J Intell Robot Syst (2010) 57:329–349
DOI 10.1007/s10846-009-9389-8

Automating Human Thought Processes for a UAV Forced Landing

Pillar Eng · Luis Mejias · Xi Liu · Rodney Walker

Received: 1 February 2009 / Accepted: 1 August 2009 / Published online: 12 November 2009

Abstract This paper describes the current status of a program to develop an automated forced landing system for a fixed-wing Unmanned Aerial Vehicle (UAV). This automated system seeks to emulate human pilot thought processes when planning for and conducting an engine-off emergency landing. Firstly, a path planning algorithm that extends Dubins curves to 3D space is presented. This planning element is then combined with a nonlinear guidance and control logic, and simulated test results demonstrate the robustness of this approach to strong winds during a glided descent. The average path deviation errors incurred are comparable to or even better than that of manned, powered aircraft. Secondly, a study into suitable multi-criteria decision making approaches and the problems that confront the decision-maker is presented. From this study, it is believed that decision processes that utilize human expert knowledge and fuzzy logic reasoning are most suited to the problem at hand, and further investigations will be conducted to identify the particular technique/s to be implemented in simulations and field tests. The automated UAV forced landing approach presented in this paper is promising, and will allow the progression of this technology from the development and simulation stages through to a prototype system.

Keywords UAV · Forced landing · Path planning · Control · Multi-criteria decision making

P. Eng (✉) · L. Mejias · X. Liu · R. Walker
Australian Research Centre for Aerospace Automation, Queensland University of Technology, Brisbane, Queensland, 4001, Australia
e-mail: p.eng@qut.edu.au

L. Mejias
e-mail: luis.mejias@qut.edu.au

X. Liu
e-mail: xi.liu@qut.edu.au

R. Walker
e-mail: ra.walker@qut.edu.au

1 Introduction

The field robotics community has existed for over 25 years and has made good progress in the areas of ground, underwater and aerial robotics. During this time, maturity in Information and Communication Technologies (ICT) and sensor technologies has brought, to a certain extent, the dream of commercial field robots to reality. There are numerous prototype field robot systems deployed for military experiments, with some robots even seeing routine use (particularly aerial surveillance robots). Many of these systems have been rapidly pressed into service due to operational demands in times of conflict, rather than through careful development of the design requirements. Military experiences with aerial robots, or Unmanned Aerial Vehicles (UAVs) have also indicated the difficulty of integrating a field robot into an environment where failure of the robot can harm the general public. The key being that, particularly in the case of a UAV, the pertinent safety regulations describe human-centred capabilities. We have learnt in recent times that it is impossible to directly apply these regulations to an autonomous system, and the problem remains in how to integrate fundamentally new technology into a highly regulated human-centred environment where failure can lead to loss of human lives. In our case, this means integrating UAVs into an environment predominated by human pilots and human air traffic controllers, where a mid-air collision between a UAV and a passenger aircraft could have catastrophic results both in the air and on the ground.

The universal position of the safety regulators is to require developers of aerial robots to prove that their systems have equivalence to human performance and to their human-centred safety regulations [1]. In this regard, algorithms for UAV Sense-and-Avoid and Force Landings are recognized as two major enabling technologies that will allow the integration of UAVs into civilian airspace [2]. In the former case, the aircraft must be able to reliably detect and avoid collision with both stationary and moving objects in its path of interest, which may/may not announce their position. However, the assumption is that the robot is still capable of powered, controlled flight. In the latter case, the aircraft is forced to perform an unplanned landing due to the occurrence of some onboard emergency, such as engine, systems, sensors or control surface failure. A forced landing due to engine failure is commonly practiced by pilots during flight training and for ongoing safety certification, and the ability to conduct a safe landing in such situations is primarily used to benchmark performance of the manned aviation industry. This procedure involves firstly visual estimation of wind conditions and recognition of appropriate landing sites, then the formulation of a descent trajectory which accounts for wind changes as well as the glide range and manoeuvrability of the aircraft. On final approach to land, the pilot must also avoid trees, power lines, buildings and other obstacles which may have been invisible from the air [1]. Note also that there is limited possibility for replanning the path, since the aircraft is devoid of thrust control and is continually descending. As many of the same problems confronting manned aircraft also affect robotic aircraft, we believe that UAVs must be capable of safe flight termination following an engine failure, as a UAV plummeting uncontrollably into the middle of a busy freeway or a school yard is a risk that the public will be unwilling to accept.

To date, no commercial system exists that allows a UAV to autonomously select the safest emergency landing area in an unknown environment. The most commonly employed method to allay the severity of a UAV forced landing is the use of parachutes or parafoils to retard the rate of descent, while still providing some degree of controllability for the aircraft [3]. Whilst this concept is attractive in that it still enables limited vehicle controllability even when both the engine and control surfaces have failed, it is highly susceptible to wind gusts and other atmospheric effects which may adversely affect the final impact point. Having a parachute or parafoil onboard also adds to the weight and complexity of the aircraft. Other safety systems currently available allow the UAV to fly towards a pre-defined safe ditching area selected from a database of such, known locations. However, these systems must be preprogrammed with up-to-date information, thus requiring a continuous communications link between a human operator and the air vehicle to ensure that the latter will not attempt landing at an unsuitable location. To date, the only reported successful UAV forced landing involves the U.S. Air Force Global Hawk, which performed a gliding descent under remotely-piloted control to an emergency airstrip in 2006 [4].

An alternative would be to have an automated system onboard the UAV which can process information in a way similar to human pilots, during emergency situations that require the aircraft to land. Here, we have restricted our discussion to the case of engine failure only, and assumed that the onboard avionics and flight control surfaces are still intact for glided flight. Further, to simplify the planning process, we have also omitted obstacles in the flight path and assumed that wind velocities can be estimated by onboard instruments (albeit with certain errors). Such a system is currently under development in this study, and is divided into the three research areas of:

1. Automated visual identification and classification of UAV forced landing sites;
2. Automated multi-criteria decision making for high-level reasoning during the descent; and
3. Automated path planning, guidance and control for descent and landing;

The site identification and classification component uses computer vision and onboard sensors to quickly identify suitable landing areas, and is described in detail in [5, 6]. This paper will present the current research progresses in the path planning, guidance and control component, as well as in the multi-criteria decision making component, respectively. Recommendations will also be given on how to further improve and enhance this research.

2 Path Planning, Guidance and Control

The current work progress in this area has involved mainly the simulation of path planning and control strategies using MATLAB. The advantages of simulation are that it simplifies the debugging of code, allows repeatable comparison of different planning and control scenarios, and allows analysis of the UAV response under ideal conditions that set the benchmark performance to be pursued in later experiments.

Previously, two algorithms derived from piloted forced landing procedures as outlined in [1] were developed and tested in simulation using the MATLAB computing program. Algorithm 1 attempted to guide the aircraft (a model of an Aerosonde UAV provided by MATLAB), along a predetermined circuit to the touchdown point on the desired landing site while correcting for the wind on course (Fig. 1a).

