Tree-based Trajectory Planning to Exploit Atmospheric Energy

Jack W. Langelaan

The Pennsylvania State University, University Park, PA 16802, USA

Abstract— This paper presents an approach to planning long distance trajectories for small unmanned aerial vehicles. The feasibility of such long-distance flights is dependent on exploiting atmospheric energy, which is possible in regions where there is a strong vertical component of wind. Here a tree-based approach is used to find a feasible trajectory between the start position and a distant (i.e. beyond simple gliding range) goal. To improve computation time a set of branches is pre-computed from the space of allowable inputs allowing fast expansion of a node. Nodes are selected for expansion based on a weighted random approach. Simulation results show that feasible paths can be found using this method.

I. INTRODUCTION

A major limitation in developing practical small uninhabited aerial vehicles (UAV) is the energy required for long-range, long endurance operations. Large aircraft such as the Global Hawk can remain on-station for 24 hours and can fly non-stop from the continental United States to Australia. However, small and micro UAVs face severe limits on the fuel that can be carried, greatly reducing both endurance and range. In addition, the best L/D attainable for small and micro UAVs is typically much smaller than for larger aircraft because of the smaller Reynolds numbers. This further reduces performance.

Significant range and endurance improvements can be realized by obtaining energy (in the form of altitude or speed) from the surrounding atmosphere. Energy can be obtained from vertical air motion, from velocity gradients and from gusts. Vertical air motion has three main causes: uneven heating of the ground, which produces buoyant instabilities known as thermals; long period oscillations of the atmosphere, generally called wave; and orographic lift, where wind is deflected by the slopes of hills and mountains. Vertical air motion is a quasi-static phenomenon, and flight which exploits vertical air motion is known as soaring. Large birds such as eagles, hawks and condors as well as human sailplane and hang glider pilots routinely use soaring flight to remain aloft for many hours and traverse hundreds of kilometers without flapping wings or the use of engines.

A second means of extracting energy from the air uses velocity gradients (which can occur near the ground due to the boundary layer) or shear layers (which often occur on the leeward side of mountains and ridges). This strategy, called dynamic soaring, was first described by Lord Rayleigh in an analysis of albatross flight [1], [2]. Dynamic soaring is again becoming the subject of research both for recreational flight (mainly by RC flying enthusiasts) and for UAV flight. However, this class of dynamic soaring generally requires

highly agile flight in close proximity to the ground: this is a very risky endeavor.

The third means of extracting energy from the air exploits gusts. It has been observed that the flight performance of large birds is improved by gusts, while it is typically reduced on human-piloted aircraft [3]. This suggests that birds are able to extract energy from gusts, and indeed Kiceniuk reports that it is even possible to extract energy from a downward gust [4]! Extracting energy from gusts is complicated by their typically short duration, hence very fast response (typically exceeding human reaction time) is required. Control laws have been developed to enable energy extraction from gusts by small UAVs [5].

These three methods of extracting energy from the environment can be used to enable autonomous long duration, long distance flight (denoted $(LD)^2$ flight) by unmanned aerial vehicles. For the remainder of this paper these three modes of energy extraction will be referred to as static soaring, dynamic soaring and gust soaring, respectively.

The time scales of each of these modes of flight are very different. Static soaring occurs over time ranging from minutes to hours, dynamic soaring typically consists of a periodic trajectory with a duration of a few tens of seconds and gusts are very short duration (less than a few seconds). In a system which exploits all three modes of energy extraction this time scale separation can be used to treat each mode almost independently. Long-duration planning can be performed to exploit spatial variation in wind speed (both vertical and horizontal), shorter duration optimal trajectories can be designed for dynamic soaring and a closed-loop controller can be designed to exploit gusts.

The major focus of this paper is on the problem of static soaring by a small UAV, specifically on enabling both long endurance and long range flights using orographic (i.e. slope) lift. It describes a tree-based planning algorithm which uses a point mass model of the vehicle and knowledge of the wind field (this may be obtained from predictions generated using meteorological forecasting tools such as MM5 [6]) to generate feasible trajectories to a distant goal (i.e. a goal which is significantly beyond simple gliding range).

