Department of Electrical
and
Computer Systems Engineering

Technical Report
MECSE-7-2006

UAV Ridge Soaring in an Unknown Environment

A.K. Goodwin, G.K. Egan and F. Crusca

MONASH
UNIVERSITY

UAV Ridge Soaring in an Unknown Environment

A.K. Goodwin, G.K. Egan and F. Crusca
Department of Electrical & Computer Systems Engineering
Monash University 3800
Melbourne, Australia

Abstract—**This paper describes a simulation environment and an initial investigation of a method of providing turn direction advice to an Unmanned Air Vehicle in an attempt to soar upon any vertical air created by a ridge or rise in the terrain. The exact layout of the terrain is unknown to the adviser. The aim of providing this advice is to increase the time aloft.**

***Index Terms*—autonomous flight, ridge soaring, UAV, unmanned aircraft.**

I. Introduction

The aim of this project was to enable a UAV to increase the amount of time it spends aloft by taking advantage of any lift or rising air that occurs naturally due to the terrain – in particular, ridge lift, which is formed when wind is deflected upwards by hills, mountains, and etcetera. By using the rising air, a UAV is less reliant on its own motor, extending the life of the batteries and increasing the possible flight time. For the project, development of a software guidance system was required, which, with no prior knowledge of the layout of the terrain, is to give advice to the (auto) pilot as to which direction of travel is more ideal.

The guidance system receives information from the UAV's onboard instrumentation – GPS, altimeter, etc. – and overtime builds a map of the airflow conditions. The information is stored and retrieved using location as the index. The purpose of remembering this information is to allow the guidance system to evaluate the possible amount of lift in one area versus another.

II. Software Design

The software written for this project is divided into two components:

- The Simulation System; and,
- The Advice System.

The Simulation System exists to replicate the real-world environment a UAV would experience. The Advice System uses information supplied by the Simulation System to complete the project objectives; namely provide advice on ideal travel directions.

The software is deliberately designed to be highly structured. Functions that belong to the same logical group are contained in the same file to provide easy navigation. This method allows easy debugging and exposes the logical structure of the software.

III. Simulation System

While there exist a number of pubic domain flight simulators FlightGear [1] and Slope Soaring Simulator (SSS) [2] which may be used for extensions to this research but for the initial study it was decided to implement a simple simulated environment within which to test the Advice Systems and its algorithms. The simulator uses code for determining the shortest distance from a point to a line [3].

The Simulation System provides a stylised 3-dimensional image to allow the UAV to be observed. The terrain of the simulated world is read in from a text file and generated dynamically. The use of a text file allows the terrain to be edited simply and quickly, without the need to use a special program. Many of the properties of the simulation, such as wind strength and direction, are specified in the same text file as the terrain.

A. Terrain & the Ridge Model

Highly accurate three-dimensional models of the behaviour of air as it flows over a ridge are complex and involve many parameters. Reference [4] tells as that the direction of airflow at a location depends on:

- Wind velocity;
- Static stability of the air;
- Terrain shape.

Time is also involved; as the first two of the above parameters are not constant (it could be argued the third is also not constant).

Fig. 1, reproduced from [5], shows possible paths taken as air travels over a ridge.

In order to model this behaviour, the following simplifications were made:

- At a certain elevation above a ridge, the effect of the ridge becomes negligible. The elevation at which this occurred was termed the "neutral elevation".

- The static velocity of the wind is constant with respect to altitude. In actual fact, air travels more slowly along the ground than above it.

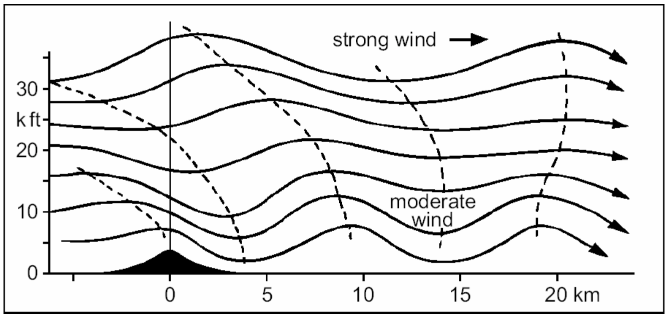

Fig. 1. An example of possible airflows as wind travels over a ridge (reproduced from [5]).

Fig. 2 shows the cross-sectional view of the proposed ridge. It can be seen how the ridge deflects the air at lower altitudes most severely, whilst the air nearer the neutral elevation is less affected.

