - p_\gamma \left(\frac{C_L S}{m} + \frac{g}{v^2} \cos(\gamma) \right)$$

$$\dot{P}*_\gamma = -\frac{\partial H}{\partial \gamma} = -p_x (v \sin(\gamma)) + p_h (v \cos(\gamma)) - p_v g \cos(\gamma) + p_\gamma \frac{g}{v} \sin(\gamma)$$

4.3.4 Initial and terminal conditions

For final time t_f free, the boundary-condition equations are given as:

$$x(0) = 0$$

$$\gamma(0) = \gamma_0$$

$$h(0) = h_0$$

$$v(0) = v_0$$

$$p_x(0) = p_{x0}$$

$$p_\gamma(0) = p_{\gamma 0}$$

$$p_h(0) = p_{h0}$$

$$p_v(0) = p_{v0}$$

$$h(t_f) = h_{t_f}$$

$$v(t_f) > v_{t_f}$$

$$x(t_f) = free$$

$$\gamma(t_f) = free$$

$$H(X^*(t_f), U^*(t_f), P^*(t_f)) = 0$$

4.4 Solution to Dynamic Optimization Problem Using SQP

4.4.1 Principles of Sequential Quadratic Programming (SQP)

The fundamental idea behind Sequential Quadratic Programming (SQP) is to solve a sequence of quadratic subproblems that approximate the original nonlinear problem.

At each iteration, the SQP method forms a quadratic approximation of the Lagrangian function of the original nonlinear optimization problem. The Lagrangian combines the objective function with the constraints, weighted by Lagrange multipliers. The resulting quadratic problem (QP) provides a local model of the original nonlinear problem, which is easier to solve than the full nonlinear problem.

The constraints of the nonlinear optimization problem are linearized at each iteration. This creates a simpler subproblem with linear constraints, which can be solved efficiently using QP solvers.

The SQP method proceeds iteratively by solving a series of these quadratic subproblems. The solution to each subproblem provides a search direction for updating the variables of the original problem. The process continues until convergence, typically when the change in variables and the change in the objective function between iterations become small.

SQP uses the Karush-Kuhn-Tucker (KKT) conditions as a basis for forming the quadratic subproblem. The KKT conditions are necessary for optimality in constrained optimization and ensure that the solution respects both the objective function and the constraints.

A merit function is used to evaluate the quality of the current iterate. The merit function balances the objective function and constraint violation, guiding the algorithm towards feasible and optimal solutions. It helps in choosing step sizes for the variable updates.

Global convergence of SQP is often ensured by using line search or trust region strategies. These techniques modify the step size at each iteration to ensure that the algorithm does not diverge or take steps that violate the problem's constraints.

4.4.2 Description of the MATLAB files

flight _ main.m

This MATLAB script implements Sequential Quadratic Programming (SQP) to solve the dynamic optimization problem.

The initial setup defines the initial and final conditions for state variables. The number of discretization points is set and scaling factors are defined to normalize the state variables and time.

For the optimization setup, the optimization solver uses 'fmincon' with specific settings defined in 'optimset'.

The state and control variables are initialized and these are scaled based on the predefined scaling factors.

In the continuation Loop, the main loop runs from the initial discretization points to the final, gradually refining the solution. Upper and lower bounds are set for the state variables at each iteration. Regarding the objective function, the 'fmincon' solver optimizes the trajectory by solving the nonlinear constraints ('collcon') and objective ('objfun') with the SQP algorithm.

After solving the optimization problem at each discretization level, the state variables are updated using spline interpolation to improve the initial guesses for the next iteration.

The results of the optimization (altitude vs. position, velocity vs. time, control and flight path angle vs. time) are plotted during each iteration to help visualize the glider's trajectory and the evolution of its state variables as the optimization progresses.

After each iteration, the number of discretization points, the final position, the distance traveled, and the terminal velocity and time are printed to the console.

The Lagrange multipliers for the equality constraints are extracted and printed at the end of the simulation.

collcon.m

This MATLAB function, collcon, defines the collocation constraints for a dynamic optimization problem involving the flight of a glider. It is used within the fmincon optimization routine to ensure that the state equations are satisfied throughout the discretization of the time interval. The function extracts and scales the state variables (position, altitude, velocity, flight path angle) and controls (lift coefficient) from the input vector X. It calculates the derivatives of the state variables at the discretization points by calling the 'dy' function, which represents the glider's dynamics. The state variables and their derivatives are interpolated at the midpoints of the discretization intervals. The equality constraints ensure that the state equations hold at the midpoints by comparing the system's dynamics at the midpoint (systmidm) to the approximated dynamics (Xdotmidm).

objfun.m

The 'objfun' defines the objective function for the optimization of the glider's flight.

