Flight path optimisation of a solar powered plane.

Bob Vergauwen & Moritz Wolter

December 19, 2014

1 Introduction

In this project we are going to find the optimal path for a solar powered airplane [1]. During flight the plane's electrical engines drain power from it's battery, while at the same time solar panels on the plane's wings convert energy from the sun's rays into electric energy. However clouds may block the sun's rays from reaching the panels. In addition the sun's intensity increases when flying closer to the equator. In order to keep as much energy in the batteries as possible the flight path has to be optimized.

In this paper we will start by explaining the mathematical parameters we have taken into count to calculate the optimal path. Afterwards we will formulate the problem in it's standard form. We will finish by presenting some of the results obtained by the optimisation.

1.1 Formulation of the problem.

The main goal of this project is to find the best path connecting the starting point of the plane to the destination. Before this question can be answered we first have to define what we mean by the best path. We could for instance look at the shortest path, the most sunny path, the fastest path, ... A logical choice to make is to define the best path as the path that yields the most energy, at the end of the flight.

To calculate this energy we take several parameters in to account like the solar energy, the drag force on the plane and the cost for accelerating the plane. The combinations of the energy losses and gains will be combined to calculate the best path. Next we will give a brief explanation about all of these parameters and their mathematical formulation.

1.1.1 Solar energy

The solar gain is roughly defined as the amount fo sun than can be picked up trough the flight. The local amount of sun is determined by two factors, the angle of the sun at that time and that place and the local cloud density.

However the angle of the sun consists of declination and hour angle. For the declination θ of the sun, which looks at the sun's position north south of the equator we use the approximation formula:

$$\theta = 23.45 * sin(360/365 * (d - 81) * pi/180); \tag{1}$$

[1] Like the one from `http://www.solarimpulse.com/`

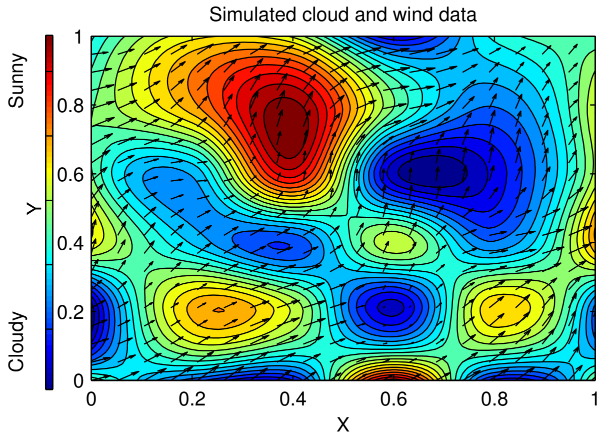


Figure 1: An example of simulated weather data. The colour map represents the intensity of the sun, the arrows point in the direction of the wind. This data is created by the interpolation on a 5 by 5 random generated matrix.

With d denoting the number of days since the beginning of the year. The other important angle we considered is the hour angle of the sun with depends on the time of the day. However as we are going keep the weather and sun data constant during the flight and we are assuming at flight at noon the hour angle is constant 0.

To simulate clouds we where looking for highly autocorrelated random data so that the clouds would be located in islands rather than being scattered without any pattern in the sky. The technique we applied was to generate a low dimensional random matrix and extrapolate the data points to get a continuous grid. An example of the simulated weather is shown in Figure 1. The contour lines on this plot represent the cloud densities and the arrows represent the local wind which is generated using the same technique.

At the end the local solar gain is eventually calculated by the product of the suns intensity related to the angle and the amount of clouds.

$$E_{sun}(x,y) = V_{sun}(x,y)V_{Cloud}(x,y) \tag{2}$$

The total amount of energy collected by the airplane is now given by

$$E_{sun}(t) = -\alpha_{sun} \oint_{path} E_{sun}(x,y)d\tau, \tag{3}$$

this path integral is evaluated over the path from time $\tau = 0$ until time $\tau = t$. The constant term in front of the integral is a scaling term that can be adjusted to tune the weight of the individual energies. Remark the minus sing in front of the integral, this implies that we want to maximise the solar gain.

