Autonomous Control and Path Planning for Autorotation of Unmanned Helicopters

Thanan Yomchinda
Graduate Research Assistant

Joseph F. Horn
Associate Professor

Jack W. Langelaan
Assistant Professor

Department of Aerospace Engineering
The Pennsylvania State University
University Park, PA

ABSTRACT

The objective of this work is to develop an autonomous autorotation landing system for an unmanned helicopter. Three-dimensional autorotation flight paths are obtained from a parametric optimization method (developed in previous work) to guide the aircraft through the three phases of a successful autorotation: entry, descent, and flare to landing at a desired location. In this study, the path planning algorithms are integrated with a flight control law and demonstrated in high fidelity simulations. Details of the autonomous autorotation trajectory following control law and analytical autorotation flight path implementation are presented. Simulation results demonstrate the feasibility of tracking complex three-dimensional autorotation trajectories from engine failure to safe touchdown at a pre-specified landing location.

INTRODUCTION

The use of autonomous rotorcraft for tasks such as resupply and casualty evacuation places a significant focus on safety and vehicle recovery after system failure. This paper focuses on autorotation.

Pilot workload during autorotation is significant. Descent rate is typically high, and this must be arrested just prior to touchdown by extracting energy from the freely spinning rotor. The flare maneuver is safety critical and correct timing is key: flare too early and rotor speed drops below the level required to sustain flight before touchdown can occur; flare too late and the descent rate will be too high for safe touchdown. Both scenarios can lead to loss of the vehicle and onboard personnel. To the authors' knowledge autonomous rotorcraft do not currently have the capability to control flight through landing during autorotation, although research on autorotation flight planning and control has been done in the past.

Researchers have investigated emergency planning for fixed wing aircraft, addressing issues such as power loss[1] or land/go around[2]. A significant amount of research has been conducted on rotorcraft trajectory planning for both powered flight[3] and autorotation[4-8]. A method based on Dubin's curves is described by Holsten et al.[9] Abbeel[10] describes a machine learning approach to autorotation. Aponso et al. noted three important points[6]: (1) vehicle parameters such as weight can have a strong influence on the computed trajectory; (2) a critical improvement would be the ability to continuously update trajectories to account for performance differences as well as errors in trajectory following; (3) optimal trajectory planning can be used to expand the V-h envelope. Real-time trajectory planning remains a difficult problem on the computational hardware likely to be available on an autonomous rotorcraft: system dynamics are complex, system models may not be accurate (autorotation only occurs after a failure), external disturbances (i.e. wind) may be large, and the touchdown site may not be flat.

In a previous paper, Tierney and Langelaan focused on the flare phase by computing the Safe Landing Set, the region in the helicopter state space from which a safe flare to touchdown is guaranteed to exist[11]. The states incorporated in the safe landing set include flare initiation point (distance to and height above the desired touchdown point), airspeed, descent rate and rotor speed. The safe landing set thus defines the region in the state space where flare initiation should occur. Any descent that passes through the safe landing set thus has a guaranteed solution to the problem of flare trajectory planning.

A follow-on paper by Yomchinda et al. addressed the problem of trajectory planning during the descent phase. The objective of the trajectory planning algorithm was to ensure that the helicopter enters the safe landing set [12] near some desired landing spot. A path planning algorithm was developed to take the aircraft from the point of engine failure to a location where it can safely perform a flare to landing at a pre-determined location. In this work, the Dubin's car approach[13,14] was modified by incorporating two additional parameters: first, descent rate was explicitly modeled to permit three-dimensional trajectories; second, constant acceleration and variable turn rate were used along the segments of the flight path instead of the constant velocity / constant turn rate approach typically used in standard Dubin's paths. Bank angle and horizontal airspeed were used as parametric controls to define a modified Dubin's path in the horizontal plane that achieves the desired constraints on final position, heading, and horizontal velocity. Vertical translation was then solved using a map of the aircraft's quasi-steady autorotation performance where rotor speed provides an additional control to regulate descent rate.

The autorotation path planning algorithm was formulated as an optimization problem. The rotor speed, bank angle, and horizontal airspeed were parametric controls used to minimize the final altitude error.

Acceleration along the path permitted the flight condition to change (e.g. from maximum range glide at the beginning of descent to the target condition which guarantees safe landing as the safe landing set is entered). It was also found that deceleration along the autorotation path could significantly increase the range of the autorotation path (effectively decrease the glide slope) and allow the helicopter to reach more distant landing spots.

The algorithm was applied to a point mass model of a generic utility helicopter. The path-planning algorithm showed promise in providing accurate and efficient trajectory planning from entry to autorotation to a safe landing condition near some a desired landing site. There were some issues with local minima, but ad hoc schemes in estimating initial guesses for the optimization algorithms were found to effectively address this issue.

This paper describes the implementation of an end-to-end autonomous autorotation system and discusses results of simulations conducted using a nonlinear FLIGHTLAB model of a generic utility helicopter. Figure 1 displays the schematic of the complete system in this study. An inner loop flight control system is built using a

model following / dynamic inversion control scheme to provide stability to the rotorcraft. This control law is designed to achieve consistent response characteristics across the flight envelope[15,16]. Ideal response models, command filters, are designed to meet the specified response characteristics in ADS-33 Handling Quality standards. A path following control system is built as the pilot control loop for the stabilized rotorcraft to command the aircraft to follow trajectory solutions. An autorotation trajectory generation is implemented to provide autorotation trajectory for specified cases. The path planner presented in this paper combines the descent path algorithm[12] and flare trajectory algorithm[11] developed in previous studies, and an entry phase path generating algorithm is introduced. The entry phase allows the vehicle to recover its rotor speed while it is generating a descent phase path solution.

The remainder of this paper provides details of path planner implementation, autonomous control system design and simulation environment. Subsequently, simulation results are discussed and conclusions are presented.
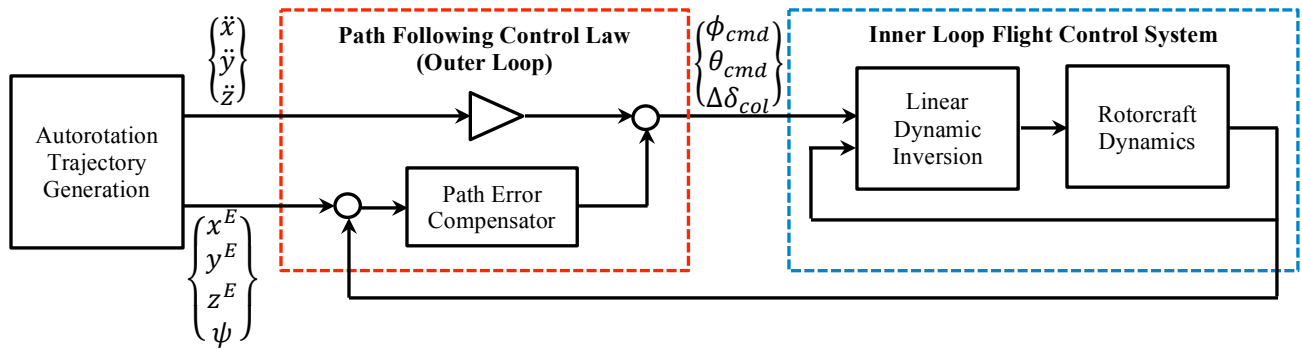