The function is used by 'fmincon' and returns the value of the objective function at a given point. Since fmincon minimizes the objective function, the negative sign is used so that maximizing the horizontal distance x at the end of the flight corresponds to minimizing the negative value.

dy.m

This function 'dy' defines the state equations for the glider, describing its dynamics during flight. The physical parameters of the glider, such as mass, gravitational acceleration, air density, drag coefficients, and reference area S are defined. The function accepts the state variables X as input the horizontal position, altitude, velocity and flight path angle. The function calculates the rate of change of the state variables using the state equations. The function returns the derivatives of the state variables, which describe how the position, altitude, velocity, and flight path angle change over time.

4.4.3 Effects of scaling

Running the Optimization Without Scaling:

When the scaling factors are set to 1, it takes a higher iteration count to converge. For example, at the first discretization, there are 33 iterations as compared to the scaled factors which had 21 iterations. The iteration counts at the successive discretization points were also higher than those of the scaled factors. Thus the absence of scaling results in slower convergence.

Regarding the quality of the solution, the optimizer found a feasible point that satisfies the constraints, but it did not fully satisfy the optimality conditions. In other words, the optimization process found a local minimum but did not achieve the best possible solution. Even though the optimization did not fully converge to an optimal solution, the constraints were met. So the solution is feasible, but it's likely not the best possible. Also, the optimization process stopped because the step size became very small.

Running the Optimization With Scaling:

Using the original scaling factors (sc = [150, 50, 20, 1, 30]) results in fewer iteration counts leading to faster convergence.

The optimization successfully found a local minimum that satisfies all constraints.

The results of the optimization confirms that the objective function is non-decreasing in all feasible directions. This means the optimizer could not find any feasible direction to further reduce the objective function value, therefore the algorithm reached the optimal solution.
The result also shows that the optimization met the optimality tolerance, meaning the solution meets the accuracy requirements for being considered optimal.
The constraints were satisfied within the given tolerance limits, meaning the solution is feasible and respects the problem's boundaries.
Unlike the case without scaling where the optimality criteria were not met, here the optimization converged successfully. This demonstrates that scaling helped the optimization algorithm navigate the solution space more effectively, leading to a valid local minimum that satisfies both the objective function and constraints. The optimizer completed the process without issues related to step size or constraints suggesting that scaling resolved the difficulties observed in the unscaled run, ensuring smoother and more reliable convergence.

Importance of Scaling

Different variables may have vastly different ranges. This can lead to numerical instability during optimization, causing difficulties in convergence. Variables should be scaled so that they have similar magnitudes.

When variables are on a similar scale, optimization algorithms can navigate the solution space more effectively, leading to faster convergence.

Many optimization algorithms, such as Sequential Quadratic Programming (SQP), perform better when the input data is normalized as it helps the algorithm to compute gradients and other derivatives accurately.