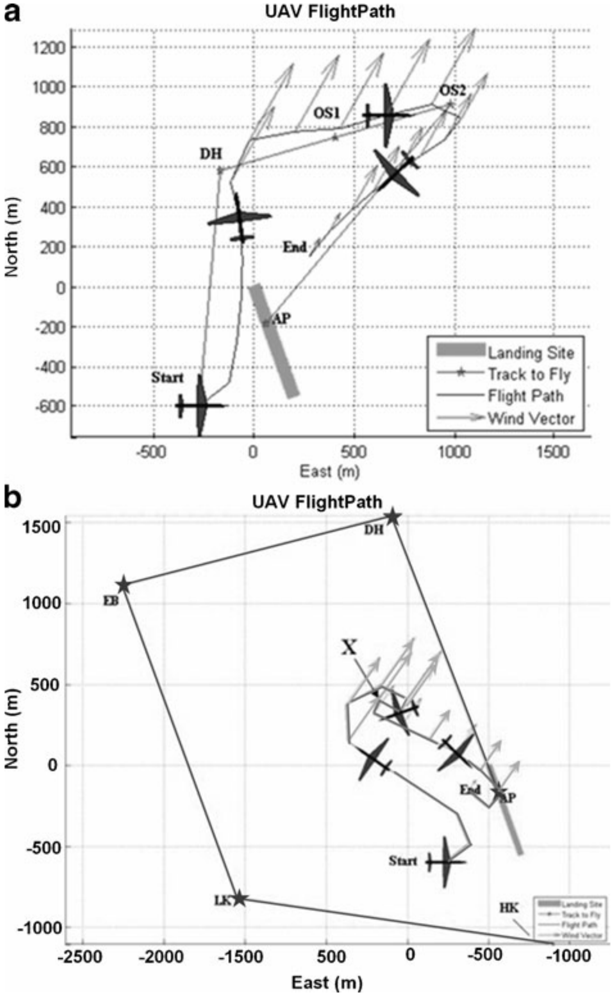


Fig. 1 Path planning simulation results showing flight path in top view for **a** the path described using Algorithm 1, and **b** the path described using Algorithm 2

A number of predefined paths were available and the UAV could choose which path to follow depending on the wind conditions and its proximity to the landing site. The second algorithm did not restrict the aircraft to a predetermined circuit; instead, the UAV was allowed to construct its own path depending on the wind conditions and its ability to reach a certain waypoint (Fig. 1b). This ability was determined from the aircraft glide slope, which was a function of the current wind condition. In both cases, flight stability for the aircraft was maintained using a cascade of Proportional-Integral-Derivative (PID) controllers and obstacles in the flight path were not considered. In addition, both algorithms utilized the great-circle navigation method [1], together with wind triangle calculations [1], to navigate between waypoints. Although a flat-earth approximation was sufficient, the great-circle method was chosen as the basic MATLAB source code had already been written for a different project at ARCAA, and it was decided to extend this work to reduce the development time. The wind velocities supplied to the model reflected average wind measurements recorded for Brisbane from 1950 to 2000 (available on the Australian Government Bureau of Meteorology website [7]), and an internal AeroSim function then used these velocities to calculate wind shear and turbulence effects on the aircraft using a von Karman approximation. To test the effectiveness of the two algorithms, a Monte Carlo simulation with 100 forced landing scenarios was set up in which the aircraft position and bearing, as well as the wind velocities were randomised. The results showed that using Algorithm 2, the UAV was able to land within the designated area 52% of the time, compared to 26% using Algorithm 1. In addition, Algorithm 2 produced a lower miss distance of less than or equal to 200 m from the touchdown point, compared to the miss distance of Algorithm 1, which was less than or equal to 400 m. A major factor affecting the miss distance was attributed to the strong winds modelled (up to 8 m/s), which were often greater than the forward speed of the aircraft. The fact that the UAV could not adjust its airspeed to counter changing wind conditions, but only its heading, could also have contributed to the large miss distances. Full details of these early developments can be found in [2, 8].

Thus, we have endeavoured to overcome these limitations in our current design approach. Firstly, we have replaced the previous Aerosonde model with a 6 degree-of-freedom (6-DOF) model of a Boomerang radio-controlled aircraft, which represents the UAV platform to be used in future flight tests. The model was constructed using the classic coefficient build-up method [9] as part of an undergraduate project at the Queensland University of Technology (QUT). Specifically, the challenge we face is that of how to guide an unpowered, fixed-wing aircraft to arrive at a specific point in space (approach point) where it is aligned with the crash site for landing/ditching, and at a certain airspeed and heading while accounting for any kinodynamic constraints, regardless of the ambient wind conditions. Note that in this case, the approach point is likened to the touchdown point in our previous experiments, but we have classified the planning and guidance involved from the approach point to the touchdown point as outside the scope of the current research. In designing the planning, guidance and control algorithms we have also assumed that a feasible landing area exists and that the desired final approach point, airspeed and heading are supplied by the multi-criteria decision making algorithm.

2.1 Path Planning

Numerous robotics path planning techniques are presented in the literature, and a comprehensive summary of existing methods can be found in [10]. Since a gliding, fixed-wing aircraft can achieve only forward motion and is also limited by constraints in its turn and descent rates, we have found that trajectories derived from Dubins curves [11] present one of the simplest solution that satisfies these constraints. We have also assumed that the aircraft can achieve a nominal lift-to-drag ratio of 9:1 in planning the path, meaning that for every 1000 ft loss in altitude, the aircraft glides 9000 ft.

Initially, a 2D Dubins path is constructed having the form:

$$\{L_\alpha R_\beta L_\gamma,\, R_\alpha L_\beta R_\gamma,\, L_\alpha S_d L_\gamma,\, L_\alpha S_d R_\gamma,\, R_\alpha S_d L_\gamma,\, R_\alpha S_d R_\gamma\} \tag{1}$$

in which L and R correspond to left and right turns at a bank angle that does not exceed the maximum bank angle of the aircraft, S corresponds to flying in a straight line, and $\alpha, \gamma \in [0, 2\pi)$, $\beta \in (\pi, 2\pi)$, and $d \geq 0$. The radii of the arcs were calculated using the equation:

$$R_{0,f} = \frac{V_{TAS}^2}{g \times tan(\phi_{0,f})} \tag{2}$$

where $R_{0,f}$ are the initial and final radii of the arcs of circumference, V_{TAS} is the True Airspeed of the aircraft, g is the gravitational acceleration constant ($9.80665\ m/s^2$), and $\phi_{0,f}$ are the initial and final bank angles respectively, which can be different.

To simplify the path planning process, we have taken V_{TAS} to be the best glide speed, V_{bg} of the aircraft, which gives the greatest straight line flight distance in still air from the potential energy of height. As shown in Fig. 2, we can estimate V_{bg} by first fitting a curve (black) through the descent rates at various airspeeds (red diamonds). The best glide speed is then obtained by drawing the blue line from the origin tangent to the curve, giving V_{bg} as 18.63 m/s. The descent rates were calculated in simulation by setting the wind speed to zero and commanding the aircraft to fly continuously in a box circuit, with a different airspeed selected for each leg of the circuit.