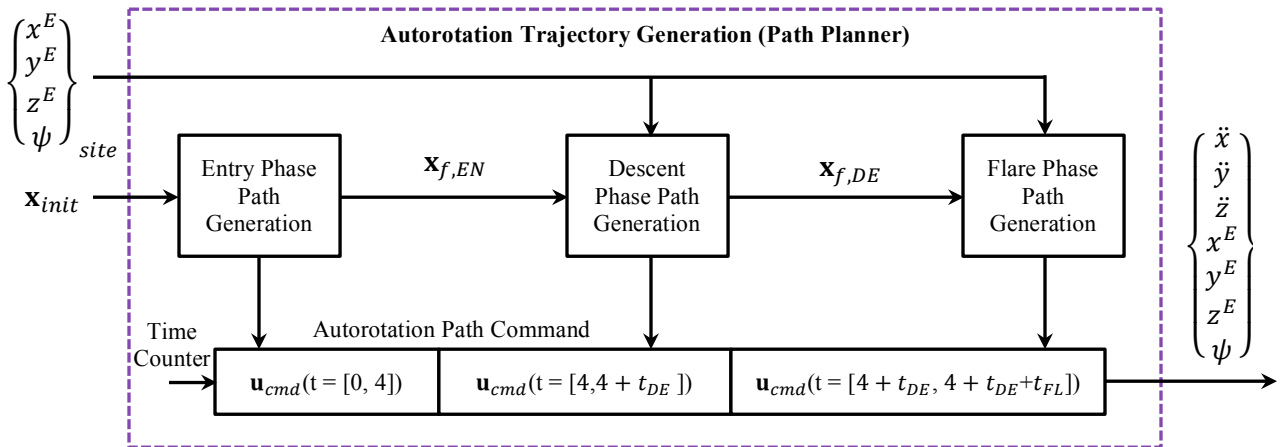**Figure 1. Schematic of autonomous control and autorotation path planning system**

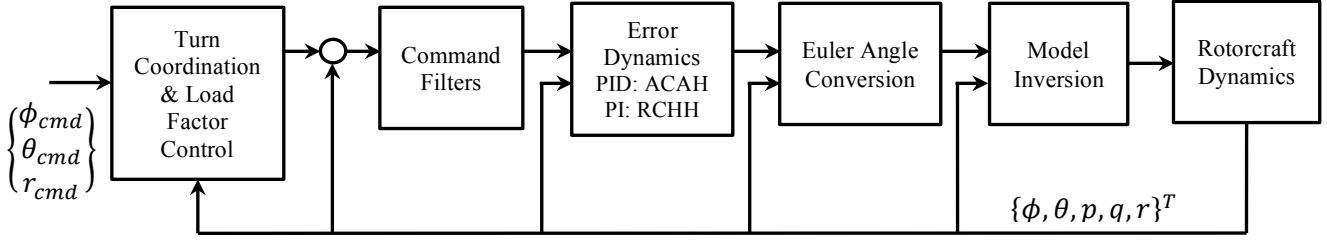**Figure 2. Schematic of path planner system**

Figure 3. Schematic of dynamic inversion control system (inner loop flight control)

PATH PLANNING

Figure 2 shows a schematic of the autorotation path planning system. The algorithm uses measured information of the initial condition at engine failure (velocity, position and heading) and a pre-determined terminal condition at touchdown (location and heading) to generate a feasible autorotation trajectory. It is assumed that feasible landing sites have been determined and stored prior to the autorotation event. The full trajectory consists of three parts (entry, descent and flare) which are generated from three different algorithms; the final condition of entry phase is required by the descent phase algorithm and the final condition of descent phase is required by the flare phase algorithm. Therefore, the paths are generated in the order: entry phase, then descent phase, and then flare phase. The terminal condition of the descent phase is based on a pre-calculated safe landing set (as discussed in reference 11).

The states representing vehicle condition for the autorotation trajectory generating algorithm include the vehicle's velocity components in a local coordinate frame, position in North-East-Down coordinate frame, heading and rotor speed.

$$\mathbf{x} = \{\dot{x}\ \dot{y}\ \dot{z}\ x^E\ y^E\ z^E\ \psi\ \Omega\}^T \quad (1)$$

where the superscript E indicates the North-East-Down (NED) inertial coordinate frame. The position, velocity, and acceleration variables without superscript E indicate values with respect to a local coordinate frame where the x-axis is aligned with the aircraft heading (projection of the aircraft x-body axis onto the horizontal plane).

The path generation algorithm produces a set of commands for the flight controller in the form of the aircraft position, heading and required acceleration as a function of time. Thus, the objective of the algorithm is to produce the following command vector for all three phases of autorotation:

$$\mathbf{u}_{cmd} = \{\ddot{x}_{cmd}, \ddot{y}_{cmd}, \ddot{z}_{cmd}, x^E_{cmd}, y^E_{cmd}, z^E_{cmd}, \psi_{cmd}\}^T \quad (2)$$

The flight controller (as discussed in a following section) attempts to follow this path using the standard primary flight controls on the helicopter.

The path generating algorithms for all three flight phases make use of a point mass model of the rotorcraft. The use of this simplified model is desirable to generate paths in a computationally efficient manner. However, the point mass model has to be well tuned to match the true autorotation performance of the aircraft. In this study, the point mass model is tuned to match autorotation performance of the FLIGHTLAB® model used in the simulations.

The point mass model was developed in a previous study[12], and includes the following assumptions: 1) The orientation of the rotor tip path plane is defined by the pitch and roll attitudes. There is no real loss in generality here, as the ϕ_R and θ_R controls technically define the rotor orientation and not the vehicle orientation. 2) Attitude changes are achieved instantaneously. 3) The aircraft is in zero sideslip coordinated flight, where the heading is aligned with the flight path, $\dot{y} = 0$. 4) Atmospheric conditions are assumed constant.

The point mass model consists of three rotorcraft dynamics states of horizontal velocity, descent rate and rotor speed, four states of NED position and heading, and three control inputs of rotor orientation (roll and pitch attitudes) and rotor thrust coefficient.

$$\begin{Bmatrix} \ddot{x} \\ \ddot{z} \\ \dot{\Omega} \\ \dot{x}^E \\ \dot{y}^E \\ \dot{z}^E \\ \dot{\psi} \end{Bmatrix} = F\left(\begin{Bmatrix} \dot{x} \\ \dot{z} \\ \Omega \\ x^E \\ y^E \\ z^E \\ \psi \end{Bmatrix}, \begin{Bmatrix} \phi_R \\ \theta_R \\ C_T \end{Bmatrix} \right) \quad (3)$$

Simulated flight test data of power-off autorotation flights were extracted from the FLIGHTLAB® model to tune the point mass model. The rotor dynamics equation of the simplified model is given by:

$$I_R\Omega\dot{\Omega} = -\frac{2}{\eta}\rho(\pi R^2)(\Omega R)^3\left(\frac{1}{8}\sigma c_d(1 + 4.7\mu^2) + C_T\lambda\right) \quad (4)$$

The drag coefficient parameter, c_d, was tuned so that the point mass model descent rate matched that of the full dynamic model in various quasi-steady autorotation conditions. It was found that making the profile drag coefficient a cubic function of the rotor tip blade Mach number resulted in reasonable correlation.

$$c_d = a_3 M^3 + a_2 M^2 + a_1 M + a_0 \geq 0.01 \quad (5)$$

where M is the rotor tip blade Mach number and a_0, a_1, a_2, a_3 are approximation constants.

Figure 4 shows the correlation between the descent rate of FLIGHTLAB model and that of tuned point mass model at various autorotation operating conditions. Data points in the figure correspond to autorotation operating conditions at airspeed between 40 and 240 ft/s, acceleration between -3 and 1 ft/s^2, rotor speed between 24 and 29 rad/s and bank angle of 0, 15 and 30 deg.
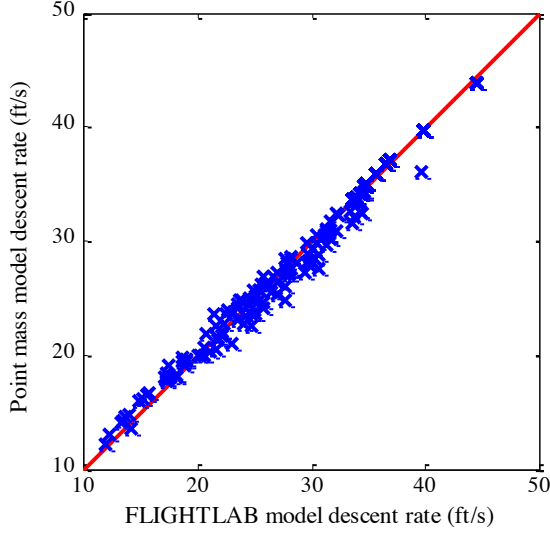