Along the plateau of the ridge, the airstreams show that the density of air has increased – in the figure it approximately triples. This will have an effect on the speed at which the air travels. If we consider the ridge as analogous to a narrowing in a pipe, the effect can be determined. When a unit volume of air enters the wide end of a pipe, it must displace the same volume of air out of the narrow end. The equation of continuity states [6]:

$$\begin{aligned} &\Delta Volume = A_1 v_1 \Delta t = A_2 v_2 \Delta t = 0 \\ &\therefore A_1 v_1 = A_2 v_2 . \end{aligned} \tag{1}$$

Thus the speed at which the air travels increases as the cross-sectional area of the pipe decreases. The same applies for the ridge. As the same volume of air is compressed as it traverses the plateau, its speed increases as a ratio of the neutral elevation and the plateau height.

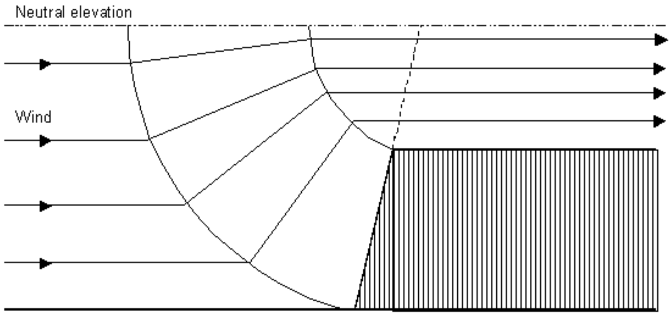

Fig. 2. A cross-sectional view of the ridge model used for simulation.

As the air is transitioning from unaffected (before the ridge) to being uniformly compressed by the plateau, it passes through non-uniform compression which has been modelled as linear. That is, the amount of compression (and hence increase in speed) changes linearly from uncompressed to fully compressed.

The model of the ridge includes a configurable valley. The air flows over the ridge, is compressed along the plateau and then spreads out again in the valley. The effect can be seen in Fig. 2, if you consider the wind as travelling in the opposite direction.

Using the advanced ridge model, the user can specify the position and parameters of any number of ridges they wish to include in the terrain. The program reads the configuration file and successively adds each ridge into the otherwise flat terrain. The terrain is divided into a grid of cells, with each cell being further subdivided. The vertices of the subdivision store height information of the terrain for that point. The division into cells allows the program to determine which cells are visible on the display and which aren't. By identifying and rendering only those cells which can be seen, the processing power required to display a scene is greatly reduced. The process of selecting the visible cells is called "culling". The particular method used is "frustum-tested culling" and was written by [7].

B. *Wind Model*

The wind used for the simulation has been modelled as having a constant direction and a randomly varied strength. The average strength and the maximum percentage change of the wind are specified as part of the simulation. These parameters are set in the terrain description file.

C. *GPS Acquisition Delay*

GPS units by their very nature do not provide an instantaneous measurement of their position. When asked to give their position, there is a delay whilst the unit interrogates the signals from the satellites and triangulates its location. If the GPS unit is moved during this time (i.e. the UAV is moved), the result will be correct for the old position - not the new. This means the GPS is continually playing "catch up" on the actual position of the UAV. A typical value for the acquisition delay is 1 to 1.5 seconds. When the UAV is travelling at a cruise speed of 20 m/s, this can mean the position is incorrect by up to 30m.

IV. Advice System

In order to give advice, the system uses present and prior data obtained from the UAV instrumentation. The data is stored in a map that is indexed by position. Rather than store the map as a contiguous array, the rows are divided such that each row is an individual array. The map maintains pointers to these rows. The design choice was made in order to minimise the cost of moving the map, should the occasion arise. For example, if the locations the map covers are to be shifted vertically by one row, it is a simple operation to discard the out-of-range row, shift the pointers to the remaining rows and create a new row. The order of this procedure is $O(n)$ n being the number of locations. Had a contiguous array been used, it would be necessary to move every element of every row to its new position explicitly. The order of doing this is $O(n^2)$, a much more severe penalty. It is noted that horizontal moves still attract a cost of $O(n^2)$. Using a linked-list could reduce this; however this improvement must be balanced against the added complexity of element traversal. It was decided that there would be no benefit in using a linked-list map.

The information each element of the map holds is stored in a structure containing two fields: a field indicating whether the information is valid, and a field indicating the amount of lift at this point.

A. *Simple Turn Adviser*

The first turn adviser, used to test the development of the simulation system, operated by simply using the roll of the UAV. When the UAV enters vertically moving air, the aircraft will roll away from the most rapidly rising air. The adviser detects this roll and uses it to provide advice. Significantly, the Simple Turn Adviser does not record and therefore or reuse any information obtained from the UAV's instrumentation; it does not *learn.*

B. *Advanced Turn Adviser*

Unlike the Simple Adviser, the Advanced Turn Advice system relies solely upon remembering environmental conditions at specific locations. The system uses what it has learnt to direct the UAV. The basic action of the algorithm implemented is to search the map starting from the UAV's current position and move radially outward.