Sampling based planning methods [7] such as probabilistic roadmaps [8], [9] have become widespread. They have been used in both static [10] and dynamic [11] environments. Variants such as rapidly-exploring random trees (RRT, [12]) can account for higher order vehicle dynamics and have been used for blimp trajectory planning [13]. These approaches are probabilistically complete (i.e. if a feasible solution exists it will be found, given enough time) but there is no guarantee of

finding a solution within a specified time. Path planning for long-range trajectories in realistic wind fields using genetic algorithms is addressed by Rubio and Kragelund [14].

This paper describes a tree-based approach to planning trajectories which exploit atmospheric energy. Since most trajectories will lead to dead ends (i.e. the lack of rising air will make energy extraction impossible), methods to bias the expansion of the tree should be very effective in reducing the time and computation required for finding a feasible path. Biased searches have been proposed (for example, goal biasing [15] or finding narrow passages [16]). In this case the search will be biased along regions where energy can be obtained from the atmosphere.

The remainder of this paper is organized as follows. Section II discusses the dynamics and kinematics of flight, Section III describes the planning algorithm, including computation of branches, selection of nodes for expansion, culling infeasible branches and the termination criterion. Section IV presents results of a sample problem: flight to a distant (i.e. beyond simple gliding range) goal. Finally Section V presents concluding remarks.

II. Vehicle Dynamics and Kinematics

It is assumed that an on-board controller is able to follow heading, airspeed and throttle commands. Moreover, it is assumed that response to step changes in commands is very fast compared with the duration of a particular command. Hence a point mass model is sufficient to describe vehicle motion for planning purposes (Figure 1). Vehicle kinematics are given by

$$\dot{x} = v_a \cos\gamma \cos\psi + w_x \qquad (1)$$
$$\dot{y} = v_a \cos\gamma \sin\psi + w_y \qquad (2)$$
$$\dot{z} = v_a \sin\gamma + w_z \qquad (3)$$

where v_a is airspeed, ψ is heading and $\mathbf{w} = [w_x \ w_y \ w_z]^T$ is the 3D wind vector.

The glide path angle γ is a function of airspeed v_a and throttle setting T, and can be obtained for steady flight. Resolving forces parallel and perpendicular to the flight path,

$$mg\cos\gamma = L + T\sin\alpha_i \qquad (4)$$
$$mg\sin\gamma = D - T\cos\alpha_i \qquad (5)$$

where m is mass of the vehicle and α_i is the angle of incidence between the thrust axis and the flight path.

It is assumed that the flight path angle γ is small, hence $\sin\gamma \approx \gamma$ and $\cos\gamma \approx 1$. A further simplifying assumption (admittedly less accurate) is that thrust is always aligned with the flight path angle (i.e. α_i is zero). From Equation 4

$$mg = L = \frac{1}{2}\rho v_a^2 S C_L \qquad (6)$$

therefore

$$C_L = \frac{2mg}{\rho v_a^2 S} \qquad (7)$$

Here C_L is lift coefficient, ρ is density of the air, and S is wing area. A polynomial approximation is used for the
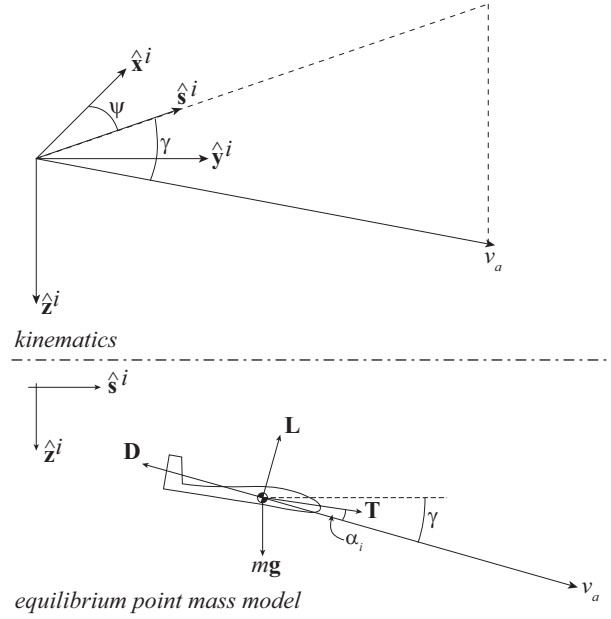