4.4.4 Optimal solution

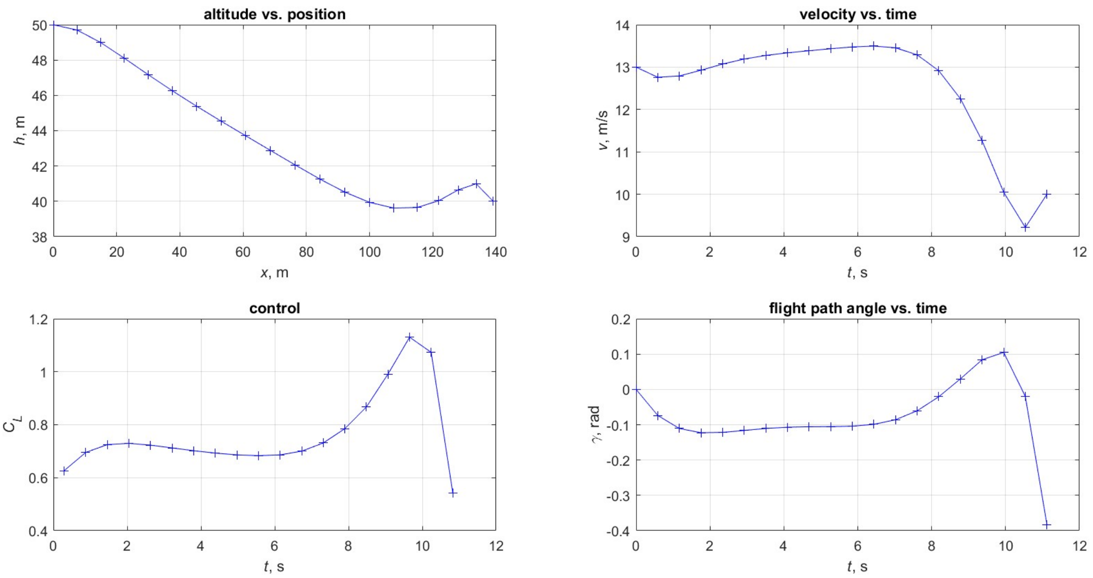


Figure 6: Solution of Dynamic Optimization Problem

From the resulting optimal solution, it can be seen that the greatest increase along the x-axis is **139.07 meters** for a drop of 10m in altitude.

4.4.5 Comparison with static problem

It is observed that the glider travels a significantly longer distance in the dynamic optimization scenario compared to the static optimization,

In dynamic optimization, the glider's control is varied throughout the flight. This allows the optimizer to adapt to the flight path and control settings based on the current state (altitude, velocity, flight path angle, etc.). As conditions change (e.g., altitude decreases, speed fluctuates), the control is adjusted dynamically to maximize the glide distance at each stage, resulting in a more efficient trajectory.

In static optimization, the control variable is constant throughout the flight. This means the glider uses the same lift coefficient and trajectory, regardless of changing flight conditions.

While the static approach finds a reasonable control setting based on a simplified analysis (maximizing the lift-to-drag ratio at a single point), it cannot react to changes during the flight, leading to a less efficient overall performance.

Dynamic optimization solves for the optimal control at every point in time, allowing the optimizer to find the best combination of altitude, velocity, and flight path angle at each stage of the flight. This flexibility leads to a longer distance traveled, as the flight is fine-tuned dynamically.

The dynamic trajectory is likely more optimized in terms of the flight path angle and velocity management. For example, the optimizer might increase or decrease the lift coefficient to minimize drag during certain phases of the flight, which is something the static optimization cannot do as effectively with a fixed control.

4.4.6 Indirect multiple shooting

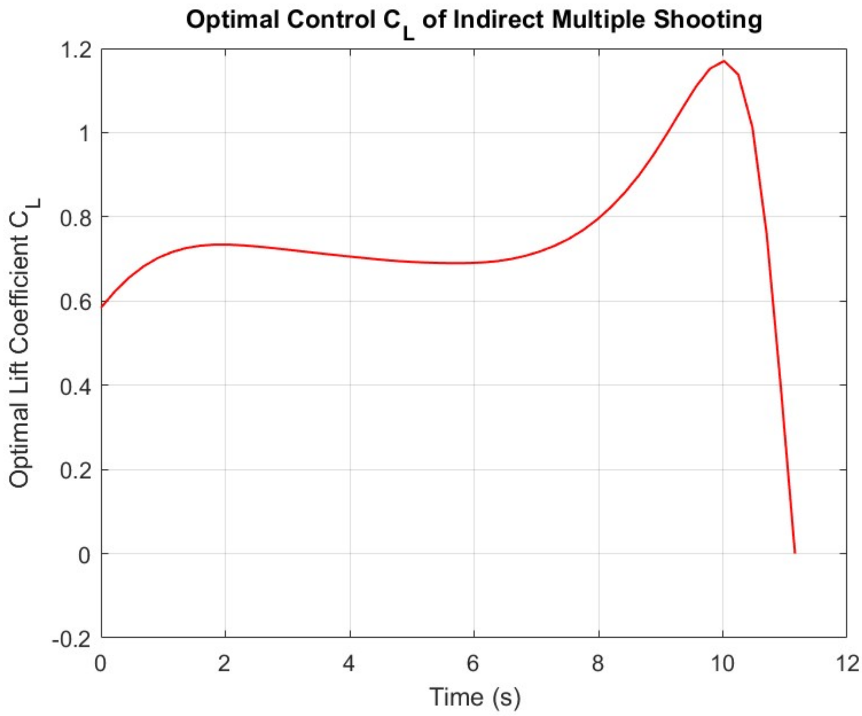


Figure 7: Optimal solution found by indirect multiple shooting, similar to the optimal control obtained by the direct method.