Figure 4. Correlation between point mass model and FLIGHTLAB model

A. Entry Phase

The entry phase path is designed to be a four-second maneuver, in which the rotorcraft hold constant heading and attempts to recover rotor speed and enter a steady descent condition. As with all phases, entry is formulated as trajectory optimization problem. In the case of entry, there are no terminal constraints. The cost function is formulated to minimize control effort (changes in thrust and pitch attitude) and rotor speed drop during the four second flight, and the altitude loss at the end of the four second flight. The aircraft flight dynamics are incorporated using Euler integration of the point mass model:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \dot{\mathbf{x}}_k \Delta t \quad (6)$$

where $\dot{\mathbf{x}}_k$ is computed by using the point mass model

The parameter optimization problem is:

Minimize $C(\{C_{T,0...N}, \theta_{R,0...N}\})$
$$\begin{aligned} &= W_1 \sum (C_{T,i+1} - C_{T,i})^2 \\ &+ W_2 \sum (\theta_{R,i+1} - \theta_{R,i})^2 \\ &+ W_3 \sum (\Omega_i - \Omega_{ref})^2 + W_4 (h_f - h_0)^2 \end{aligned} \quad (7)$$

Subject to

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \dot{\mathbf{x}}_k \Delta t \quad (8)$$

$$\dot{\mathbf{x}}_k = G(\mathbf{x}_k, \{C_{T,k}, \theta_{R,k}\}) \quad (9)$$

$$-0.1g \leq \ddot{x}_i \leq 0.1g \quad (10)$$

$$80\%\Omega_{ref} \leq \Omega_i \leq 105\%\Omega_{ref} \quad (11)$$

where $G()$ represents helicopter dynamics and Ω_{ref} is a nominal rotor speed for the generic utility helicopter, $\Omega_{ref} = 27$ rad/s

The dimension of parameter optimization problem is reduced by discretizing the flight with a bigger time step (small number of nodes) and using a cubic spline fit to generate control inputs for smaller time steps between nodes.

$$\{C_{T,0...N}, \theta_{R,0...N}\} = S(\{C_{T,0...K}, \theta_{R,0...K}, t_{0...N}\}) \quad (12)$$

where $S()$ is a cubic spline fit function, N is number of smaller time step node, K is number of bigger time step node and $N > K$

The parameter optimization problem was formulated in MATLAB® software. The MATLAB® FMINCON function is used to solve the parameter optimization problem. The parameters, $\mathbf{x}_k$ and $\dot{\mathbf{x}}_k$, at each time step from the path solution are stored and used as a path command for the entry phase.

B. Descent Phase

The descent phase autorotation trajectory generating algorithm was developed in a previous study[12] to find the path solution from the final condition of entry phase, $\mathbf{x}_{f,EN}$, to the condition closest to the center of safe landing set[11] for the pre-selected landing site. The safe landing set location for zero wind at a flat landing location is located 300 ft above and 600 ft downrange from the landing site. The center of safe landing set is obtained by following equation.

$$\mathbf{x}_{d,DE} = \begin{cases} 80 \\ 0 \\ n/a \\ x_{site}^E - 600 \cos \psi \\ y_{site}^E - 600 \sin \psi \\ z_{site}^E - 300 \\ \psi_{site} \\ \Omega_{ref} \end{cases} \quad (13)$$

The descent phase path consists of three segments (a turn, a straight flight and another turn segment). The optimal solution of descent phase is found by solving a trajectory optimization problem from the initial condition $\mathbf{x}_{f,EN}$ to the desired final condition, $\mathbf{x}_{d,DE}$. The descent phase is formulated using terminal constraints on the final position and heading. As discussed in detail in reference 12, a modified Dubin's path was developed to calculate the flight path in the horizontal plane that meets the terminal constraints on x, y position and heading. The path is different from the classic Dubin's path in that the vehicle is allowed to vary speed along each segment at a

constant rate of acceleration/deceleration. The turns are performed at constant bank angle but variable speed, so the turn segments are spiral paths instead of circular arcs. The three-dimensional path is ultimately described by seven parametric control variables:

$$\mathbf{u}_{DE} = \{\dot{x}_1 \ \phi_1 \ \Omega_1 \ \dot{x}_2 \ \Omega_2 \ \phi_3 \ \Omega_3\}^T \quad (14)$$

where the subscripts 1, 2 and 3 indicate that variable corresponds to the first turn, straight and second turn segments respectively. The parametric controls describe the bank angle, acceleration, and rotor speed along each segment.

It is assumed the helicopter is in a quasi-steady autorotation along each segment in which the rotorcraft operates at constant horizontal acceleration, constant bank angle and constant rotor speed. The descent rate in this quasi-steady condition can be described by a nonlinear mapping of the form:

$$\dot{z} = W(\dot{x}, \ddot{x}, \phi, \Omega) \quad (15)$$

The complete mapping of quasi-steady state descent is obtained by using the tuned point-mass model for the complete operating region. The maximum acceleration and deceleration of the aircraft are allowed to be ±0.1g (3.217 ft/s²). The rotor speed is limited to be within 24.3 rad/s and 28.35 rad/s, with a nominal operating point of 27 rad/s. The minimum horizontal velocity for autorotation is assigned at 80 ft/s (implying minimum airspeed of ~47.4 knots) and the maximum autorotation horizontal velocity is limited to 240 ft/s (implying maximum airspeed of ~142.2 knots). The bank angle is constrained to be within +/-30º.

The descent height along each segment of the autorotation is obtained by integrating the descent rate along the path. The final descent height is used to determine the deviation of final altitude from the target height. The autorotation trajectory planning problem is written as a parameter optimization problem to find a set of parametric control variables that minimize the cost function:

$$\text{Minimize} \quad C(\mathbf{u}_{DE}) = C_z + C_{\Omega_1} + C_{\Omega_2} + C_{\Omega_3} \quad (16)$$

Subject to

$$-0.1g \leq \ddot{x}_i \leq 0.1g \quad (17)$$

$$\mathbf{u}_{DE,min} \leq \mathbf{u}_{DE} \leq \mathbf{u}_{DE,max} \quad (18)$$

where

$$C_z = (z_0 + h_1 + h_2 + h_3 - z_{des})^2 / W_z^2 \quad (19)$$

$$C_{\Omega_i} = (\Omega_i - \Omega_{ref}) / W_\Omega^2 \quad (20)$$

$$i = 1,2,3 \quad (21)$$

where Ω_{ref} is the desired rotor speed in autorotation flight.

The parameter optimization problem was formulated in MATLAB® software. The MATLAB® FMINCON function is used to solve the parameter optimization problem to find the solution of parametric control variables which can be used to obtain the path parameters and commands at every time step in the descent phase path. Figure 5 illustrates the descent phase path solution for a specific scenario.
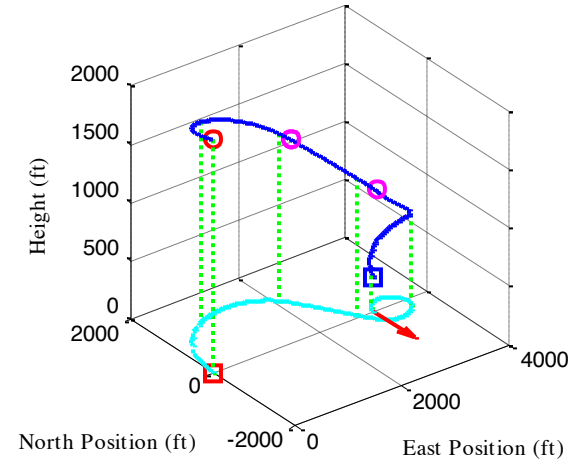

Figure 5. Trajectory result from the descent phase path generation

C. Flare Phase

The descent phase ends in the safe landing set and a flare trajectory computed using the algorithm described in Tierney's work[11] is followed for the remainder of the flight.