1.1.2 Drag resistance

A second import aspect that influences the energy balance of the airplane is the drag force. Introducing a drag force will keep the velocity of the plane bounded. We chose to use a simple quadratic dependency of the drag force and the speed. This drag force is then defined as,

$$E_{drag}(t) = \alpha_{drag} \oint_{path} v(x,y)^2 d\tau. \tag{4}$$

In this integral the function $v(x,y)$ represents the speed of the airplane, all the same conventions for the integral apply as noted above.

1.1.3 Acceleration force

The energy needed to accelerate the airplane is the last parameter we looked at. We did assume there was now energy needed to decelerate the plane which is a realistic approximation for small airplanes. Ta calculated the acceleration energy we used

$$E_{accel}(t) = \alpha_{accel} \oint_{path} s(x,y)^2 d\tau \tag{5}$$

In this integral the function $s(x,y)$ is defined as

$$s(x,y) = \begin{cases} a(x,y) & \text{if } a(x,y) \geq 0 \\ 0 & \text{if } a(x,y) < 0 \end{cases} \tag{6}$$

Where the function $a(x,y)$ is the acceleration of the airplane at each point on the path. The point of integrating the function $s(x,y)$ is just to get rid of the contribution of the negative acceleration.

2 The optimisation problem

In this section we will formulate the optimisation problem in a formal way.

2.1 Parameters

In the previous section we talked a lot about the path of the airplane but we did not mention how we are going to define such a path. The path is the thing we want to see optimised, as a result of this we take it to be the input of our optimisation algorithm. To define the path we used a m by 3 matrix,

$$\mathbf{P} = [\mathbf{X}, \mathbf{Y}, \mathbf{t}] \, . \tag{7}$$

typically the value of m was of the order of 30, so we had a 90 dimensional input space. The first column of the matrix $\mathbf{X}$ represents the x values of the trajectory, the second column $\mathbf{Y}$ represents the y coordinates and the third column $\mathbf{t}$ represents the time steps. Using these three columns it is possible to calculate all the flight aspects of the airplane, for instance its speed or its acceleration.

An other possibility was to exclude the time vector and assume that the time taken in every step was constant. To compensate for different step lengths along the path the speed of the plane had to be adjusted in every step. We tried both models but found the first model to be more realistic. In addition it gave better results.

2.2 Optimal solution

As mentioned above, the optimal solution was defined to be the path for which the maximum amount of energy remained in the batteries at the end of the flight. This residual energy is defined as:

$$E(\mathbf{P}) = \oint_{path} \alpha_{drag} v(x,y)^2 + \alpha_{accel} s(x,y)^2 - \alpha_{sun} E_{sun}(x,y) d\tau. \qquad (8)$$

This integral has to be evaluated over the whole path to get the energy at the end. As can be seen this function has no explicit solution so it will not be possible to determine the order of the problem and we will have to use a non-linear solver to find the maximum.

2.3 Formulation

Let us now put everything together to formulate the actual optimisation problem. In oder to numerically solve this problem we have to slice the flight time into discrete parts dt. With x and y denoting of the airplane as before we have:

$$\begin{aligned}
\min_{\mathbf{P}}(-E([\mathbf{X},\mathbf{Y},\mathbf{t}])) \quad \text{s. t.} & \\
& x \in [0,1]^m && |\forall x \in \mathbf{X} \\
& y \in [0,1]^m && |\forall y \in \mathbf{Y} \\
& [\mathbf{X}[0], \mathbf{Y}[0]] = [x_0, y_0] && \\
& [\mathbf{X}[m], \mathbf{Y}[m]] = [x_m, y_m] && \\
& dt > 0 && |\forall dt \in \mathbf{t} \\
& \sum_{i=1}^{m} \mathbf{t}[i] = 30. &&
\end{aligned}$$