Based upon the cruise speed of the UAV, and the radius of its turning circle at that speed, the algorithm calculates the area of sky adjacent to the UAV that cannot be reached (through inability to turn tightly enough). Because this region of sky cannot be reached, the algorithm does not test it.

The algorithm divides the space around it into 30° sections and accumulates the lift from each search path into its accompanying section. Once all the paths have been processed, the centre angle of the section with the greatest accumulated lift is decided to be the most optimal angle of travel. Should all the sections have an aggregate lift of zero, then the algorithm will determine the section with the most negative lift and advise that the UAV should turn *away* from that section. If there are no sections whatsoever with any kind of lift, positive or negative, the algorithm advises the UAV to continue upon its current heading.

The Advanced Turn Adviser contains a "safety" option: should the UAV stray too far from the centre of the map, the turn adviser will instruct the UAV to head back towards the centre. The distance the UAV can travel from the centre was dubbed the Escape Distance, and is specified in the configuration file. It is usually set to a value large enough to allow the UAV to roam freely over the entire map but can be set to zero to disable it.

C. *Pre-advice Pattern Flight*

When the Advanced Advice System is called upon to operate, it can immediately attempt to give advice or can firstly fly a predefined pattern – the Pre-advice Pattern.

The purpose of instructing the UAV to fly the search pattern is to partially fill the Advice System's internal map of lift locations. Without this information, the system cannot give advanced advice.

The path the UAV follows when flying the search pattern is a clockwise decreasing spiral. The search pattern starts by following the outside of the area covered by the internal map, and proceeds to spiral inwards with user-definable spacings between successive spirals. The spacing can be changed in the simulation configuration file.

V. Performance of the Advice System

Two investigations were conducted when testing the system's performance. The first was to test the benefit of using the Pre-advice Pattern, the other to compare the Advice System to an unaided UAV.

The testing revealed that using the Pre-advice Pattern greatly improved the performance of the Advice System. In fact, it provided an approximate twofold increase in the amount of time the UAV spent in lift and threefold increase in the average lift per unit time. Separations of 50, 100 and 200 metres performed similarly, though the best separation was 100 metres.

The results of the second investigation are not as encouraging. The algorithm implemented performed as follows:

Average lift per unit time:	$0.5\ \text{ms}^{-1}$
Fraction of time spent in lift:	$0.28\ \text{s s}^{-1}$

This can be compared against an unaided system:

Average lift per unit time:	$0.33\ \text{ms}^{-1}$
Fraction of time spent in lift:	$0.35\ \text{s s}^{-1}$

Whilst the UAV tended to have a larger amount of lift, it spent less time in that lift. The project has provided a framework to allow us to rapidly and easily implement and test new advice algorithms.

VI. Extensions and Improvements

Some possible extensions to this work are:

- A more sophisticated ridge model, e.g. include rotor effects beyond the ridge line;
- Integration with FlightGear;
- The introduction of turbulence and random

variations in wind direction;

ACKNOWLEDGMENT

We wish to thank the Members of the Aerobotics Research Group at Monash University [8], Lawrence Goodwin, for his technical advice and practical experience in hang-gliding, and Jane Waugh.

REFERENCES

[1] "FlightGear Flight Simulator," [online] 2005, Available: http://www.flightgear.org (Accessed: 1/5/2005).

[2] D. Chapman, "Slope Soaring Simulator," [online] 2005, Available: http://www.rowlhouse.co.uk/sss/index.html (Accessed: 26/4/2005).

[3] D. Sunday, "About Lines and Distance to a Line," [online] 2005, Available: http://geometryalgorithms.com/Archive/algorithm_0102/algorithm_0102.htm (Accessed: 5/6/2005).

[4] T. Bradbury, "Wave Soaring over the British Isles (Some Theoretical Aspects and Practical Observations)," [online] 1984, Available: http://www.go.ednet.ns.ca/~larry/bsc/articles/wave/wavesoar.html (Accessed: 18/5/2005).

[5] T. Bradbury, "A Look at Wave Clouds," *Free Flight*, (6/92), pp. 6-10, 1992.

[6] D. Halliday, R. Resnick, J. Walker, *Fundamentals of physics*, 6th ed., New York: John Wiley & Sons, Inc., 2001.

[7] M. Morley, "Markmorley.Com | Frustum Culling in Opengl," [online] 2000. http://crownandcutlass.sourceforge.net/features/technicaldetails

[8] *Monash Aerobotics*, [online], Available: http://www.ctie.monash.edu.au/hargrave/aerobotics.html, 2002, (Accessed February 2006).