For completeness the flare trajectory generation is briefly summarized. Flare is assumed to begin from trimmed autorotation flight and heading is assumed to be nearly constant (i.e. there are no lateral deviations from the path and it is a "straight in" approach). The Safe Landing Set for the desired touchdown point computed by Tierney is the set of all trimmed autorotation flight conditions and flare initiation points that result in safe landing. This safe landing set was computed by finding the optimal flare trajectory from a candidate flare initial condition to safe touchdown. If this optimal trajectory exists then the candidate initial condition is a member of the safe landing set.

A challenge in computing optimal flare trajectories is that landing time is unknown. However, the touchdown *altitude* is known, and equations of motion are recast in terms of altitude rather than time. Altitude is thus the independent variable for the purpose of flare trajectory optimization, and touchdown time is computed as part of the process.

To improve computational tractability the trajectory is found by parameterizing inputs C_T and θ_R using a cubic

spline, and the flare trajectory optimization problem is now a parameter optimization problem.

A trajectory following controller (discussed in the next section) follows the flare trajectory to touchdown.

AUTONOMOUS CONTROL

The control architecture consists of an inner loop, which tracks aircraft attitude commands, and an outer loop which follows path commands, as shown in Figure 1. The inner loop flight control law is a model following scheme with dynamic inversion.

The inner loop control law follows attitude commands in the roll and pitch axes and a yaw rate commands in the yaw axis. The collective input comes from the outer loop control law and simply passes through inner loop so that the inner loop inverse model can account for collective coupling effects.

A second-order command filter is used for the roll and pitch attitude control. The desired response is described by:

$$\ddot{\phi}_c + 2\zeta_\phi \omega_\phi \dot{\phi}_c + \omega_\phi^2 \left(\phi_c - \phi_{cmd} \right) = 0 \quad (22)$$

$$\ddot{\theta}_c + 2\zeta_\theta \omega_\theta \dot{\theta}_c + \omega_\theta^2 \left(\theta_c - \theta_{cmd} \right) = 0 \quad (23)$$

The yaw axis uses a first-order command filter:

$$\tau_r \dot{r}_c + \left(r_c - r_{cmd} \right) = 0 \quad (24)$$

In this study, a natural frequency of 2 rad/s and damping ratio of 0.7 were selected for the roll and pitch command filters. For yaw axis, a time constant of 0.4 seconds was chosen.

Feedback compensation used to track the difference between the desired responses and the current responses of the aircraft is tuned to meet disturbance rejection and cross-coupling requirements. A proportional-integral-derivative (PID) compensator is used to minimize the tracking error for the roll and pitch attitudes to achieve ACAH response type. For the yaw axis, a RCAH response type, a proportional-integral (PI) compensator is used. The "pseudo-commands" are calculated as follows:

$$\begin{bmatrix} \ddot{\phi}_D \\ \ddot{\theta}_D \\ \dot{r}_D \end{bmatrix} = \begin{bmatrix} \ddot{\phi}_c \\ \ddot{\theta}_c \\ \dot{r}_c \end{bmatrix} + \begin{bmatrix} K_{P,\phi}\tilde{\phi} + \int K_{I,\phi}dt + K_{D\phi}\dot{\tilde{\phi}} \\ K_{P,\theta}\tilde{\theta} + \int K_{I,\theta}dt + K_{D,\theta}\dot{\tilde{\theta}} \\ K_{P,r}\tilde{r} + \int K_{I,r}\tilde{r}dt \end{bmatrix} \quad (25)$$

where $\tilde{\phi} = \phi_c - \phi$; $\tilde{\theta} = \theta_c - \theta$; $\tilde{r} = r_c - r$

The error dynamics for the roll and pitch attitudes can be represented by the following equations.

$$\ddot{e} + K_D\dot{e} + K_P e + K_I \int e = 0 \quad (26)$$

$$s^3 + K_{D,\phi}s^2 + K_{P,\phi}s + K_{I,\phi} = 0 \quad (27)$$

$$s^3 + K_{D,\theta}s^2 + K_{P,\theta}s + K_{I,\theta} = 0 \quad (28)$$

$$(s^2 + 2\zeta\omega_n s + \omega_n^2)(s + p) = 0 \quad (29)$$

In the yaw axis, the error dynamics can be represented by the following equations.

$$\dot{e} + K_P e + K_I \int e = 0 \quad (30)$$

$$s^2 + K_{P,\theta}s + K_{I,\theta} = s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \quad (31)$$

The feedback gains of the PID compensator (K_P, K_I, K_D) in Eq.41, 42 and 45 are obtained by assigning the compensator parameters (natural frequency, damping ratio and real pole) for each axis. In this study, a natural frequency of 3 rad/s, damping ratio of 0.9 were selected for roll and pitch axes. For yaw axis, a natural frequency of 3 rad/s, damping ratio of 0.8 were selected. The real poles for roll and pitch axes were placed at -0.75.

The inversion model is scheduled to account for changes in aircraft dynamics with airspeed. In this study, the design flight conditions include hover to the forward flight speed of 160 knots, at sea level standard.

$$\begin{bmatrix} \delta_{lat} \\ \delta_{long} \\ \delta_{ped} \end{bmatrix} = \left[B(V_T) \right]^{-1} \cdot \left\{ \begin{bmatrix} \dot{p}_D \\ \dot{q}_D \\ \dot{r}_D \end{bmatrix} - \left[A(V_T) \right] \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right\} \quad (32)$$

An Euler angle conversion scheme is used to convert the Euler angle angular acceleration commands into body axis angular acceleration commands for the model inversion.

$$\dot{p}_D = \ddot{\phi}_D - \frac{\left(\begin{matrix} \ddot{\theta}_D \sin\phi\sin\theta + \dot{r}_D \sin\theta + \dot{\theta}\dot{\phi}\cos\phi\sin\theta \\ + \dot{\psi}\dot{\theta}\cos\phi + \dot{\psi}\dot{\phi}\sin\phi\sin\theta\cos\theta \end{matrix} \right)}{\cos\phi\cos\theta} \quad (33)$$

$$\dot{q}_D = \frac{\left(\ddot{\theta}_D + \dot{r}_D \sin\theta + \dot{\psi}\dot{\phi}\cos\theta \right)}{\cos\phi} \quad (34)$$

A turn coordination control law is added as an outer loop feedback control using a computed yaw rate approach. The control law calculates the yaw rate command to achieve a desired lateral acceleration using:

$$r_{cmd} = \left(a_{y,cmd} + w \cdot p + g \cdot \sin\phi \cdot \cos\theta \right)/u \quad (35)$$

In forward flight, a zero a_y command can be used to keep the aircraft coordinated. The turn coordination control law is phased out as airspeeds drop below 45 knots.

The outer loop path following control law is designed to calculate the inner loop commands in order to maneuver the aircraft along the desired path. Decoupled control laws are used for the longitudinal, lateral and vertical axes. The acceleration components in the local frame longitudinal, lateral and vertical axes are used to generate pitch angle, bank angle and collective commands respectively. A PID compensator is added to minimize the tracking path error in each axis.

The following simplified equations of helicopter motion (expressed in its local coordinate frame) are used to formulate the control law:

$$\ddot{x} = g \tan \theta \approx g\theta \qquad (36)$$

$$\ddot{y} = g \, (\cos \theta)^{-1} \tan \phi$$
$$\approx g(1 + \tan^2 \phi_{Ref}) \, (\phi - \phi_{Ref}) \qquad (37)$$

$$\ddot{z} = g - \frac{T}{m}\cos \theta \cos \phi \approx Z_{\delta_{col}}\Delta\delta_{col} \qquad (38)$$

where $\Delta\delta_{col} = \delta_{col} - \delta_{col,trim}$ and $Z_{\delta_{col}}$ is the control derivative modeling vertical acceleration response to collective.