Fig. 1.   Kinematics and point mass model of aircraft dynamics.

aircraft's drag polar:

$$C_D = \sum_{i=0}^{n} a_i C_L^i \qquad (8)$$

Thus the drag force is

$$D = \frac{1}{2}\rho v_a^2 S \sum_{i=0}^{n} a_i C_L^i \qquad (9)$$

Typically a second order polynomial is used to represent drag coefficient. However, this is often only valid over a fairly narrow speed range, and here a fourth order polynomial is used.

Substituting into Equation 5, the flight path angle for a particular speed and thrust can thus be computed as

$$mg\gamma = \frac{1}{2}\rho v_a^2 S \sum_{i=0}^{n} a_i C_L^i - T \qquad (10)$$

The vehicle's flight path is thus completely specified by inputs $\mathbf{u} = [v_a \ \psi \ T]^T$ and wind speed $\mathbf{w}$. This model is adequate as long as the length of time of each trajectory segment is large compared with the time constant of the vehicle's step response with respect to the inputs $\mathbf{u}$.

III. Tree-based Approach to Planning

Given a wind field $\mathbf{w} = w(x, y, z)$ and terrain map $z_{terrain} = a(x, y)$ the problem is to find a trajectory from an initial position and velocity to a goal position and velocity.

This can be expressed as

$$\text{minimize} \quad C(\mathbf{x}, \mathbf{u}) \tag{11}$$

$$\text{subject to} \quad \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{w}) \tag{12}$$

$$\mathbf{x}(t=0) = \mathbf{x}_0 \tag{13}$$

$$\mathbf{x}(t=t_f) = \mathbf{x}_f \tag{14}$$

$$\mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max} \tag{15}$$

$$\mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max} \tag{16}$$

where C is a cost function, Equation 12 defines constraints on vehicle dynamics, Equation 15 defines constraints on the state (for example minimum altitude or minimum/maximum airspeed), and finally Equation 16 defines constraints on control inputs (for example control surface deflections). Because of vehicle kinematics and the complex wind field this is generally a highly non-convex optimization problem. Solvers such as MatLab's fmincon generally require a feasible initial guess, but even finding a feasible solution is non-trivial.

In this case a randomly expanding tree can be used to find a feasible solution or set of feasible solutions.

A. Pre-computation of Branches

To reduce the time required for expanding the tree, a set of branches for a family of allowable inputs can be defined:

$$\mathbf{U} = \left\{ [v_{a,m} \ \Delta\psi_n]^T \ \middle| \ v_{a,m} \in V_a, \ \Delta\psi_n \in \Delta\Psi \right\} \tag{17}$$

$$m = 1, \ldots, M \quad n = 1, \ldots, N$$

where $v_{a,m}$ is an airspeed command and $\Delta\psi_n$ is a heading change command. The set of allowable inputs V_a and $\Delta\Psi$ are obtained by discretizing the range of inputs $v_{a,min} \leq v_a \leq v_{a,max}$ and $\Delta\psi_{min} \leq \Delta\psi \leq \Delta\psi_{max}$. The degree of discretization is a parameter which can be varied depending on computational and accuracy considerations.