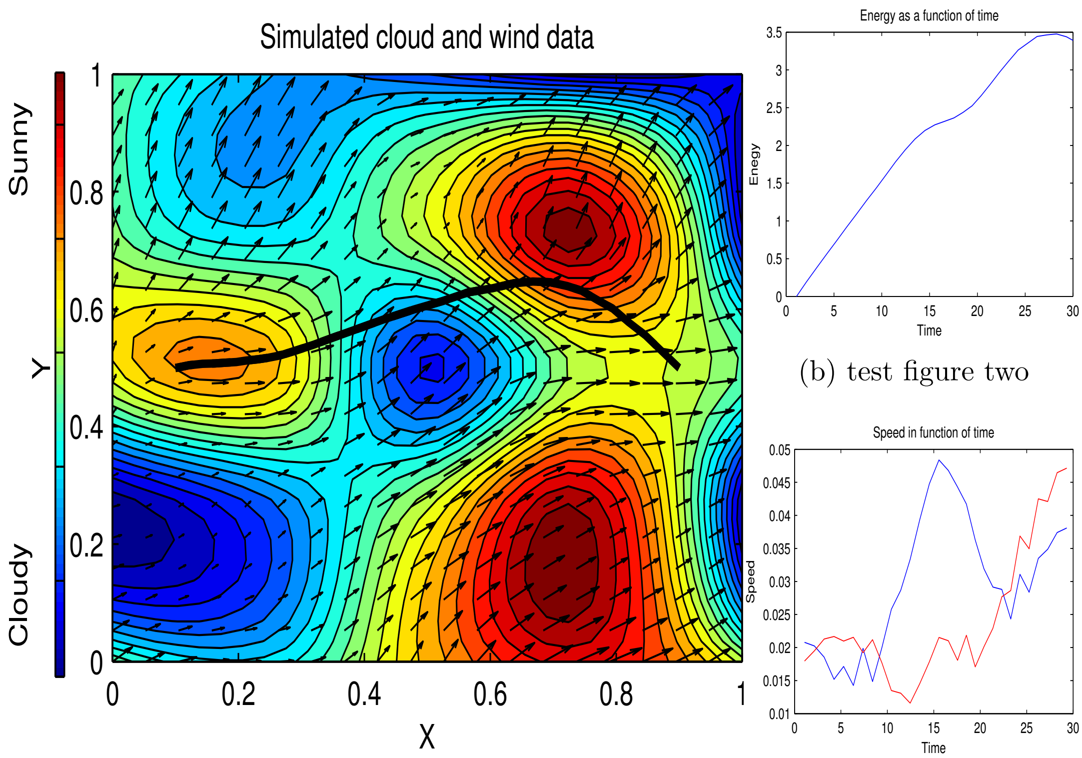
All these boundary conditions arise very naturally. The boundary conditions concerned about $\mathbf{X}$ and $\mathbf{Y}$ come from the fact that the plane has to stay in the zone where we know the weather and they fix the starting and the ending locations of the plane. The boundary conditions for the time insures that the all the timesteps done by the plane are positive and fix the arrival time at 30. Having formalized our problem we can proceed to solving it.

3 Results

In this section we will take a look at the results of the optimisation problem. We look at two differed cloud set-ups for the clouds and wind, a first one will be a system with one big cloud in the center. the second one will have a collection of several smaller clouds.

4 A first setup

The results returned by the optimization algorithm for the first set-up are shown in Figure 2. On this Figure 3 plots are given, one of the actual path, one of the energy stored in the batteries and one of the relative and absolute speed of the plane. We observe that the flight path nicely avoids the clouds in the centre, while at the same time the plane flies into the sunny area, to charge its batteries. In addition the path starts at the start point and ends at the endpoint. A condition we fed into the solver initially. When taking a closer look at the speed and sun gain plots we observe, that the plane speeds up initially and slows down when it reaches the sunny region that is close to it's destination. A behaviour that we would expect from a good solver, since it leads to higher battery levels if the plane spends more time in the sunny regions.