The primary commands for pitch angle, bank angle and collective input can be found from simplified equations of helicopter motion.

$$\theta_{cmd} = \frac{\ddot{x}_{cmd}}{g} \qquad (39)$$

$$\phi_{cmd} = \phi_{Ref} + \frac{\ddot{y}_{cmd}}{g(1 + \tan^2 \phi_{Ref})} \qquad (40)$$

$$\Delta\delta_{col} = \frac{\ddot{z}_{cmd}}{Z_{\delta_{col}}} \qquad (41)$$

where ϕ_{Ref} is a constant bank angle command which the path planner assigns for coordinated flight in the turning segments of the autorotation trajectory.

An additional set of PID feedback compensators are used to compensate errors in flight path. The compensators use the commanded position in North-East-Down (NED) coordinates ($x^E_{cmd}, y^E_{cmd}, z^E_{cmd}$) and commanded heading (ψ_{cmd}) generated by the path planner and provides compensation to regulate the position and heading errors. The ideal transfer functions of position displacement from command inputs can be derived from the simplified equations of helicopter motion and the command filters.

$$\frac{X}{\Theta_{cmd}}(s) = \frac{\omega_\theta^2}{s^2 + 2\zeta_\theta\omega_\theta s + \omega_\theta^2}\left(\frac{g}{s^2}\right) \qquad (42)$$

$$\frac{Y}{\Phi_{cmd} - \Phi_{ref}}(s) = \frac{(1 + \tan^2 \phi_{Ref})\,\omega_\phi^2}{s^2 + 2\zeta_\phi\omega_\phi s + \omega_\phi^2}\left(\frac{g}{s^2}\right) \qquad (43)$$

$$\frac{Z}{\Delta\delta_{col}}(s) = \frac{Z_{\delta_{col}}}{s^2} \qquad (44)$$

The feedback gains in the PID compensator are selected to place the poles of feedback transfer functions to achieve desired damping ratios and natural frequencies of displacement error feedback system on each axis. The path following control laws combine the primary command and error compensating command to generate command inputs for the inner loop control system as shown in Fig.1.

$$\theta_{cmd} = \frac{\ddot{x}_{cmd}}{g} + K_{P,x}e_x + K_{I,x}\int e_x + K_{D,x}\dot{e}_x \qquad (45)$$

$$\phi_{cmd} = \phi_{ref} + \frac{\ddot{y}_{cmd}}{g(1 + \tan^2 \phi_{Ref})}$$
$$+ K_{P,y}e_y + K_{I,y}\int e_y + K_{D,y}\dot{e}_y \qquad (46)$$

$$\Delta\delta_{col} = \frac{\ddot{z}_{cmd}}{Z_{\delta_{col}}} + K_{P,z}e_z + K_{I,z}\int e_z + K_{D,z}\dot{e}_z \qquad (47)$$

where e_x, e_y, e_z are components of the position error vector in the local coordinate frame as calculated by:

$$\begin{Bmatrix} e_x \\ e_y \\ e_z \end{Bmatrix} = \begin{bmatrix} \cos\psi_{cmd} & \sin\psi_{cmd} & 0 \\ -\sin\psi_{cmd} & \cos\psi_{cmd} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x^E - x^E_{cmd} \\ y^E - y^E_{cmd} \\ z^E - z^E_{cmd} \end{Bmatrix} \qquad (48)$$

The derivative, $Z_{\delta_{col}}$, is obtained from linearization of the rotorcraft model and was found to vary moderately with airspeed; however, it was found that a constant value gave reasonable controller performance. The poles of feedback transfer functions for longitudinal and lateral axes are placed at $-0.7242 \pm 0.7628j$, $-0.6116 \pm 0.6363j$ and -0.1284 by the selection of the feedback gains (K_P, K_I, K_D). For vertical axis, the poles of feedback transfer functions are placed at $-1.786 \pm 0.2854j$ and -0.428 by the selection of the feedback gains (K_P, K_I).

The power-off autorotation flight in this work assumes that the rotorcraft is in zero sideslip coordinated flight which the heading is always aligned with the flight path. The pedal command of inner loop control system is used to minimize the vehicle's sideslip angle.

$$a_{y,cmd} = K_\beta\beta \qquad (49)$$

SIMULATION ENVIRONMENT

The rotorcraft flight dynamics model used in this study is a non-linear FLIGHTLAB® simulation model of a generic utility helicopter.

The FLIGHTLAB® rotorcraft model is a bare-airframe helicopter model with an engine model. The engine model allows RPM variation due to the time lag response of engine power; no engine governor is included. Model inputs are main rotor collective blade pitch angle, lateral cyclic pitch angle, longitudinal cyclic pitch angle and tail rotor collective blade pitch angle (rad). An actuator model is included to represent the time lag of control surface due to mechanical hardware. A first-order actuator model with time constant of 0.02 is used in each control surface. The key properties of this generic utility helicopter model are shown in Table 1.

Table 1: Properties of generic utility helicopter

Description	Value
Aircraft weight (lbf)	16285.1
Main rotor: Articulated rotor	
number of blades	4
nominal speed (rad/s)	27
radius (ft)	27
Tail rotor:	
number of blades	4
nominal speed (rad/s)	124.62
radius (ft)	5.5
Control surface range: [min, max]	
main rotor lateral cyclic (deg)	[-8, 8]
main rotor longitudinal cyclic (deg)	[-12.5, 16.3]
main rotor collective (deg)	[5.0, 25.9]
tail rotor collective (deg)	[0.0, 36.5]

The model following/inversion control law uses 9th order linearized models extracted from the bare-airframe FLIGHTLAB® simulation model in various operating points. The linearized model's states are the nine rigid body fuselage states (3 velocities, 3 angular rates and 3 Euler angles).

The model following/inversion control law is implemented in FLIGHTLAB® Control System Graphical Editor (CSGE) by using the control law design scheme presented in previous section. The path following control law mentioned previously is also transitioned to FLIGHTLAB® using CSGE.

A power-off condition is simulated by disconnecting engine power from the rotor system during flight.

RESULTS AND DISCUSSION

Several cases of varying initial altitude, distance to touchdown point and initial speed were tested to examine performance. Results of two representative cases are presented here.

Initial conditions (as well as touchdown conditions) of the two cases are given in Table 2. In both cases the aircraft is in straight and level flight heading north. Case 1 is a high altitude, high-speed engine failure condition with the landing site located 5700 feet to the north-west, and the desired heading at landing is 60 degrees from north. Case 2 is a lower altitude, lower speed condition with the landing site located 2200 feet to the north-east. Again the desired heading at landing is 60 degrees from north.

Table 2: Summary of results

Case	Initial condition	Touchdown condition
1	altitude: 2000 ft distance: 5700 ft airspeed: 100 kt	groundspeed: 28.8 kt vertical speed: 10.49 ft/s rotor speed: 23 rad/s
2	altitude: 900 ft distance: 2200 ft airspeed: 60 kt	groundspeed: 29 kt vertical speed: 10.5 ft/s rotor speed: 21 rad/s

Figures 4 through 6 show results for Case 1. Figure 4 shows the flight path from the moment of engine failure to touchdown. State and control histories are shown in Figures 5 and 6. The time from engine failure to touchdown is approximately 65 seconds, and the actual touchdown point is 18 feet away from the desired touchdown point. Note that there are no disturbances acting on the aircraft and the error in touchdown location is due to inability of the controller to follow the desired path exactly. However, 18 feet is well within one rotor diameter.

In this case the aircraft speed decreases steadily during the descent. Figure 5 shows that the trajectory following controller performs quite well: maximum deviation from the desired path is about 30 feet (again this is within one rotor diameter). Figure 6 shows that actual bank angle does not track commanded bank angle exactly. This occurs because the Dubin's path solution assumes that turn rate can change instantaneously: this corresponds to step changes in bank angle.

Figures 7 through 9 show results for Case 2. Figure 7 shows the flight path from the moment of engine failure to touchdown; state and control histories are shown in Figures 8 and 9. The time from engine failure to touchdown is approximately 28 seconds and average descent rate is 32 ft/s. In this case the actual touchdown point was 8 feet away from the desired touchdown point.

In this case airspeed does not drop significantly until later in the descent. As a result, rotor speed drops significantly. Rotor speed does increase enough to permit safe flare and touchdown, however.