Given a choice of input $\mathbf{u}_{mn} \in \mathbf{U}$ and a time interval δt the change in position and altitude can be computed from the kinematics (Equation 1 through Equation 10):

$$\Delta\mathbf{x}^b_{mn} = \begin{bmatrix} \Delta x^b_{mn} \\ \Delta y^b_{mn} \\ \Delta z^b_{mn} \end{bmatrix} = \delta t \begin{bmatrix} v_{a,m} \cos\gamma \cos\Delta\psi_n \\ v_{a,m} \cos\gamma \sin\Delta\psi_n \\ v_{a,m} \sin\gamma \end{bmatrix} \tag{18}$$

Note that wind speed $\mathbf{w}$ is not included in this set of pre-computed branches: it is included when a particular node in the tree is expanded. The set of branches $\mathbf{X}^b$ consists of all possible changes in position and altitude given the set of possible inputs $\mathbf{U}$, and is formed by concatenating the vectors of branches:

$$\Delta\mathbf{X}^b = \begin{bmatrix} \Delta\mathbf{x}^b_{11} & \ldots & \Delta\mathbf{x}^b_{MN} \end{bmatrix} \tag{19}$$

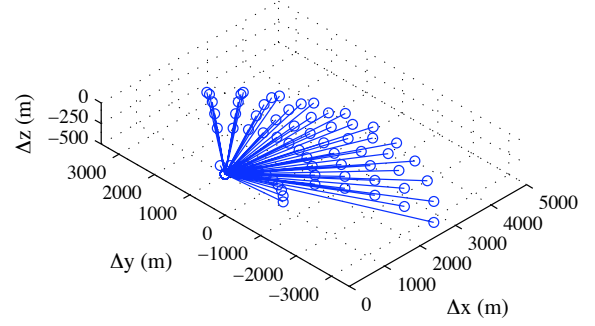A representative set of branches is shown in Figure 2.

Fig. 2. Pre-computed branches for set of allowable inputs.

B. Node definition, selection and expansion

In addition to geometric position, each node in the tree contains values for internal variables whose changes are not directly encoded in the branches but whose values are affected by the path taken to reach that point. Each node encodes

$$\mathbf{n}_i = \begin{bmatrix} x_i & y_i & z_i & \psi_i & v_{a,i} & E_{tot,i} & r_{goal,i} \end{bmatrix}^T \tag{20}$$

where $[x_i \ y_i \ z_i]$ is vehicle position, ψ_i is heading, $v_{a,i}$ is airspeed, $E_{tot,i}$ is the specific total energy of the vehicle at the node ($E_{tot} = gh + \frac{v_a^2}{2}$) and $r_{goal,i}$ is distance remaining to the goal. Note that both $E_{tot,i}$ and $r_{goal,i}$ can be computed from the other variables: they are encoded in the node for convenience.

1) Node selection: In many tree-based planning algorithms nodes are selected for expansion at random. While this ensures even coverage of the space, it also has the potential to lead to a large number of 'dead' nodes that cannot lead to a feasible solution. In this application there are generally only narrow corridors that lead to the goal (e.g. a ridge which triggers a long, narrow region of rising air), and a biased approach is used to favor nodes which are more likely to lead to a feasible path to the goal.

To increase the likelihood that widely separated nodes are selected a stratified approach is used for node selection. Nodes are grouped according to distance from the goal, a random group is chosen and a node is chosen from this group using a weighted random approach. Each node is assigned a weight based on energy altitude h_E and the horizontal distance to the goal. Energy altitude is the altitude which results when all the vehicle's kinetic energy is transformed to potential energy, and is a common way of representing total energy:

$$h_E = h + \frac{v_a^2}{2g} \tag{21}$$

where h is height above a datum (e.g. the goal).

The weight assigned to the ith node is

$$w_i = \left(\frac{h_{E,i}}{r_{goal,i}} \right)^2 \tag{22}$$

and the probability of choosing a particular node from the group is proportional to its weight.

The weighted random approach was chosen because it biases the search in a direction likely to lead to a feasible solution but has the potential to explore the space. Note that the value of the exponent on the weight w_i serves to "stretch" the range of values, and can thus be used to increase the likelihood of choosing a higher weighted node by increasing the difference between the smallest and largest weight.