4.5 Glider Flight in Thermal Conditions

4.5.1 State equations when u(x) = 0

Given total velocity relative to the air is

$$v_r = \sqrt{v_x^2 + (v_h - u(x))^2}$$

and

$$\eta = \arctan\left(\frac{v_h - u(x)}{v_x}\right)$$

when u(x) = 0, these equations become

$$v_r = \sqrt{v_x^2 + v_h^2} = V \tag{i}$$

and

$$\eta = \arctan\left(\frac{v_h}{v_x}\right) = \gamma \tag{ii}$$

Now we take each of these equations and find their time derivatives.

For (i),

$$V = \sqrt{v_x^2 + v_h^2} = (v_x^2 + v_h^2)^{1/2}$$

$$\frac{d}{dt}V = \frac{1}{2}(v_x^2 + v_h^2)^{-1/2} \cdot \frac{d}{dt}(v_x^2 + v_h^2)$$

Using chain rule and the relation $\dot{v_x} = \frac{d}{dt}v_x$,

$$\frac{d}{dt}V = \dot{V} = \frac{1}{2\sqrt{v_x^2 + v_h^2}} \cdot (2v_x\dot{v_x} + 2v_h\dot{v_h})$$

$$\dot{V} = \frac{v_x\dot{v_x} + v_h\dot{v_h}}{\sqrt{v_x^2 + v_h^2}}$$

$$\dot{V} = \frac{v_x\dot{v_x} + v_h\dot{v_h}}{V}$$

Now substituting the equations of $\dot{v_x}$ and $\dot{v_h}$ where u(x)=0 is taken into consideration, we get the following relation

$$\dot{V} = \frac{\frac{v_x}{m}(-L\sin\gamma - D\cos\gamma) + \frac{v_h}{m}(L\cos\gamma - D\sin\gamma - mg)}{V}$$

$$= \frac{-D(v_x\cos\gamma + v_h\sin\gamma) + L(v_h\cos\gamma - v_x\sin\gamma) - v_h mg}{mV}$$

Now since $v_x = V\cos\gamma$ and $v_h = V\sin\gamma$, we have that $L(v_h\cos\gamma - v_x\sin\gamma) = L(V\sin\gamma\cos\gamma - V\cos\gamma\ \sin\gamma = 0$.

$$\dot{V} = \frac{-D(V\cos^2\gamma + V\sin^2\gamma) - mgV\sin\gamma}{mV}$$

$$= \frac{-DV - mgV\sin\gamma}{mV}$$

$$= \frac{-D - mg\sin\gamma}{m}$$

$$= \frac{-D}{m} - g\sin\gamma$$

This is earlier state equation for velocity.

For (ii),

$$\gamma = \arctan\left(\frac{v_h}{v_x}\right)$$

$$\frac{d}{dt}\gamma = \frac{1}{1 + (\frac{v_h}{v_x})^2} \cdot \frac{d}{dt}(\frac{v_h}{v_x})$$

$$\frac{d}{dt}\gamma = \frac{v_x^2}{v_x^2 + v_h^2} \cdot \frac{v_x\dot{v_h} - v_h\dot{v_x}}{v_x^2}$$

$$\dot{\gamma} = \frac{v_x\dot{v_h} - v_h\dot{v_x}}{V^2}$$

Now substituting the equations of $\dot{v_x}$ and $\dot{v_h}$ where u(x)=0 is taken into consideration, and also the equations of v_x and v_h we get the following relation

$$\dot{\gamma} = \frac{LV\cos^2\gamma - DV\cos\gamma\sin\gamma - mgV\cos\gamma + LV\sin^2\gamma + DV\sin\gamma\cos\gamma}{mV^2}$$

$$= \frac{LV - mgV\cos\gamma}{mV^2}$$

$$= \frac{L - mg\cos\gamma}{mV}$$

$$= \frac{L}{mV} - \frac{g\cos\gamma}{V}$$

This is earlier state equation for the flight path angle.