Once the radii are determined, the optimal 2D path is obtained with a geometrical construction adapted from [12]. Initially, two circles with radii R_0 are drawn containing the starting point P_0 and a vector pointing along the aircraft's initial heading ψ_0 (Fig. 3a). The circumferences of the circles are denoted by $\sum_{0A}$ and $\sum_{0B}$. Next, the same process is repeated at the goal point P_f with the final aircraft heading ψ_f, and circumferences $\sum_{fA}$ and $\sum_{fA}$. Following this, tangent lines are constructed that join the circumferences of these circles, such as depicted in Fig. 3b for $\sum_{0A}$ and $\sum_{fA}$. Considering Fig. 3b, we readily observe that there are four paths connecting P_0 to P_f, where a path is formed by the union of an arc on the circumference $\sum_{0A}$, a segment K on one of the four tangent lines, and finally an arc on the circumference $\sum_{fA}$. However, only one of these paths, Γ_{AA} is compatible with the initial and final

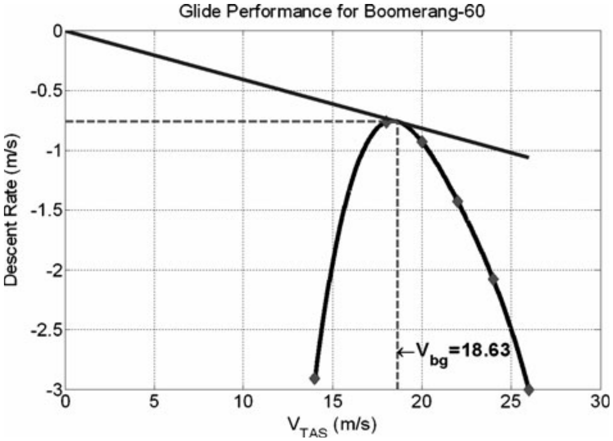


Fig. 2 Speed polar diagram for a Boomerang 60 size UAV, showing how the best glide speed (V_{bg}) is obtained

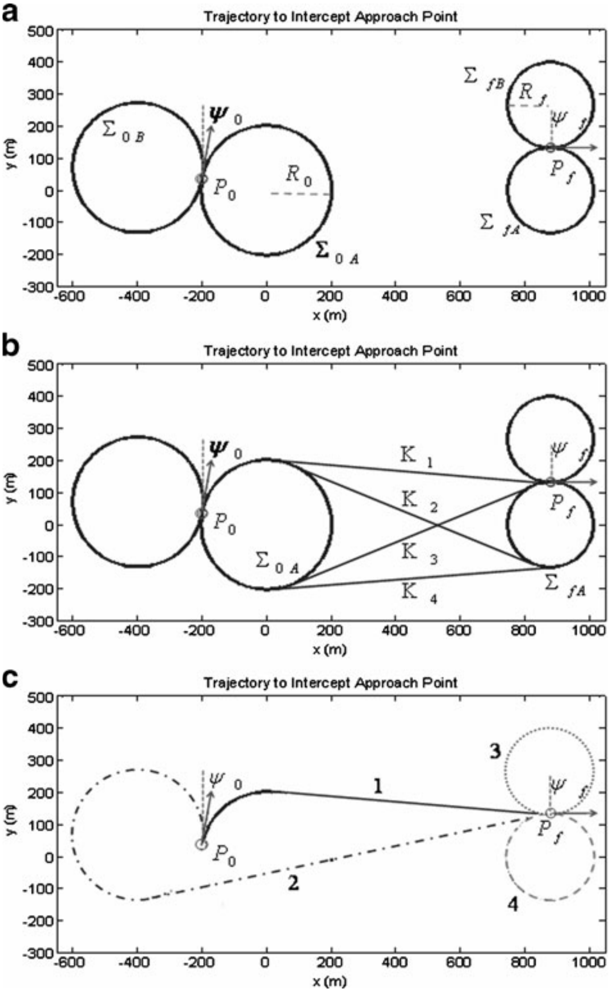


Fig. 3 **a** Step 1 of generating the 2D path. **b** Step 2 of generating the 2D path. **c** Step 3 of generating the 2D path. Four plausible paths are obtained; the optimal path is path no. 1

headings of the UAV (Fig. 3b and c). In a similar way, three other paths Γ_{AB}, Γ_{BA} and Γ_{BB} can be obtained—the optimal path is the shortest between Γ_{AA}, Γ_{AB}, Γ_{BA} and Γ_{BB} and is depicted as a thick, solid line in Fig. 3c.

Following the construction of Γ_{xy}, and given the distance d_{tgt} (distance from the initial point of failure to the approach point), we can then obtain the path angle:

$$\gamma_{xy} = tan^{-1}\left(\frac{z_{arc_f} - z_{arc_0}}{d_{tgt}}\right) \tag{3}$$

which allows the UAV to descend from an altitude of z_{arc_0} to z_{arc_f}. To ensure stability, $\gamma_{min} \leq \gamma \leq \gamma_{max}$. However, if the difference in altitude between the start and end positions should result in the maximum allowable path angle being exceeded, one of the other suboptimal paths can be selected to lose the approximate amount of altitude required. Other options include enlarging R_0 and/or R_f, as well as commanding the aircraft along a helical trajectory (similar to a spring) to lose excess altitude, before joining the path at the start of the first arc.

To form the 3D path, we note that a gliding aircraft that is rolled into a steady, coordinated turn at a constant bank angle ϕ and flies at a constant descent angle will trace a helical path γ on an imaginary cylinder with radius R. Thus, the 3D path can be formed by a straight line at a constant pitch angle that joins two arc sections. In order to simplify the design, we have not included the $\{L_\alpha R_\beta L_\gamma\}$ or $\{R_\alpha L_\beta R_\gamma\}$ types paths, these will be addressed in future work. The relationship between ϕ and γ is given by:

$$cot_{\gamma_{0,f}} = \frac{V_{TAS}}{V_S} cos_{\phi_{0,f}} \tag{4}$$

and V_s is the descent/sink rate of the powerless aircraft as shown in Fig. 2. Now, the altitude lost while transversing the two arc sections can be calculated as:

$$S_{0,f} = \frac{1}{2\pi} \frac{\| \sum_{0,f} \|}{R_{0,f}} S_{\phi_{0,f}} \tag{5}$$

Giving the altitude where the arc sections join the 3D line as:

$$\begin{aligned} z_{arc_0} &= z_0 - S_0 \\ z_{arc_f} &= z_f + S_f \end{aligned} \tag{6}$$

where, z_0 is the altitude at the start of a forced landing, and z_f is the desired altitude to achieve at the final approach point. Given the terminal points on the arcs $P_{arc_0} = [x_{arc_0}, y_{arc_0}, z_{arc_0}]$ and $P_{arc_f} = [x_{arc_f}, y_{arc_f}, z_{arc_f}]$, it is then a straightforward process to obtain Γ_{line}. The relationship between the different elements of the 3D path ($\Gamma = \Gamma_{arc_0} \cup \Gamma_{line} \cup \Gamma_{arc_f}$) is illustrated in Fig. 4.

