FLIGHT PATH OPTIMISATION

BOB VERGAUWEN AND MORITZ WOLTER

Project Report

Supervised by Prof. dr. ir. Toon van Waterschoot
December 15, 2014 – Version 1

1 MOTIVATION AND IMPLEMENTATION

1.1 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 suns intensity increases when flying closer to the earth's equator. In order to keep as much energy in the planes batteries as possible the flight path has to be optimized.

1.2 PROBLEM FORMULATION AND ANALYSIS

1.2.1 *Time invariant case*

In the optimization process we use randomly generated weather data and sun data corresponding to a flight between 48° and 50° latitude. We generate the weather data by cubicly interpolating a random matrix of size `n` over a grid. The larger `n` becomes the more complex the weather will be. This functionality is implemented in `Static_Weather.m` [2] The class returns a weather data matrix W. Data for the sun comes from a couple of established equations that relate the local time and position to the sun's intensity which happens in `sun.m`. The class returns a sun data Matrix S. We define low values in the weather Data Matrix as cloudy and high values as sunny weather. As clouds reduce the electric yield of our plane's solar cells we define the overall intensity as:

$$I = W.*S \tag{1}$$

While following a convention where $.*$ denotes element wise multiplication we end up with an intensity matrix I, which contains sun intensity data weighted according to the random weather.

Next we took a closer look at the airplane. Here we consider the degree to which the plane can recharge its battery during flight, which is the path integral over the intensity data:

$$E(x) = \oint_{\Gamma} I \, dx. \tag{2}$$

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

2 All matlab code can be found on https://github.com/double2double/OptimisationProject/ in the MATLAB folder.

With dx being the length of each step. Having found the additional energy we are able to convert during flight we now look at losses, which are due to drag and acceleration. We define acceleration cost in the time invariant case as the scalar product of the vectors containing local accelerations:

$$A(\mathbf{x}) = \mathbf{a}^T \cdot \mathbf{a} \tag{3}$$

To calculate the derivatives in the time invariant case we approximate `dt` *as the total time over distance traveled.*

However we only count cases where the acceleration is positive since deceleration does not drain battery power. Finally we consider the drag force which the plain experiences on its path, which we approximate as the dot product of the local speed vector:

$$D(\mathbf{x}) = \mathbf{s}^T \cdot \mathbf{s} \tag{4}$$

Having investigated the approximate battery power lost and gained on a specific path, we can now proceed to formulating a cost function for the time invariant case.

1.3 COST FUNCTION FORMULATION

1.3.1 *Time invariant case*

In the time invariant case we define the optimal path as the path from start to destination where we loose as little battery power as possible. The problem parameter is a path vector $\mathbf{x} \in \mathbb{R}^m$. Which contains m points on the path. Introducing weights for the different power gains and losses we define the cost function:

$$V(\mathbf{x}) = -E(\mathbf{x}) + 0.01A(\mathbf{x}) + 200D(\mathbf{x}). \tag{5}$$

Which is subjected to the constraint that the path my not leave the weather data grid. This translates into $0 \leqslant x \leqslant 1$ and $0 \leqslant y \leqslant 1$ for the elements of $\mathbf{x} = (x\ y)^T$. We solved this problem starting with the straight line from (0.1,0.5) to (0.9,0.5) as initial input using matlab's `fmincon` solver. This const function is implemented in the `Airplane.m` class file.

2 RESULTS

2.1 TIME INVARIANT CASE

The optimized path (shown in red) for the time invariant case is shown in figure 1. We observe that the path leads nicely trough sunny weather grids including a detour to the sunniest spot on the grid at (0.5,0.35), indicating that our problem is well formulated.

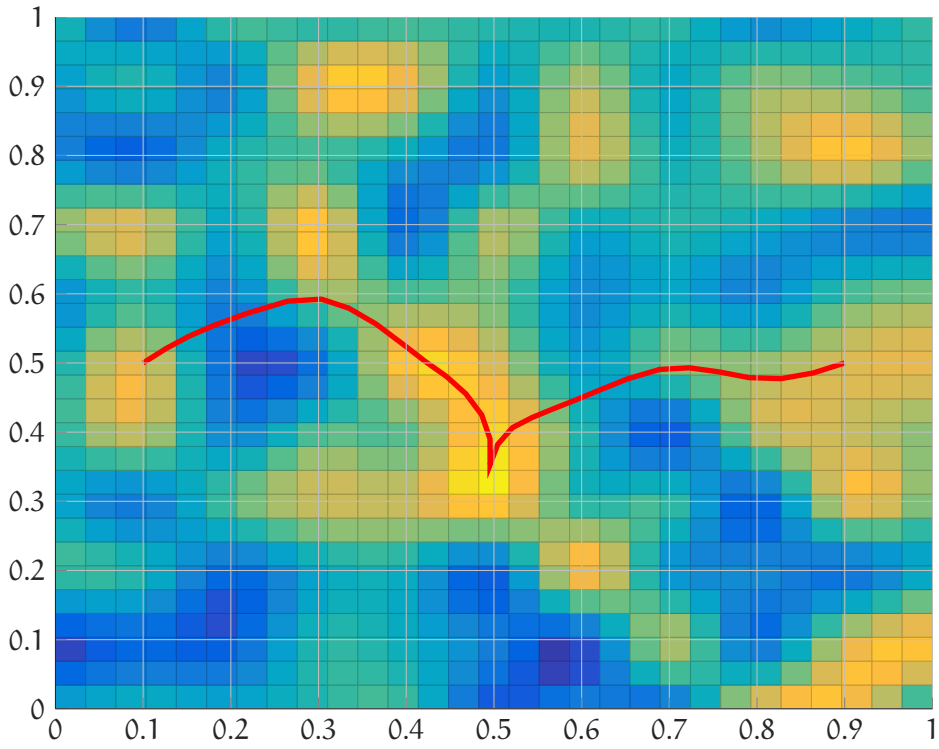

Figure 1: Optimized flight path from the coordinates 0.1,0.5 to 0.9,0.5, with different weather data.