Again one can see step changes in commanded bank during the descent and corresponding overshoot in aircraft bank response. Improving this will be a focus of future work: post-processing of the trajectory to remove the step changes in bank angle should improve performance significantly.

CONCLUSIONS

This paper has described a method for end-to-end control of autonomous helicopter autorotation. The objective of the method is to autonomously guide the helicopter from the moment of engine failure to safe landing at a specified touchdown point. The system includes an inner loop flight controller to stabilize the vehicle and follow a desired trajectory and a trajectory generator that computes safe, dynamically feasible trajectories to the desired touchdown point.

The trajectory planner divides flight into three phases: entry, which brings the helicopter from engine failure to a steady descent condition; descent, which guides the helicopter to a state from which flare can be safely

initiated; and flare, which guides the helicopter to safe landing.

The inner loop flight controller uses dynamic inversion and model following control to stabilize the vehicle. An outer loop trajectory following controller uses a combination of feed forward control inputs and a PID controller to reduce path error.

Results of two representative cases are presented to show the utility of the approach: a high, fast case with significant distance from engine failure to the touchdown point and a low, slow case with significantly shorter distance to touchdown. In both cases safe touchdown occurs: descent rate is approximately 10.5 ft/s and ground speed is approximately 30 knots. Trajectory following is quite good, with maximum error during descent of about 30 feet (distance from desired touchdown point is significantly less: 18 feet for the high fast case and 10 feet for the low, slow case). Error in trajectory following appears to be due to step changes in commanded bank angle in the desired trajectory: this causes overshoot in actual bank and some lag in trajectory following during turning flight.

REFERENCES

[1]   Atkins, E. M., Portillo, I. A., and Strube, M. J., "Emergency Flight Planning Applied to Total Loss of Thrust," Journal of Aircraft, Vol. 43, No. 4, 2006, pp. 1205-1216.

[2]   Sprinkle, J., Eklund, J. M. and Sastry, S. S., "Deciding to Land a UAV Safely in Real Time," American Control Conference, Portland, OR, 2005.

[3]   Bottasso, C. L., Croce, A., Leonello, D., and Riviello, L., "Rotorcraft Trajectory Optimization with Realizability Considerations," Journey of Aerospace Engineering, Vol. 18, No. 3, Jul 2008, pp. 146-155.

[4]   Lee, A. Y., Bryson Jr., A. E. and Hindson, W. S., "Optimal Landing of a Helicopter in Autorotation," Journal of Guidance, Control and Dynamics, vol. 11, no. 1, January-February 1988, pp. 7-12.

[5]   Johnson, W., "Helicopter Optimal Descent and Landing and Power Loss," Technical Memorandum TM 73244, NASA Ames Research Center, May 1977.

[6]   Aponso, B. L., Bachelder, E., N., and Lee, D., "Automated Autorotation for Unmanned Rotorcraft Recovery," American Helicopter Society International Specialist Meeting on Unmanned Rotorcraft, Chandler, AZ, 2005.

[7]   Aponso, B. L., Lee, D., and Bachelder, E., N., "Evaluation of a Rotorcraft Autorotation Training Display on a Commercial Flight Training Device," Proceeding of American Helicopter Society 61st Annual Forum, Grapevine, TX, 2005.

[8]   Carlson, E. B., "Optimal Tiltrotor Aircraft Operations during Power Failure," Ph.D. Dissertation, Aerospace Engineering and Mechanics Dept., Univ. of Minnesota, Minneapolis, MN, 1999.

[9]   Holsten, J., Loechelt, S., and Alles, W., "Autonomous Autorotation Flights of Helicopter UAVs to Known Landing Sites," American Helicopter Society 66th Annual Forum, Phoenix, AZ, 2010.

[10] Abbeel, P., Coates, A., Hunter, T., and Ng, A. Y., "Autonomous Autorotation of an RC Helicopter," International Symposium on Robotics, 2008.

[11] Tierney, S., and Langelaan, J. W., "Autorotation Path Planning Using Backwards Reachable Set and Optimal Control," American Helicopter Society 66th Annual Forum, Phoenix, AZ, 2010.

[12] Yomchinda, T., Horn, J.F., and Langelaan, J. W., "Flight Path Planning for Descent Phase Helicopter Autorotation," AIAA Guidance, Navigation and Control Conference, Portland, OR, 2011.

[13] Dubins, L. E., "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," American Journal of Mathematics, Vol. 79, 1957, pp., 497-516.

[14] Dubins, L. E., "On plane curves with curvature," Pacific Journal of Mathematics, Vol. 11, No. 2, 1961, pp. 471-481.

[15] Yomchinda, T., Horn, J.F., and Cameron, N., "Integrated Flight Control Design and Handling Qualities Analysis for a Tilt Rotor Aircraft," AIAA Atmospheric Flight Mechanics Conference, Chicago, IL, August 10-13, 2009.