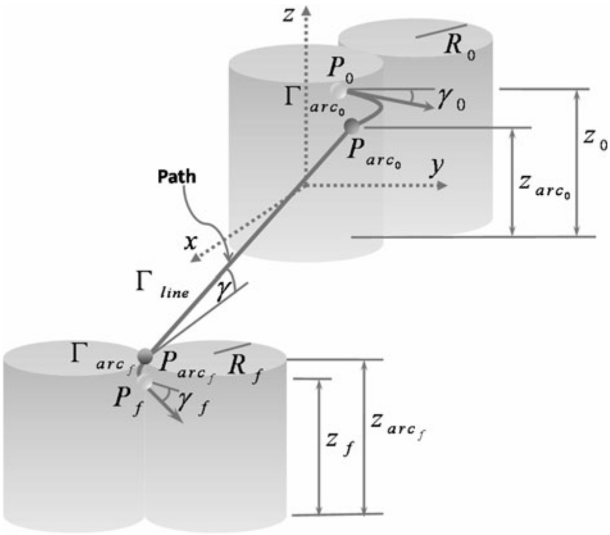


Fig. 4 Relationship between elements of the generated 3D flight path. The generated path is $\Gamma_{arc_0} \cup \Gamma_{line} \cup \Gamma_{arc_f}$

2.2 Guidance and Control

Our guidance algorithm is composed of both a lateral and longitudinal component. The lateral component is based on the work presented in [13, 14]; however, we have also built upon this algorithm to encapsulate wind information in the guidance logic, rather than merely treating wind as an adaptive element for the control system. This addition to the control law has demonstrated robust performances for linear path following in strong winds. We have also made a simple assumption in formulating the guidance equation for circular path following, such that the guidance logic is simplified while still providing acceptable performance. In addition, we have implemented a longitudinal guidance and control element that caters for the dynamics of powerless flight. Following well-established aerospace control design procedures [9], we have chosen to separate our design into two modes: an inner control loop that provides aircraft dynamic stability, and an outer guidance loop that generates the required acceleration and position commands to follow a path.

In the lateral guidance mode, a reference point P_{ref} is first selected on the desired trajectory, and this reference point is then used to generate a lateral acceleration command. As shown in Fig. 5, P_{ref} is located a distance L_1 ahead of the vehicle and, at each point in time, a circular path (dotted line) can be defined by the position of L_1, the vehicle position, and tangential to V, the aircraft velocity vector.

The acceleration required to follow the instantaneous circular segment, for any radius R, is then given by:

$$a_{s_{cmd}} = \frac{V^2}{R} = 2\frac{V^2}{L_1} sin\eta \tag{7}$$

Thus, the guidance logic will tend to rotate the aircraft such that its velocity direction will always approach the desired path at an angle that is proportional to the relative distance between vehicle and path. For following a straight line, we can model the vehicle kinematics as shown in Fig. 6.

Now, consider the UAV in a straight glide at an arbitrary position relative to the path between waypoints P_1 and P_3, and at a heading ψ (Fig. 6a). Given the aircraft velocity and position in the North, East reference frame, and the angular

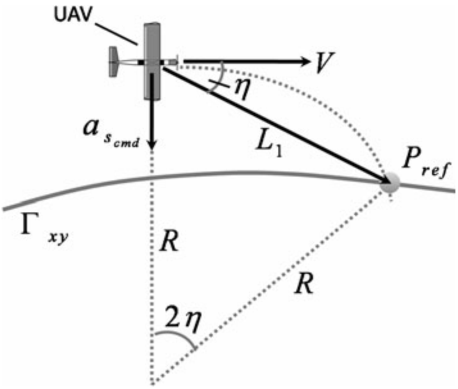


Fig. 5 Diagram showing the lateral guidance logic

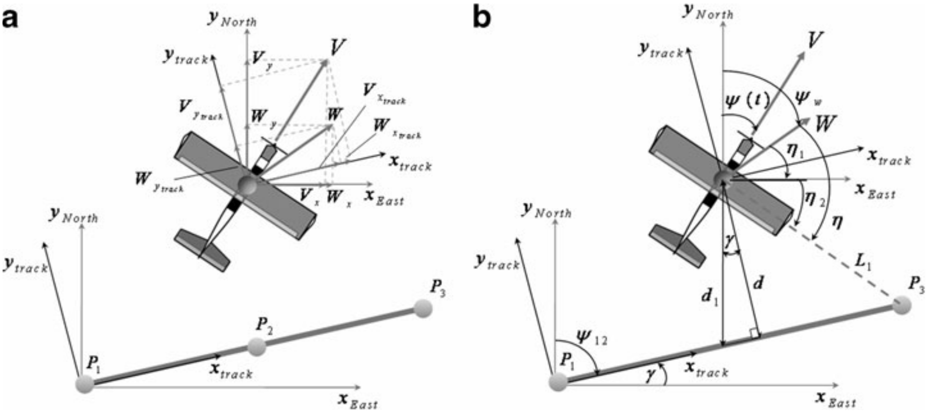


Fig. 6 Vehicle kinematics for straight line following, showing **a** the relationship between the aircraft velocity V and wind velocity W, and **b** the relationship between the aircraft bearing $\psi(t)$, the path bearing ψ_{12}, the wind bearing ψ_w, and the angle η. In addition, the relationship between the cross-track error d, the hypotenuse d_1 and the look-ahead distance L_1 is also shown

measurements defined in Fig. 6b, we can obtain the position and velocity components in the $\{x_{track}, y_{track}\}$ reference frame by:

$$\begin{aligned} \overrightarrow{V}_{track} &= T_\phi \overrightarrow{V} \\ \overrightarrow{W}_{track} &= T_\phi \overrightarrow{W} \end{aligned} \tag{8}$$

where the rotation matrix

$$T_\psi = \begin{bmatrix} cos(\psi_{12} - \pi/2) & -sin(\psi_{12} - \pi/2) \\ sin(\psi_{12} - \pi/2) & cos(\psi_{12} - \pi/2) \end{bmatrix} \tag{9}$$

The cross-track velocity can then be written as:

$$\begin{aligned} \dot{y}_{track} = {} & V_{y_{track}} + W_{y_{track}} \\ & = -Vsin(\psi(t) - \psi_{12}) - Wsin(\psi_w - \psi_{12}) \end{aligned} \tag{10}$$

And assuming η is small, we get:

$$sin\eta = \eta_1 + \eta_2 \tag{11}$$

and

$$\begin{aligned} \eta_1 &\approx \frac{d}{L_1} \\ \eta_2 &\approx \frac{\dot{d}}{V} \end{aligned} \tag{12}$$

where the cross-track velocity $\dot{y}_{track}$ has been relabeled as $\dot{d}$, and d is the cross-track error. If N and E are the North and East coordinates, we can obtain d by letting

$$d_1 = N_{aircraft} - tan(\gamma) E_{aircraft} \tag{13}$$

and

$$d = d_1 cos\gamma \tag{14}$$

Combining Eq. 7 to 12, we obtain:

$$a_{s_{cmd}} = 2\frac{V}{L_1}\left(\dot{d} + \frac{V}{L_1}d\right) \tag{15}$$

For following an arc of circumference, we can model the vehicle kinematics as shown in Fig. 7.