2) Node expansion: The selected node with position $\mathbf{x}_i$ is expanded using the pre-computed branches and wind speed to define a set of candidate nodes.

$$\mathbf{X}_{i,new} = \mathbf{x}_i \mathbf{1} + \mathbf{T}_i \Delta \mathbf{X}^b + \mathbf{w}_i \mathbf{1} + \begin{bmatrix} 0 \\ 0 \\ \Delta \mathbf{z}_i \end{bmatrix} \quad (23)$$

where $\mathbf{1}$ is a $1 \times MN$ array of ones, $\mathbf{w}_i$ is the wind vector computed at $\mathbf{x}_i$, and

$$\mathbf{T}_i = \begin{bmatrix} \cos\psi_i & \sin\psi_i & 0 \\ -\sin\psi_i & \cos\psi_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (24)$$

is the transformation which rotates the set of precomputed branches to the local frame defined by the heading ψ_i. Finally $\Delta \mathbf{z}_i = [\Delta z_{i,1} \ \Delta z_{i,2} \ \ldots \Delta z_{i,MN}]$ with

$$\Delta z_{i,m} = \frac{v_{a,i}^2 - v_{a,m}^2}{2g} \quad (25)$$

The term Δz_m accounts for the change in altitude which occurs with a change in speed, assuming that total energy is constant during the transition. This is not reflected in the kinematic model of the aircraft and must therefore be accounted for separately.

The wind vector $\mathbf{w}_i$ is computed at $\mathbf{x}_i$ and is assumed to be constant over the time δt required to fly the branch. Weather prediction tools such as MM5 and WRF are able to compute wind field over a grid spacing of approximately 1.5 - 2 km: at the flight speeds of small gliders (10 m/s to 30 m/s) a time interval of 50s to 150s will provide adequate discretization. Clearly smaller values of δt will allow more accurate computation of wind over each branch of the tree, but this will result in increased computational cost.

Before adding them to the tree the set of candidate nodes $\mathbf{X}_{i,new}$ is checked for feasibility. Currently two criteria are used to cull infeasible nodes: height above ground and maximum allowable divergence of heading from the goal. Nodes for which

$$z < z_{terrain} + z_{safety} \quad (26)$$
$$|\psi - \beta_{goal}| > \Delta\psi_{goal} \quad (27)$$

are declared infeasible, the remaining nodes are added to the tree. Here z_{safety} is an altitude buffer to ensure adequate height above terrain, $\Delta\psi_{goal}$ is the maximum allowable divergence of heading from β_{goal}, the current bearing to the goal.
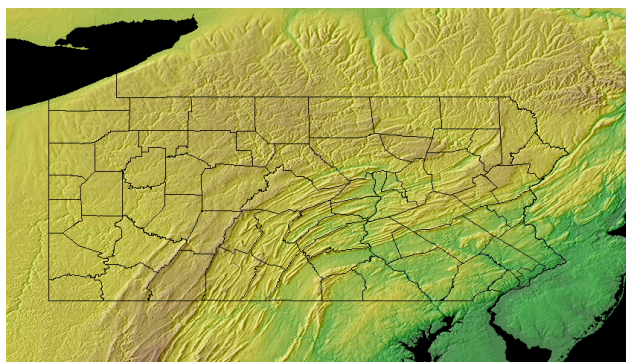

Fig. 3. Digital elevation map of Pennsylvania showing ridges of the Appalachian mountains.

C. Termination

Tree expansion terminates when a node is within gliding distance of the goal. This can be expressed as $r = (L/D)h$, where L/D is the glide ratio. Maximum glide ratio occurs at a particular airspeed which depends on wind speed: in still air the maximum glide range occurs when the ratio of lift to drag is maximized, and the speed for maximum L/D will be denoted $v_{L/D,max}$.

For a vehicle flying at $v_{L/D,max}$ the end game region is the set $\{r_{goal}, h\}$ which satisfies $r_{goal}/h \leq L/D|_{max}$, where h is the altitude above the goal. Recalling that airspeed can be exchanged for altitude, the endgame region can be expressed as the set $\{r_{goal}, h, v_a\}$ which satisfies

$$\frac{r_{goal}}{h + \frac{1}{2g}\left(v_a^2 - v_{L/D,max}^2\right)} \leq L/D|_{max} \quad (28)$$

This formulation of the endgame region reflects that a vehicle which is low but flying fast may still have enough total energy to reach the goal in a simple glide.