4.5.2 Optimal solution

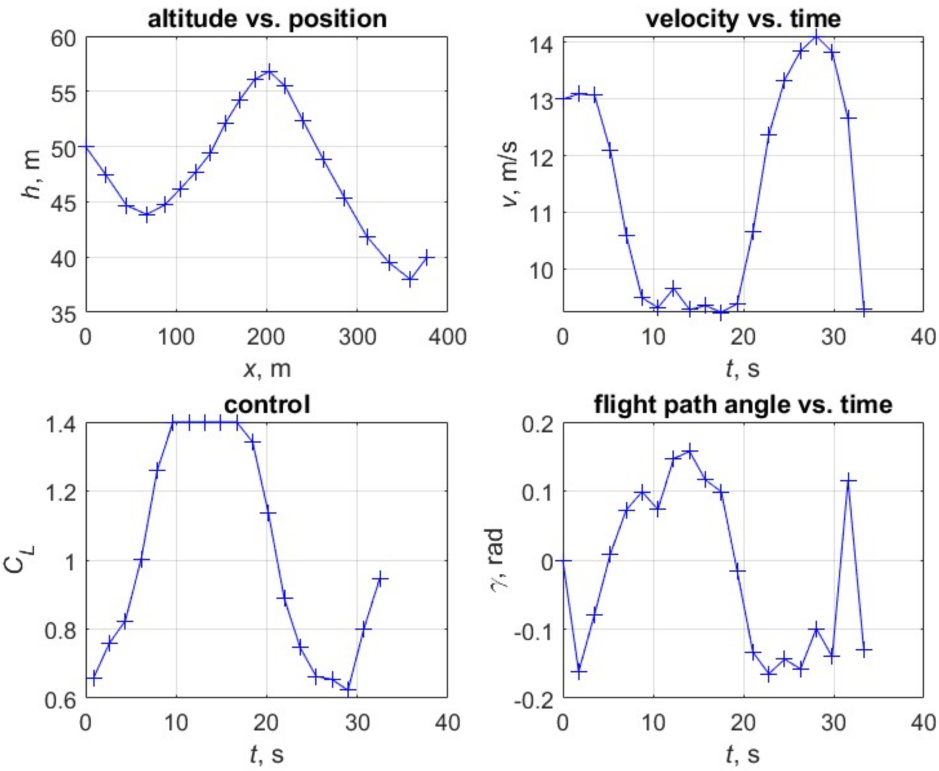


Figure 8: Optimal flight of glider in thermal conditions

4.5.3 Interpretation of results

The glider traveled a distance of 376.86 meters, which is a substantial glide distance. This indicates that the thermal updrafts likely helped extend the glider's range significantly compared to what it might achieve in windless conditions (where distances were lower).

The total flight time was 33.34 seconds, suggesting that the glider maintained an efficient trajectory for a relatively long period of time in thermal conditions, maximizing the glide duration and distance.

5 Summary and Conclusions

The glider optimization problem was addressed using direct collocation in combination with sequential quadratic programming (SQP). The glider's flight was modeled as a dynamic system of first-order differential equations, capturing the key state variables: horizontal position, altitude, velocity, and flight path angle. The control variable in the system was the lift coefficient, which was adjusted to optimize the glider's trajectory.

In this dynamic optimization framework, the terminal time was left free, and the objective was to maximize the horizontal distance traveled by the glider. The optimization was subject to constraints on altitude, velocity, and the lift coefficient. The time discretization of the problem transformed the continuous-time dynamic system into a finite-dimensional optimization problem, solved using nonlinear programming. This method falls under the category of direct methods, where the state equations are required to hold only at specific discretization points.

In this case, direct collocation was chosen, approximating the state variables using third-degree piecewise polynomials and assuming the control variables to be piecewise linear. This approach ensures that the state and control trajectories remain continuous, smooth, and satisfy the state equations at the midpoints of the discretized segments, leading to an accurate and efficient solution.

The Hamiltonian function played a crucial role in deriving the necessary conditions for optimality. The Hamiltonian is constructed from the state equations and the associated costates (or Lagrange multipliers). In this optimization framework, the costates were introduced to capture the sensitivity of the objective function to the state variables, providing a deeper insight into the control problem.

By taking the derivative of the Hamiltonian with respect to the control variable, the necessary conditions for optimality were derived. These conditions dictated the optimal lift coefficient trajectory, ensuring that the glider maximized the horizontal distance while respecting the constraints. The costate equations were solved alongside the state equations to iteratively refine the solution.

The optimization process iteratively improved the solution by starting with a coarse time grid and gradually increasing the number of discretization points. This continuation method ensured that the problem was solved efficiently without being computationally overwhelming. The method successfully found an optimal solution in both windless and thermal conditions. In the presence of thermals, the upward airflow was modeled as a function of the horizontal position, adding complexity to the dynamics. The results clearly demonstrated that leveraging thermal updrafts could significantly enhance the glider's range compared to windless conditions.

The use of direct collocation with SQP provided an effective solution to the complex dynamic optimization problem. The state and control variables were modeled with a high degree of accuracy, and the discretization of the problem allowed for smooth and continuous flight trajectories. By continuously adjusting the lift coefficient, the glider was able to maintain an efficient trajectory, avoiding stalling and maximizing the horizontal distance traveled.