Here, the angles η_1 and η_2 are assumed to be small, but η_3 is not necessarily small,

$$\eta_1 \approx 0, \eta_2 \approx 0, |\eta_3| >> 0. \tag{16}$$

As shown in [13], we can estimate

$$sin\eta_3 = \frac{L_1}{2R} \tag{17}$$

and define

$$c \equiv cos\eta_3 \approx \sqrt{1 - \left(\frac{L_1}{2R}\right)^2} \tag{18}$$

Then, using small angle assumptions for η_1 and η_2, we can show

$$a_{s_{cmd}} = \frac{2V^2}{L_1}\{\eta_1 cos\eta_3 + \eta_2 cos\eta_3 + sin\eta_3\} \tag{19}$$

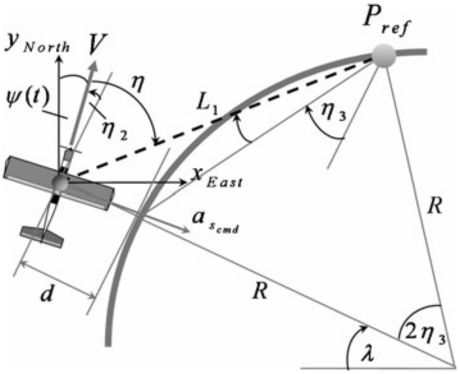
with

$$\eta_1 \approx \frac{d}{L_1}cos\eta_3\ ,\ \dot{d} = Vsin\eta_2 \approx V\eta_2 \tag{20}$$

and applying Eqs. 17 and 18, Eq. 19 becomes

$$\frac{2V^2}{L_1} = \frac{2V^2c^2}{L_1^2}d + \frac{2Vc}{L_1}\dot{d} + \frac{V^2}{R} \tag{21}$$

Fig. 7 Vehicle kinematics for circular path following



Now, if we assume that a good yaw damper can be designed to damp the aircraft Dutch roll motion and reduce the sideslip to zero, then we can neglect the second term on the R.H.S of Eq. 21 to obtain;

$$a_{s_{cmd}} = \frac{2V^2}{L_1} sin\eta = \frac{2V^2c^2}{L_1^2} d + \frac{V^2}{R} \tag{22}$$

To convert the acceleration to a desired roll command and simplify calculations, we assume that the aircraft maintains sufficient lift to balance weight, even though banked at an angle ϕ. This gives

$$Lcos\phi = W = mg, \;\; Lsin\phi = ma \tag{23}$$

and

$$\phi_d = tan^{-1}\left(\frac{a_s}{g}\right) \tag{24}$$

In actual fact, a gliding aircraft will never overcome gravity, but will descend with a vertical velocity $\dot{h}$; this is an inherent limitation in the forced landing problem. Notice also that here we have not included additional terms in the equation for wind effects, as we have done with the case of following a straight line. The reason is that in this case the vehicle ground speed (as a surrogate for inertial velocity) is used for V in Eq. 7 at each instant in generating the acceleration command. Since the ground speed is a function of the airspeed and wind speed, the guidance logic accounts for the inertial velocity changes due to wind and adapts to the situation accordingly.

Following the design of the outer loop guidance logic, we then proceeded to use a PID control approach as it is widely used in UAV applications [9], and would also provide a seamless integration with an off-the-shelf UAV flight computer for field testing. Here, the difference between the desired and actual roll angle is used to produce an error signal that activates the aileron control servo. The controller also includes roll rate feedback for improved damping, and saturation limits on the outputs to avoid instability. A classical yaw damper was also included in the design which has enabled the aircraft to follow straight and circular paths with greatly reduced oscillations. In the longitudinal guidance mode, the difference between the desired and actual aircraft flight path angles is used to generate the desired pitch angle, which in turn controls the elevators for longitudinal path following. The desired path angle is composed of $\gamma_{\phi_{0,f}}$, the path angles corresponding to the sections of helices in the 3D path, and γ_{line}, the path angle of the line segment.

To cater for the effects of wind while gliding, we use the well-known MacCready theory (discussed in [15]). From the speed polar diagram (Fig. 8), we see that to counter a headwind, a glide speed above V_{bg} must be selected; this increases the ground speed and allows the aircraft to penetrate further through the air. Similarly, the glide speed must be reduced below V_{bg} when flying downwind to avoid overshooting the target waypoint. In a tailwind, the starting point on the horizontal axis (airspeed) is shifted left by a distance equal to the magnitude of the wind speed, and a line is drawn from this point tangential to the curve to obtain the desired glide speed. For a headwind the starting point is shifted to the right, for sinking air it is shifted upwards on the vertical axis, and for rising air it is shifted downwards. Using the speed polar, one can also determine the speed-to-fly in different combinations of vertical and horizontal winds when both exist. Finally, to convert the speed-to-fly into

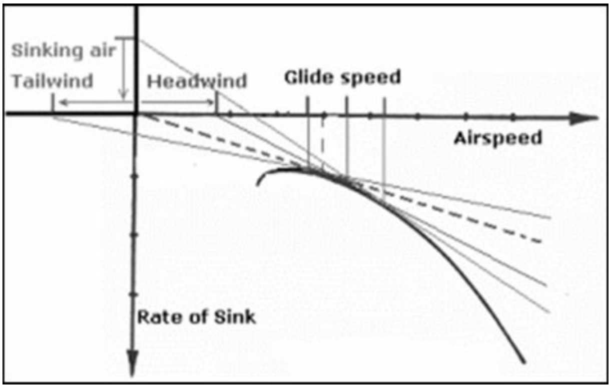


Fig. 8 Using the speed polar diagram while gliding in winds (Image obtained from [12])

the desired path angle, we simply take the inverse tangent of the slope corresponding to the tangent line.

In designing the inner loop pitch controller, we have used the pitch error to control the desired elevator deflection and the pitch rate to provide additional damping. As before, PID controllers are used to regulate the system, and limiters are placed on the outputs to prevent saturating the elevator servomotor, which we have assumed to possess similar characteristics to that controlling the ailerons. We have also implemented scheduling of the PID and L_1 gains such that the required path following performance could be met. For testing purposes, this performance is specified as having a horizontal (lateral) and vertical (longitudinal) cross-track error at the approach point of no greater than 2 m (approx. 6.56 ft), and a maximum vertical and horizontal deviation of no greater than 30 m (approx. 100 ft) on average. These upper and lower bounds are commonly accepted as the performance standard for general aviation aircraft [1].

2.3 Results and Discussion

A total of 128 simulations have been performed to gain an initial understanding of the efficacy of our planning, guidance and control algorithms. In each case, the aircraft initial altitude and wind conditions were allowed to vary, while the initial and final aircraft headings and positions were kept constant. A Monte Carlo simulation using a larger input set will be performed at a later date. For these experiments, we have also assumed perfect knowledge of the wind conditions and no errors in the sensor readings. A sample of the test data is included in Table 1.

Table 1a shows the initial and desired final aircraft positions used by the path planner in constructing a trajectory from an altitude of 1640 ft to 500 ft in calm conditions, while Table 1b shows the positions for a descent from 850 ft to 500 ft in winds, with a maximum wind speed of 7 m/s. Note that as we are concerned with gliding flight, the required airspeed at the approach point is absorbed by the flight path angle requirement at this point.