[16] Horn, J.F., Guo, W., and Ozdemir, G.T., "Use of Rotor State Feedback to Improve Closed Loop Stability and Handling Qualities," Proceedings of the American Helicopter Society 66th Annual Forum, Phoenix, AZ, May 11-13, 2010.
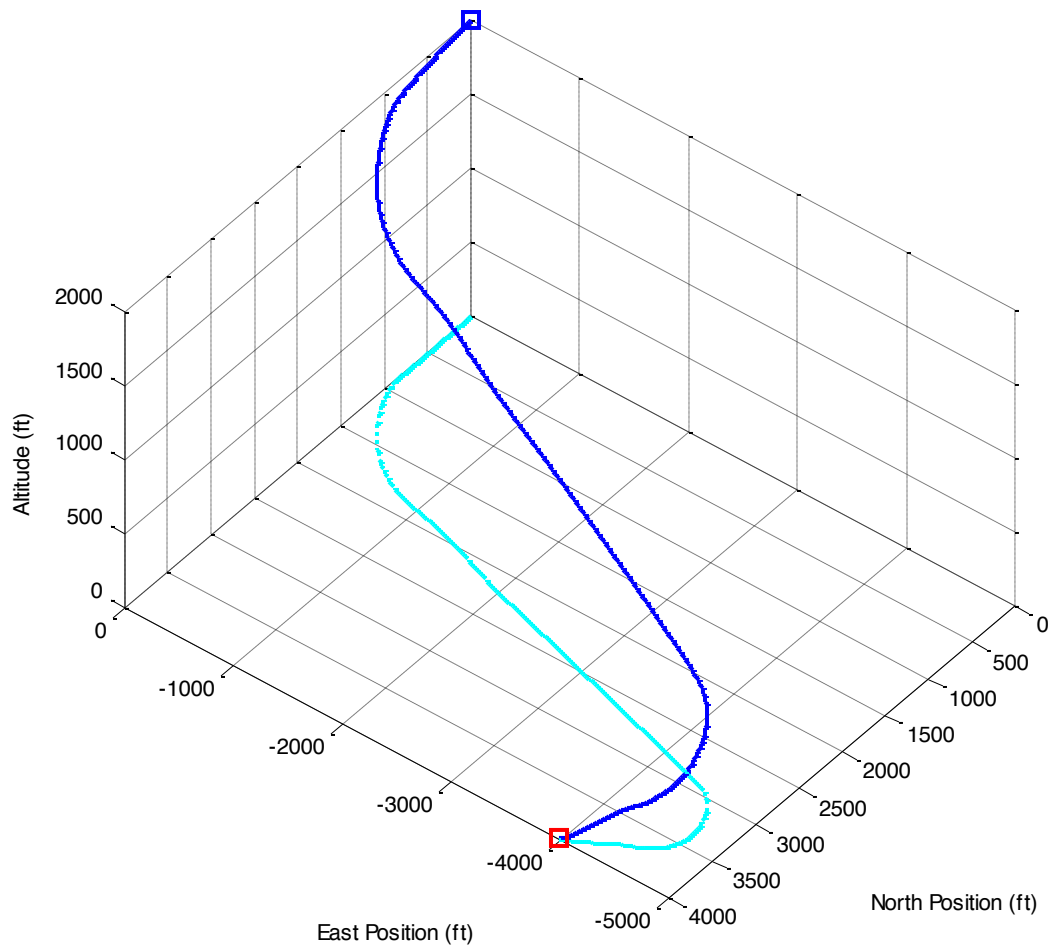
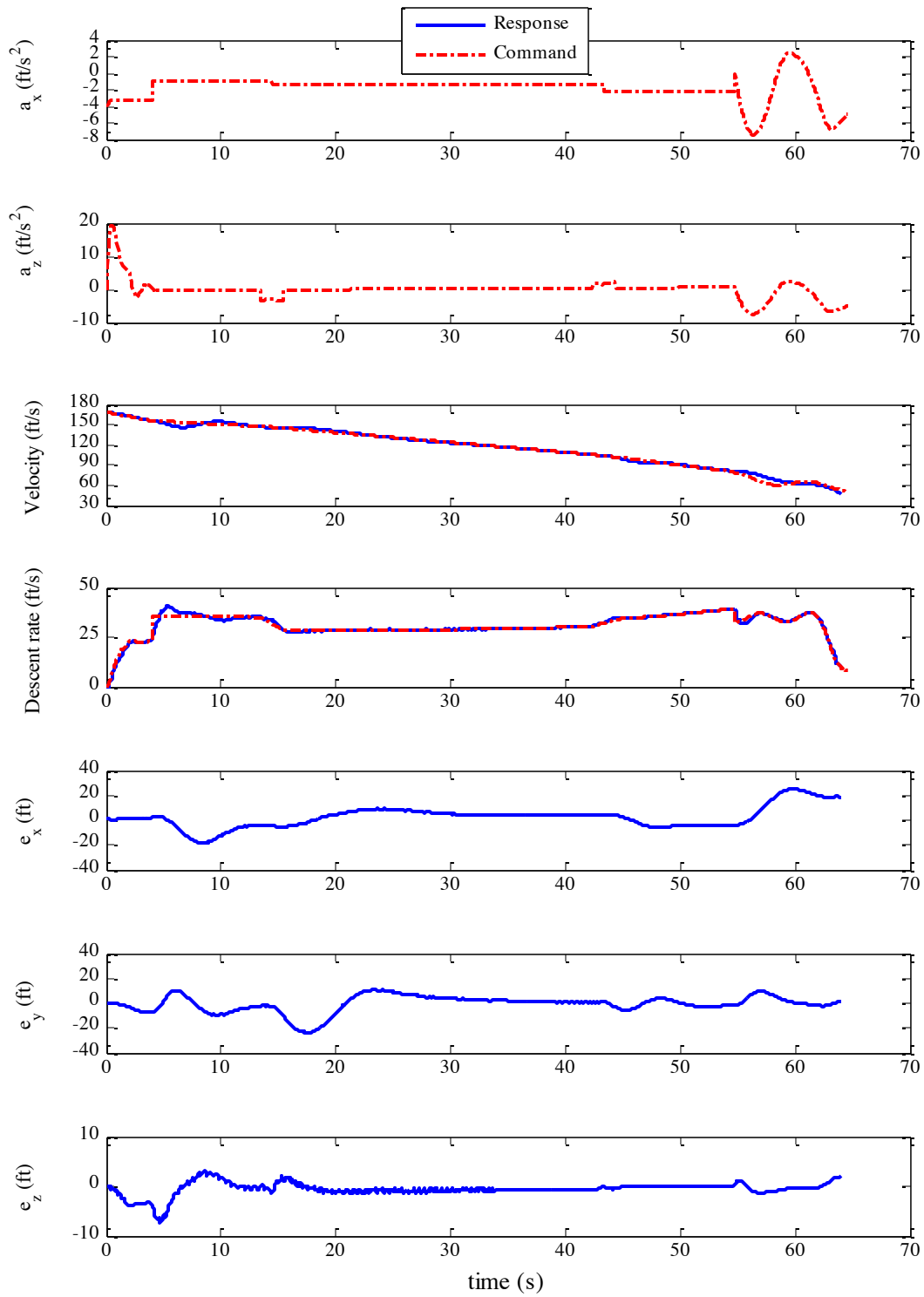**Figure 6. Flight path of autorotation flight to landing for case 1**