(a) test figure one

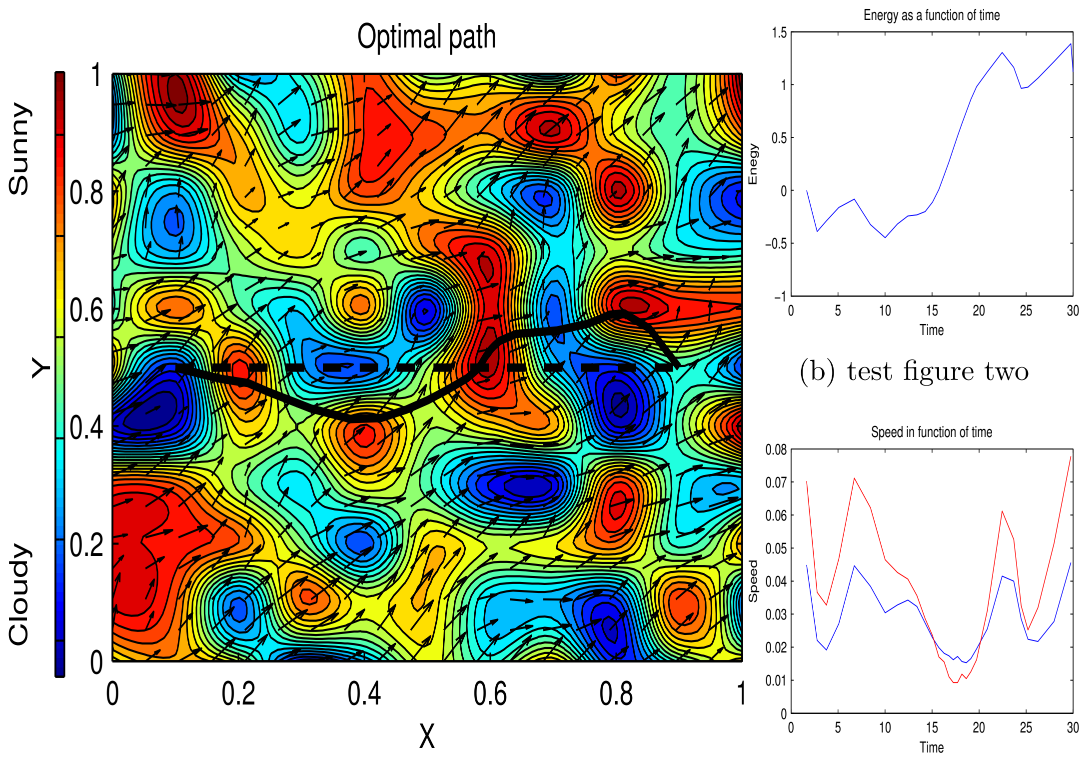
(b) test figure two

(c) test figure three

Figure 2: The left plot shows the optimized path for our optimization problem. The speed of the airplane (blue) and the relative speed (red) are shown in the bottom right position. In top the right position we show the amount of energy that's stored in the plane's solar batteries as a function of time. For this setup this function is nearly everywhere increasing.

5 A second setup

As a second set-up we tried an other set-up for the weather just to look at how the algorithm performers on a complexer problem. The result of this calculation is given in Figure 3. Looking at the plots it may look that this solution is not the optimal one, the speed is stil a bid bumpy, especially around time 15. To calculate this solution we ran let the algorithm run for about an hour, it stopped after 15000 iterations. The flat spots in the curves is due to to the big steps it is taking at uninteresting places. When the plane has to manoeuvre very carefully the stepsize gets smaller, this is clearly seen around time 15.



(a) test figure one

(b) test figure two

(c) test figure three

Figure 3: The left plot shows the optimized path for our optimization problem. The speed of the airplane (blue) and the relative speed (red) are shown in the bottom right position. In top the right position we show the amount of energy that's converted in the plane's solar cells as a function of time. The bottom left shows the acceleration as a function of time. The bottom middle plot depicts the energy level in the battery throughout the flight. Finally the bottom right plot depicts the derivative of the cost function.

6 Conclusion

A lot of assumptions have been made throughout the creation of the model. Most of these are oversimplifications of the real world and we barley scratched the surface of the real optimisation problem behind the airplane. We only optimised the residual energy at the end of the flight, but did not enforce conditions on the energy levels of the air plane, it makes sense that the energy level have to stay positive at all the time during the flight. The same applies to the speed of the plane. In order to prevent the plane from reaching dangerously low or high speeds, constraints should be enforced here as well. Possibly we can make the weather more realistic and time varying as well, this could for instance be done by letting the clouds move with the wind. If we did so than $\mathbf{t}[0]$ would no longer be a fixed condition. Which would result in an extra freedom to chose the best time of departure. The next logical extension for a model of an airplane is to include the flight altitude as well. Since at higher altitudes the solar panels of the plane work more efficiently and the clouds will be less. Introducing the flight altitude will come to the cost of extending the dimensionality of the problem by an other column vector of m values.