IV. SIMULATION

The Appalachians of central Pennsylvania consist of long, parallel ridges, often separated by only a few kilometers (Figure 3).

Here we consider a flight to a distant goal which requires crossing two ridges. The starting position is within gliding distance of the near ridge, but the starting altitude is not high enough to clear the ridge without exploiting orographic lift. Further, significant altitude must be gained in order to cross the gap between the ridges, and then sufficient altitude must be gained on the second ridge to enable a glide to the goal. The ridges are parallel to the y axis and are modeled as infinitely long hemi-cylinders (Figure 4). Far from the ridges the wind speed is uniform, with only a component of 5 m/s in the positive x direction. Terrain, wind and trajectory parameters are given in Table I.

The wind field is computed using potential flow, which allows a closed-form computation of the 3D wind vector at any (x, y, z). A vertical slice through the wind field is shown in Figure 5. The vertical component of wind is greatest just upwind of the ridge, and the magnitude of the vertical
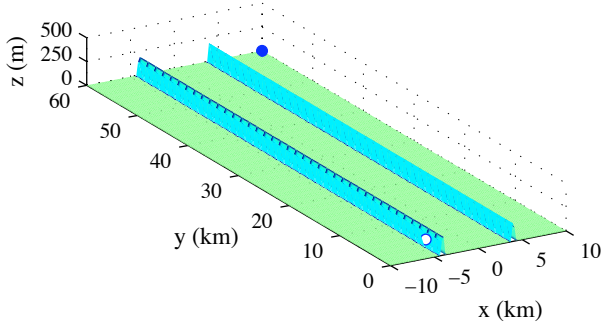
Fig. 4. Scenario for simulation of glider flight. The start is shown by the open circle, the goal is shown by the closed circle.

parameter	value
ridge centerline x-coordinate (m)	[-4000, 4000]
ridge radius (m)	[200, 200]
wind speed at ∞ (m/s)	[5 0 0]
trajectory start (m)	[-6000 0 200]
trajectory goal (m)	[10000 60000 0]

component falls off quickly with altitude. Thus successful exploitation of orographic lift requires flight in fairly close proximity to terrain.

Branches were computed for a small autonomous glider based on an RnR Products SB-XC radio-controlled glider (properties given in Table II) using the following sets of allowable inputs:

$$V_a = \begin{bmatrix} 10 & 15 & 20 & 25 & 30 & 35 \end{bmatrix} \quad (29)$$
$$\Delta\Psi = \begin{bmatrix} -50° & -40° & \dots & 40° & 50° \end{bmatrix} \quad (30)$$

with time interval $\delta t = 120$s. The maximum allowable heading divergence from the goal was $\Delta\psi_{goal} = 60°$.

Given the starting altitude of 200m, the maximum gliding distance in still air is 5000m, approximately 8% of the distance to the goal. Flying at best glide ratio to the nearest ridge will place the glider about 100m below the crest of the ridge when it arrives: clearly it must extract energy from the atmosphere if it is to continue with safe flight.
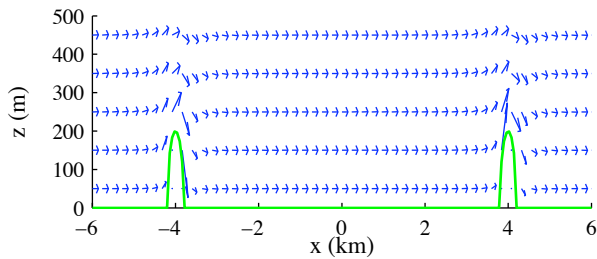

Fig. 5. Vertical slice through terrain and wind field. Vectors show wind speed, the heavy line shows terrain.

parameter	symbol	value
mass	m	10kg
wing area	S	1m^2
drag coefficient	C_D	$0.1723C_L^4 - 0.3161C_L^3 + 0.2397C_L^2$ $-0.0624C_L + 0.0194$
best glide ratio	$L/D|_{max}$	25

	max	min	median	90%	95%
CPU time (s)	423.6	2.886	31.27	96.01	143.9
# nodes	196272	2667	25636	69240	98456

To assess planning performance, 100 runs were performed with the same scenario (randomness enters the problem through the node selection algorithm). Results are tabulated in Table III. A 2.8GHz dual-processor Dell Xeon desktop computer was used for these simulations.