As depicted in Fig. 9, in planning a path for a forced landing from a high initial altitude (Table 1a), the algorithm is able to generate the required number of helix spirals to “bleed off” the excess altitude, before joining the spirals with an arc-line-arc Dubins path (solid black line). This ensures that constraints on the flight path

Table 1 Sample test data for the path planning and path following algorithms

	Initial cond.	Final cond.
a) No wind		
x (m)	−199	885
y (m)	37	133
z (ft)	1640	500
ψ (deg)	10	90
γ (deg)	0	−6
b) With wind		
x (m)	−199	885
y (m)	37	133
z (ft)	850	500
ψ (deg)	10	90
γ (deg)	0	−6

angle are satisfied and prevents excessive stress to the UAV structure. The horizontal and vertical track errors at the approach point are 0.3 m and 1.3 m, which are well within our stated tolerances and comparable to the results obtained for other flight path angles. The apparent difference in altitudes between the aircraft and the path at the start is due to the planning algorithm rounding the required number of helix spirals to the nearest complete (360°) spiral turn, in order to preserve the desired initial heading. As can be seen from the diagram, the aircraft is still able to converge onto the path at Point A despite this offset in altitudes. However, at Point B the aircraft descends below the path and reaches a maximum vertical deviation of 600 ft at Point C (the horizontal error at this point was approximately 9 m), but recovers to intercept the approach point with the errors stated above. The poor longitudinal path following performance in the first halve of the descent may be attributed to the assumption that the airspeed remains constant at the best glide speed, as it is not possible to predict in advance what the actual airspeed due to the control actions may be. In an actual descent, the airspeed is allowed to vary and this gives rise to a

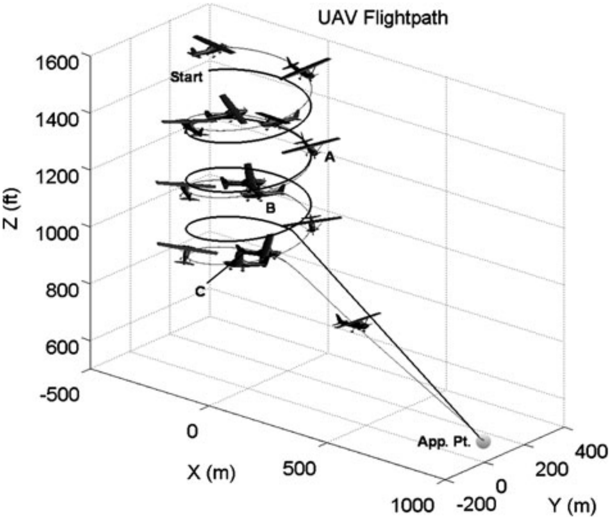


Fig. 9 Path planning and following for a forced landing from a high initial altitude in nil wind conditions

non-uniform loss in altitude. In addition, an aircraft rolled into a continuous banking motion will also experience some amount of yawing motion called sideslip, no matter how good the yaw damper may be, and this in turn increases the altitude lost. Thus the amount of loss in altitude factored into the path planning equations is ideal at best, and does not fully take into account the associated loss in altitude due to varying airspeeds and other atmospheric effects. Hence, the current solution relies on the path following algorithm being robust to these uncertainties in guiding the aircraft to the desired approach point. A possible alternative is to increase the path angle of the initial helices to more closely match that of the straight segment, and/or increase the radii of the helices such that the number of spirals is reduced. These will help reduce the amount of altitude loss due to sideslips and a prolonged banking action.

Next, we show the performances of our path following algorithm in winds (Table 1b) and compare the results with those obtained using the original path following algorithm in [13], hereby referred to as the unmodified non-linear guidance (UNG) algorithm. Two different wind scenarios are chosen for illustration. Figure 10a shows the aircraft able to follow the desired path (solid black line) in a 6 m/s South-South-Westerly wind (green arrows), while Fig. 10b shows a similar case albeit with the wind coming from the South-South-East. As shown in the top halves of Fig. 10c and d, using our path following algorithm, the lateral error (blue line) at the approach point was 1.8 m and 1.2 m respectively; compare this with the performance of UNG, which has lateral errors (red dashed line) of 14.1 m and 6 m for the two different wind conditions. Although UNG did not include a longitudinal path following component, we have nonetheless also plotted the vertical track error to show what might have transpired had that lateral guidance algorithm, coupled with our longitudinal guidance algorithm, been used to follow the path. As shown in the lower half of Fig. 10c and d, the vertical track errors for our guidance algorithm is approximately 1.2 m and 1.5 m respectively for the two wind conditions, while that for the case of UNG coupled with our longitudinal guidance algorithm is 1.5 m and 2.4 m. Thus it can be clearly seen that our path following algorithm outperforms the UNG algorithm. In addition, the average lateral and longitudinal path errors for both cases are well within 100 ft, which as mentioned earlier is commonly accepted as the maximum allowable path deviation for general aviation aircraft.

From the simulations, we have also observed that the path following algorithm is able to contain the errors at the approach point, and within the stated tolerances, for wind speeds not exceeding 7 m/s. In stronger winds, these errors can degrade to >20 m horizontally, and up to 5 m vertically, or the aircraft may lose control and crash. A possible explanation for these factors is that since we are using an unpowered, scaled model of a real aircraft as our UAV platform, the small size and weight (5.55 kg) of the aircraft, as well as the limited thrust available means that it cannot achieve the necessary control authority to overcome strong winds and gusts. We have also noticed from simulations that the vertical track error at the approach point is >7 m in sinking air of 1 m/s. Once again, this relatively poor performance may be due to the structural and aerodynamic factors stated above. However, when compared to our previous work as discussed in Section 2, we find that the performances of our current algorithms are far superior. We believe that this improved performance may be largely due to the fact that we have allowed the airspeed and flight path angle to vary to counter different wind scenarios. In addition, the simulation results presented here indicate that the path planning, guidance and

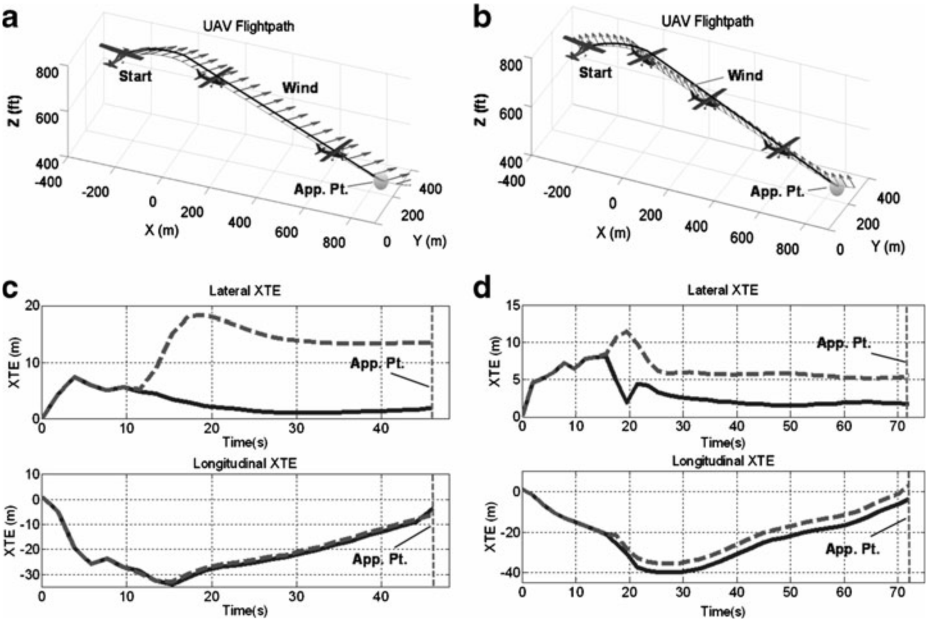


Fig. 10 Path planning and following in ambient winds, showing **a** Aircraft response in 6 m/s wind from SSW; **b** Aircraft response in 6 m/s wind from SSE; **c** Horizontal and vertical track errors for case a; and **d** Horizontal and vertical track errors for case b

control techniques developed are suitable for further assessment in Monte Carlo simulations and even flight trials.

2.4 Future Work

In the future, we plan to enhance the control elements to be robust to uncertainties in sensor errors and wind measurements. Secondly, we will experiment with different techniques to reduce the vertical track error when the aircraft is following a helix spiral, and several options have already been proposed in the preceding section. Thirdly, we desire to extend the path planning component to include a re-planning capability for cases where the efficacy of the original plan is reduced or nullified. This could be due to winds that are simply too strong for the aircraft to overcome, or when a better landing site is identified as the aircraft nears the ground. These modular enhancements will also be progressively flight tested.

3 Multi-criteria Decision Making

One of the most important aspects in the initial stages of a forced landing is to make the right decision regarding which site to land on and how to approach the chosen landing site. These two aspects are closely related to the multi-criteria decision analysis and the path planning, guidance and control component of the overall approach, respectively. This section will shed light on the main concepts behind the challenging decision-making process, which in reality is continuously validated and updated throughout much of the descent if new information should yield a more appropriate landing site.

3.1 Multiple Criteria

According to the Australian Civil Aviation Safety Authority's latest Visual Flight Rules flight guide [1] there are seven "S" criteria to selecting the best site for a forced landing, in addition to the critical factor of wind strength and direction. These are:

- Size
- Shape
- Surface
- Slope
- Surrounding (risk to nearby infrastructure/population)
- S(c)ivilisation (proximity to aid)
- Sun (reduced visual capability)

These are the primary elements which a human pilot use when making decisions on the selection of a preferred landing site. When applied in the context of UAVs, many of these factors still hold their significance, and a number of other variables also come into consideration which are not explicitly stated for piloted aircraft. These include the aircraft dynamics, the uncertainty of sensor data and wind estimation, etc.

Also to be considered is the geometrical relationship between the various candidate sites. As the aircraft descends, the number of landing site options will rapidly decrease. Thus, it is generally better to plan the approach towards several possible sites in close proximity than to one that is isolated, as this keeps multiple landing site options open for as long as possible. This is important so as to have several options if obstacles are detected on the candidate landing sites at lower altitudes.

The number of structures and the population density that lies in the descent path to each site must also be accounted for if possible, as it would be safer to fly over empty terrain than a populated area, in case further mishaps occur. These points, along with other factors which remain to be identified, will be evaluated to reach an optimal, verifiable decision on which candidate landing site the aircraft will aim for.

Further investigations will be conducted in order to identify any other influences that affect this decision process, possibly including surveys and simulations involving experienced pilots and/or UAV controllers.

3.2 Multiple Objectives

The complexity of the forced landing decision process due to multiple criteria is further increased by multiple objectives that must be met. In many cases, these objectives may be conflicting, and thus compromises must be made to accommodate the achievement of the most critical objective/s.

According to the Civil UAV Capability Assessment [16], in the event of an emergency landing the UAV needs to be able to respond according to the following objectives in the following order:

1. Minimize expectation of human casualty;
2. Minimize external property damage;

3. Maximize the chance of aircraft survival; and
4. Maximize the chance of payload survival.

In many scenarios, the best landing site for meeting objectives 3 and 4 may compromise the more important objectives 1 and/or 2, or vice versa. This complex trade-off process between the risks and uncertainties involved with each possible choice is an example of multiple objectives that the system must evaluate between and is what makes this problem difficult.

3.3 Decision Considerations

The descent planning and decision making modules will initially have preplanned contigency plans from map data to give fast, reflex responses to emergencies that guide the aircraft towards known landing sites, or large flat areas based on the slope map data.

The guidance and control module (discussed in the previous section) will constantly make estimates of the wind speed and direction, which will be taken as input for decision making. The aircraft dynamics will also be accounted for and necessary constraints applied when judging the feasibility of a decision.

The decision making algorithm must run in real time; in the highly dynamic and critical situation of an aircraft forced landing, it is important that the decision making component is able to respond to changing circumstances in a timely manner. The decision making algorithm will also need to be deterministic. Even in uncertain conditions, where the site identification phase is not able to provide candidate landing sites with an adequate level of confidence, or in situations where there are no desirable alternatives, a single decision outcome must always be reached.

As the aircraft descends, the landing site identification and classification module will continually analyse the terrain the aircraft is flying over. Possible landing sites, buildings, and roads will be identified, including the associated uncertainties of objects in each map. Armed with this information, the decision making module will then be able to continually validate and update its decision in real-time. Another consideration arising from this is the length of time committed to a decision, which essentially means that the longer one particular alternative has been committed to, the higher the cost to switch to another alternative. Thus, to a certain extent, past decision outcomes must be taken into consideration.

It is expected that uncertainties will reduce as the aircraft descends, however the options available will also reduce. It may be possible that an initially selected landing site will eventually be deemed unsuitable by the site selection module, and an alternative must be sought after. It is the responsibility of the decision making module to be prepared for such situations by maximizing the number of alternative choices available.

3.4 Decision Making Methods

From the literature review, it was concluded that there are essentially two broad classes of multi-criteria decision analysis methods; one follows the outranking philosophy and builds a set of outranking relations between each pair of alternatives, then aggregates that according to some suitable technique. The other essentially involves determining utility/value functions for each criterion, and finding out the 'utility' of each alternative based on each criterion, then aggregating those with a suitable technique to find the overall utility of the alternative.

Many of the existing techniques are not designed for 'decision making'; rather they are intended as 'decision aid' methods, and hence some only generate additional information to aid the human decision maker, who makes the final decision. Decision making is in many ways also a subjective matter. As discussed earlier, in most cases there is no 'best' decision, and it is subject to the preference of the human decision makers. Due to the nature of the forced landing problem, where decisions made could potentially lead to damage to property or even harm life, it is critical then that the decision making system to be developed must be based on justifiable and generally accepted preference data. This means that the technique chosen should require preference data that is clear and understandable by people who don't understand the mathematics of method, and also that the technique should be as transparent as possible for purposes of accountability. Additional requirements used to evaluate the various techniques include the ability to handle uncertainty in terms of input data, and the assumptions made regarding the decision problem. A number of the techniques are currently under trial, such as PROMETHEE [17] and MAUT [18]. PROMETHEE is an outranking method that requires relatively simple preference data in terms of criterion weights and preference functions. MAUT, which is based on Expected Utility Theorem [19] makes the assumption of independence, meaning that only the probability distribution of risks of individual criterion are considered, and they don't affect each other. This may be unrealistic for the forced landing scenario, yet it can be addressed by using Fuzzy Choquet Integrals [20], which addresses synergy and redundancy between criteria.