In half the cases a path was found within one quarter of the time taken to fly a single segment of the trajectory (31s to find a path, segment time is 120s). In only 6% of the cases was the time required to find a path longer than that required to fly a single segment. While unexpected variation in wind field is not explicitly addressed by this planning approach, fast re-planning is one possible avenue for coping with a time-varying or uncertain wind field. Given *in situ* measurements of wind (collected during flight), an updated wind field can be computed and used for generating new plans.

Ground tracks for all paths computed in this simulation are shown in Figure 6. The influence of the two ridges is clearly seen on the ground tracks: paths cluster along the ridges, and the gap crossing appears uniformly distributed.

Since the time of flight is the same for each flight segment, the time of flight to reach the goal for a particular path can be obtained directly from the number of nodes traversed. The flight path, flight speed and heading angle for the minimum time trajectory is shown in Figure 7.

The path begins with flight along the near ridge at almost constant airspeed. Note that the heading is not exactly 90°: the horizontal component of wind speed keeps the ground track of the aircraft parallel to the ridge.

V. CONCLUSION

This paper has presented a tree based approach to planning trajectories which exploit atmospheric energy. A kinematic model is used for the vehicle and a set of branches is pre-computed for the space of allowable inputs (for the glider considered here, air speed and heading). Upon selection of a node, this set of branches is added to the tree and checked for feasibility. Feasibility of the resulting nodes is checked based on height above terrain and allowable divergence from the goal and infeasible nodes are pruned.
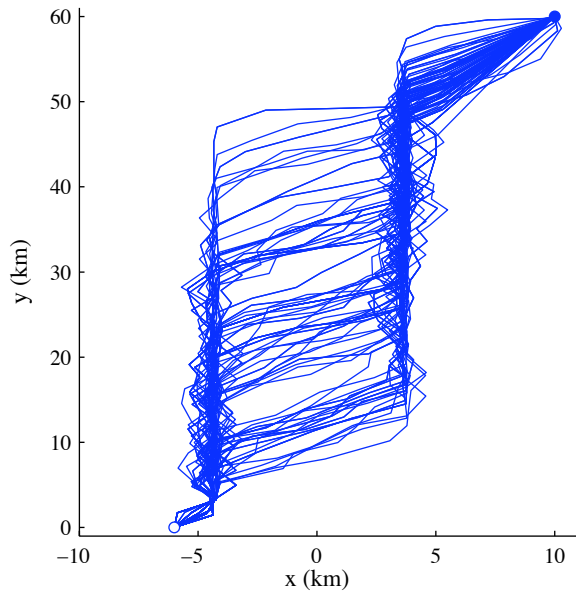
Fig. 6. All paths computed using the tree-based planner for 100 runs (projected onto the plane $z = 0$). The start position is shown by the open circle, the goal is shown by the solid circle.

A weighted random approach is used for node selection, with the weight dependent on energy altitude and distance from the goal. This directs the search towards the goal but allows for exploration of the space. Tree expansion terminates when a node is within gliding distance of the goal.