The technique of most interest does not readily fit in to either of the main families of multicriteria decision analysis methods, namely, the Decision Rules Approach [21], and the one specifically described here is the Dominance-based Rough Set Approach (DRSA) [22]. This method takes samples of decisions made by human experts, and analyses them to determine the minimum set of decision rules expressed in the form of "if, then" statements. These statements are then used to evaluate the alternatives in the multi-criteria decision problem, and aggregated with an appropriate aggregation technique such as the Fuzzy Net Flow Score [23]. There is the capacity to deal with inconsistent preference information from the human decision makers by using the rough sets, and fuzzy sets can be implemented to address uncertainty in the input data. This method is the most transparent and understandable of all those investigated so far, and further comparisons will be made between these decision making techniques under consideration.

4 Conclusions

In this article we have presented in detail the design, implementation and simulation of the path planning, guidance and control strategies for an automated forced landing system for UAVs. We have also introduced the multi-criteria decision making approach that aims to emulate human pilot thought processes in the event of a forced landing. Simulated results of the planning, guidance and control module demonstrate the ability of the gliding aircraft to follow the prescribed path in winds, with average path deviation errors that are comparable to or even better than that of manned, powered aircraft. Although the path planning, guidance and control strategies were derived from existing work in the literature, we have enhanced these algorithms and added further functionality to suit the case of an unpowered fixed-wing UAV forced landing. Some examples include the extension of Dubins curves to 3D while accounting for aircraft dynamics, and encapsulating wind information in the guidance logic, rather than treating wind as an adaptive element to be overcome by the control system. However, many complex decision making problems still remain to be investigated. These problems involve multiple conflicting objectives, and it is often true that no dominant alternative will exist that is better than all other alternatives. Generally it is impossible to maximize several objectives simultaneously, and hence the problem becomes one of value tradeoffs. The tradeoff issue often requires subjective judgement of the decision maker, and there may be no right or wrong answers to these value questions. Currently, we are in the process of testing several multi-criteria decision making techniques, and simultaneously integrating strategies between the path planning, guidance and control and the decision making modules before progressing to flight trials. It is believed that the automated UAV forced landing approach presented here will allow the progression of this technology from the development and simulation stages through to a prototype system that can demonstrate its effectiveness to the UAV research community.

Acknowledgements The work presented in this paper is sponsored by the Australian Postgraduate Award, the Queensland University of Technology (QUT) Vice-Chancellors Top-Up Award and the CSIRO ICT Scholarship Top-Up Award.

References

1. CASA, Visual Flight Rules Guide, 2nd ed. Civil Aviation Safety Authority Australia, Canberra (2007)
2. Fitzgerald, D., Mejias, L., Eng, P., Liu, X.: Towards flight trials for an autonomous uav emergency landing using machine vision. In: Australasian Conference on Robotics and Automation, Brisbane, Australia, December 2007
3. Redelinghuys, C.: A flight simulation algorithm for a parafoil suspending an air vehicle. J. Guid. Control Dyn. **30**, 791–803 (2007)
4. Froeschner, C.P.: UAS—Flying safety. http://findarticles.com/ (2008). Accessed 27 Nov 2008
5. Fitzgerald, D.: Candidate landing site selection for uav forced landings using machine vision. Ph.D. dissertation, Queensland University of Technology. School of Engineering Systems (2007)
6. Mejias, L., Fitzgerald, D., Eng, P., Liu, X.: Aerial vehicles. I-Tech Education and Publishing, 2009, ch. Forced Landing Technologies for Unmanned Aerial Vehicles: Towards Safer Operations, pp. 415–440
7. Australian Bureau of Meteorology: The wind across Australia. http://www.bom.gov.au/ (2007). Accessed 24 June 2007
8. Eng, P., Mejias, L., Fitzgerald, D., Walker, R.: Simulation of a fixed-wing uav forced landing with dynamics path planning. In: Australasian Conference on Robotics and Automation, Brisbane, Australia, December 2007
9. Nelson, R.C.: Flight Stability and Automatic Control, 2nd ed. McGraw-Hill Higher Education, New York (1998)
10. LaValle, S.M.: Planning Algorithms. Cambridge University Press, New York (2006)
11. Dubins, L.E.: On curves of minimal length with a constraint on average curvature with prescribed initial and terminal positions and tangents. Am. J. Math. **79**, 471–477 (1957)
12. Ambrosino, G., Ariola, M., Ciniglio, U., Corraro, F., Pironti, A., Virgilio, M.: Algorithms for 3-d uav path generation and tracking. In: 45th IEEE Conference on Decision & Control, pp. 5275–5280. San Diego, CA (2006)

13. Park, S.: Avionics and control system development for mid-air rendezvous of two unmanned aerial vehicles. Ph.D. dissertation, Department of Aeronautics and Astronautics, M.I.T (2004)
14. Park, S., Deyst, J., How, J.P.: Performance and lyapunov stability of a nonlinear path-following guidance method. J. Guid. Control Dyn. **30**, 1718–1728 (2007)
15. Brandon, J.: Forced landing procedures. http://www.auf.asn.au/emergencies/forcedlanding.html. Accessed 25 March 2007
16. Cox, T.H., Nagy, C.J., Skoog, M.A., Somers, I.A.: Civil uav capability assessment. NASA, Tech. Rep., draft Version (2004)
17. Brans, J.-P., Mareschal, B.: Promethee methods. In: Figueira, J., Greco, S., Ehrgott, M. (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys. Springer, New York (2005)
18. Dyer, J.S.: Maut—multiattribute utillity theory. In: Figueira, J., Greco, S., Ehrgott, M. (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys. Springer, New York (2005)
19. von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, Princeton (1944)
20. Grabisch, M., Roubens, M.: Application of the choquet integral in multicriteria decision making. In: Grabisch, T.M.M., Sugeno, M. (eds.) Fuzzy Measures and Integrals - Theory and Applications, pp. 348–374. Physica Verlag, Würzburg (2000)
21. Greco, S., Matarazzo, B., Slowinski, R.: Decision rule approach. In: Figueira, J., Greco, S., Ehrgott, M. (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys. Springer, New York (2005)
22. Graco, S., Matarazzo, B., Slowinski, R.: Rough approximation by dominance relations. Int. J. Intell. Syst. **17**, 153–171 (2002)
23. Bouyssou, D.: Ranking methods based on valued preference relations: a characterization of the net flow method. Eur. J. Oper. Res. **60**(1), 61–67 (1992). Available http://www.sciencedirect.com/science/article/B6VCT-48NBG9R-1TT/2/3590a17dce5c9186172b2099f2c50f47