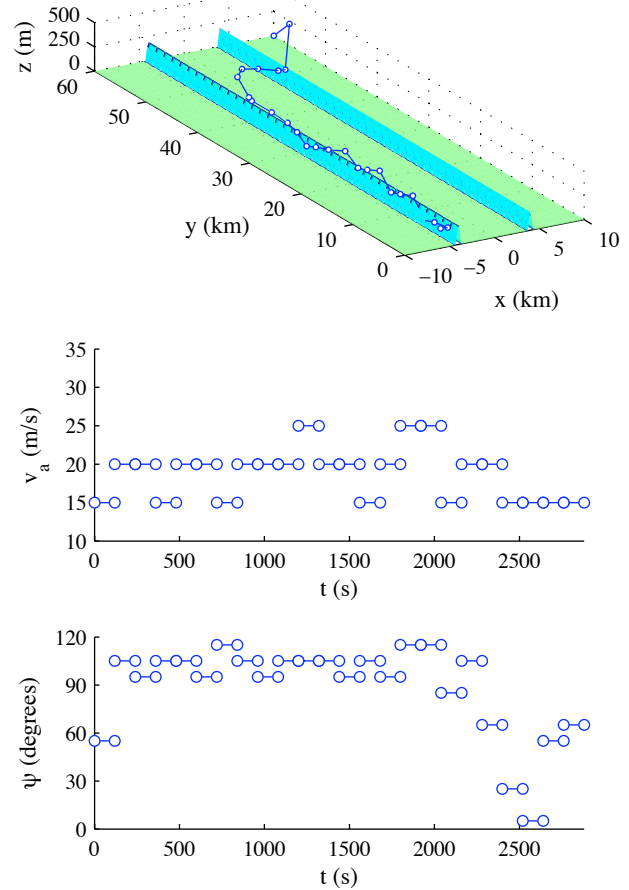Simulation results show that feasible soaring trajectories can be found for long distance flights.

Fig. 7. Path and input history for the fastest flight to goal. The top plot shows flight path, the middle plot shows airspeed, the lower plot shows heading. The gap is crossed approximately 2300s into the flight. *note: the apparent flight through terrain in the third segment is an artifact of plotting.*

REFERENCES

[1] J. W. S. Rayleigh, "The soaring of birds," *Nature*, vol. 27, pp. 534–535, 1883.

[2] ——, "The sailing flight of the albatross," *Nature*, vol. 40, p. 34, 1889.

[3] M. B. E. Boslough, "Autonomous dynamic soaring platform for distributed mobile sensor arrays," Sandia National Laboratories, Sandia National Laboratories, Tech. Rep. SAND2002-1896, 2002.

[4] T. Kiceniuk, "Calculations on soaring in sink," *Technical Soaring*, vol. 25, no. 4, pp. 228–230, October 2001.

[5] C. K. Patel and I. Kroo, "Control law design for improving UAV performance using wind turbulence," in *AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2006-0231. Reno, Nevada: American Institute of Aeronautics and Astronautics, January 2006.

[6] MM5 community model homepage. [Online]. Available: http://www.mmm.ucar.edu/mm5/

[7] S. R. Lindemann and S. M. LaValle, "Current issues in sampling-based motion planning," in *Robotics Research: The Eleventh International Symposium*, P. Dario and R. Chatila, Eds. Berlin: Springer-Verlag, 2005, pp. 36–54.

[8] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[9] J. Barraquand, L. Kavraki, J.-C. Latombe, T.-Y. Li, R. Motwani, and P. Raghavan, "A random sampling scheme for path planning," *International Journal of Robotics Research*, vol. 16, no. 6, pp. 759–774, 1997.

[10] S. M. LaValle and J. J. Kuffner, Jr., "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001.

[11] D. Hsu, R. Kindel, J. C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, March 2002.

[12] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, B. R. Donald, K. M. Lynch, and D. Rus, Eds. Wellesley, MA: A. K. Peters, 2001, pp. 293–308.

[13] J. Kim and J. P. Ostrowski, "Motion planning of aerial robot using rapidly exploring random trees with dynamic constraints," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, September 2003, pp. 2200–2205.

[14] J. C. Rubio and S. Kragelund, "The trans-pacific crossing: long range adaptive path planning for uavs through variable wind fields," in *Proceedings of the 22nd Digital Avionics Systems Conference*, vol. 2. Piscataway, New Jersey: IEEE, October 2003.

[15] B. Glavina, "Solving findpath by combination of goal-directed and randomized search," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, May 1990, pp. 1718–1723.

[16] D. Hsu, L. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin, "On finding narrow passages with probabilistic roadmap planners," in *Proceedings of the International Workshop on Algorithmic Foundations of Robotics*, 1998. [Online]. Available: http://citeseer.ist.psu.edu/article/hsu98finding.html