Figure 7. State and control histories for case 1

The top two plots show acceleration, the next two plots show airspeed and descent rate, the bottom three plots show trajectory tracking error.
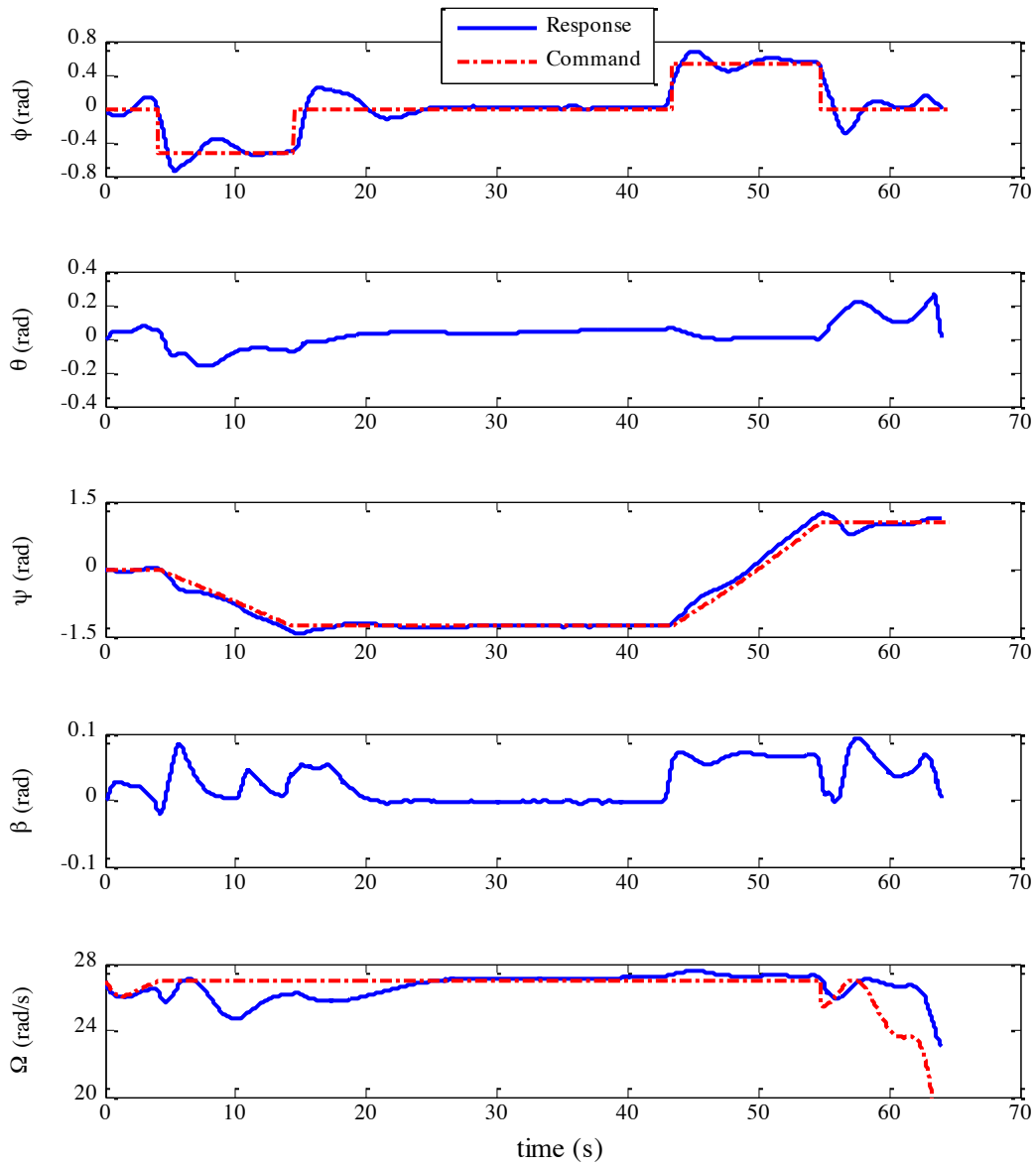
Figure 8. State and control histories for case 1 (continued)
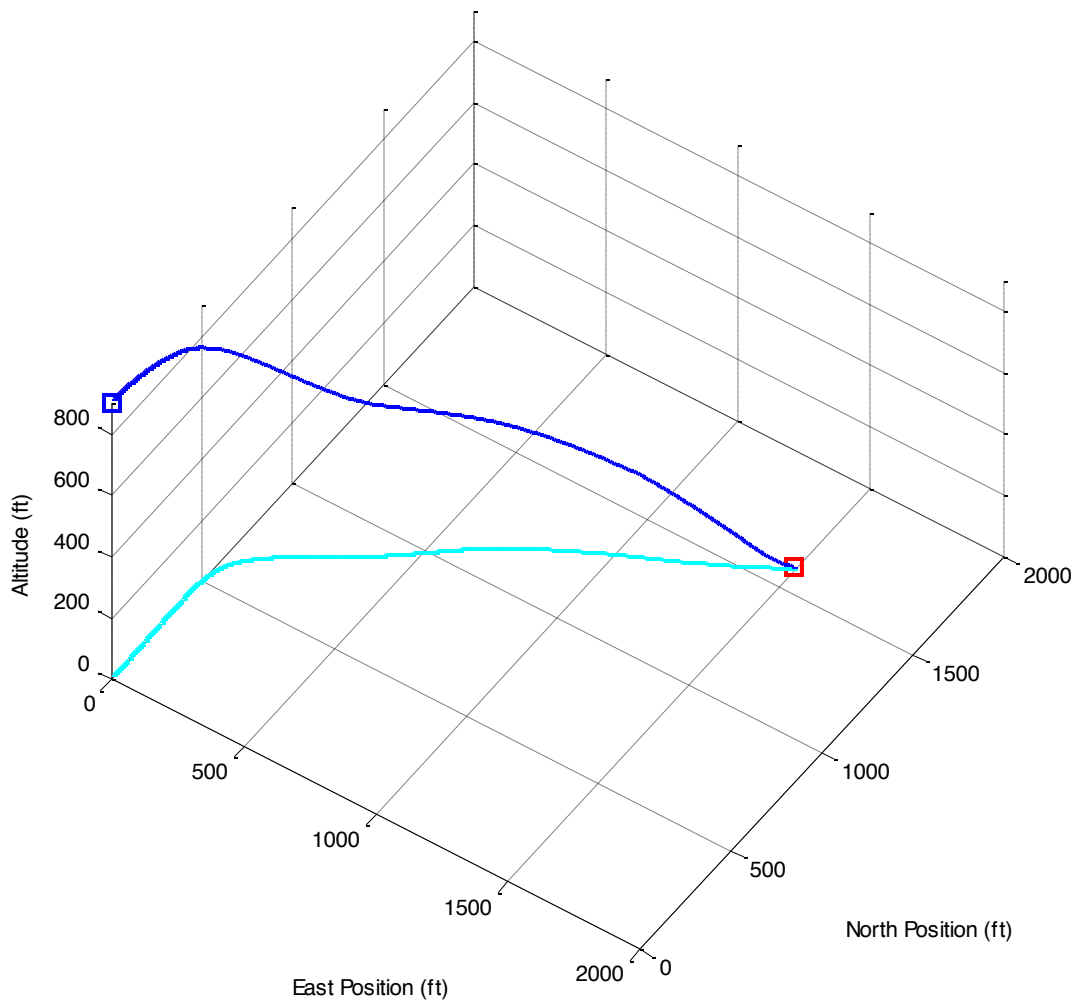*The top three plots show angles (bank, pitch, heading), the fourth plot shows sideslip, the fifth plot shows rotor speed.*

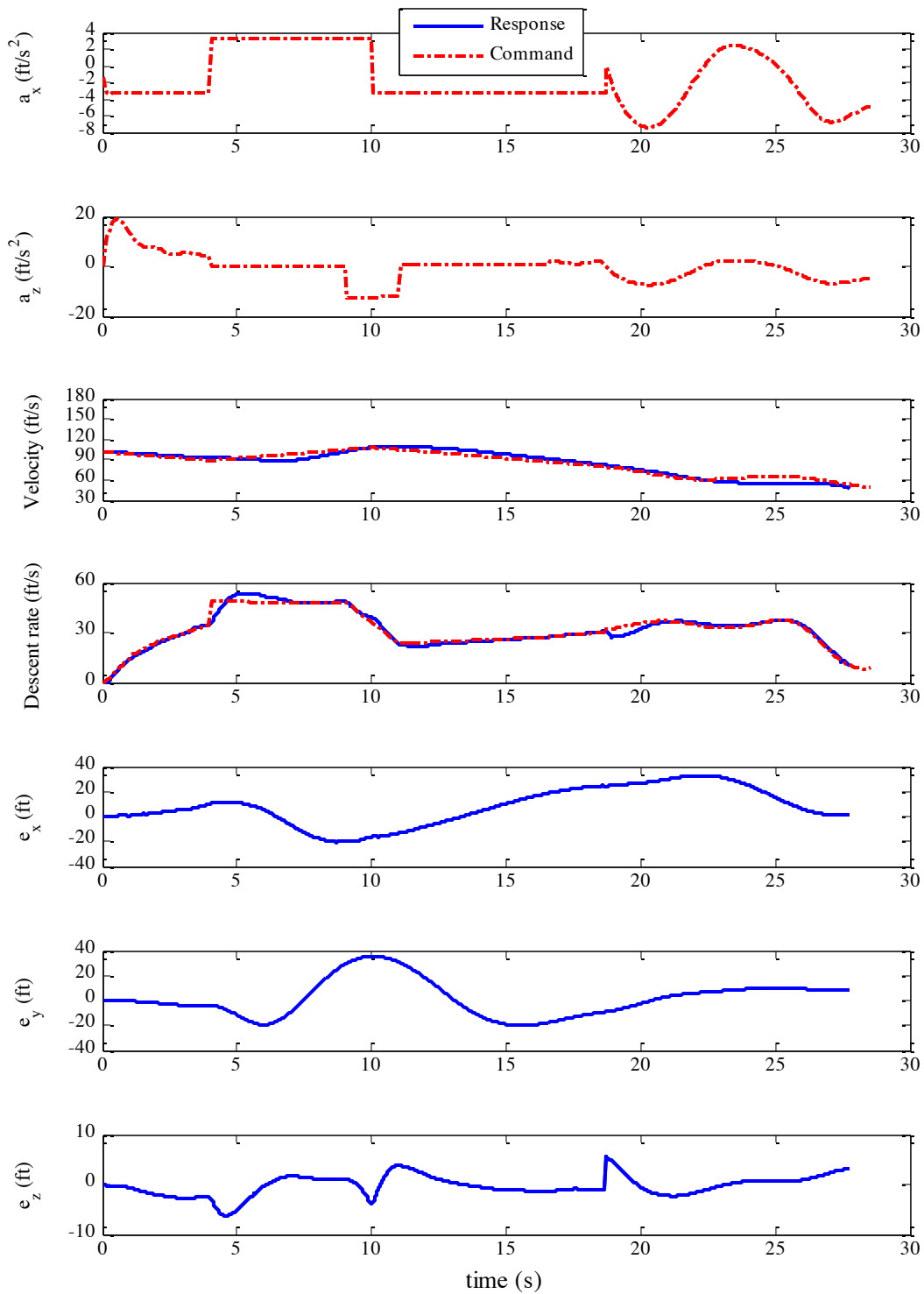**Figure 9. Flight path of autorotation flight to landing for case 2**

Figure 10. State and control histories for case 2

The top two plots show acceleration, the next two plots show airspeed and descent rate, the bottom three plots show trajectory tracking error.
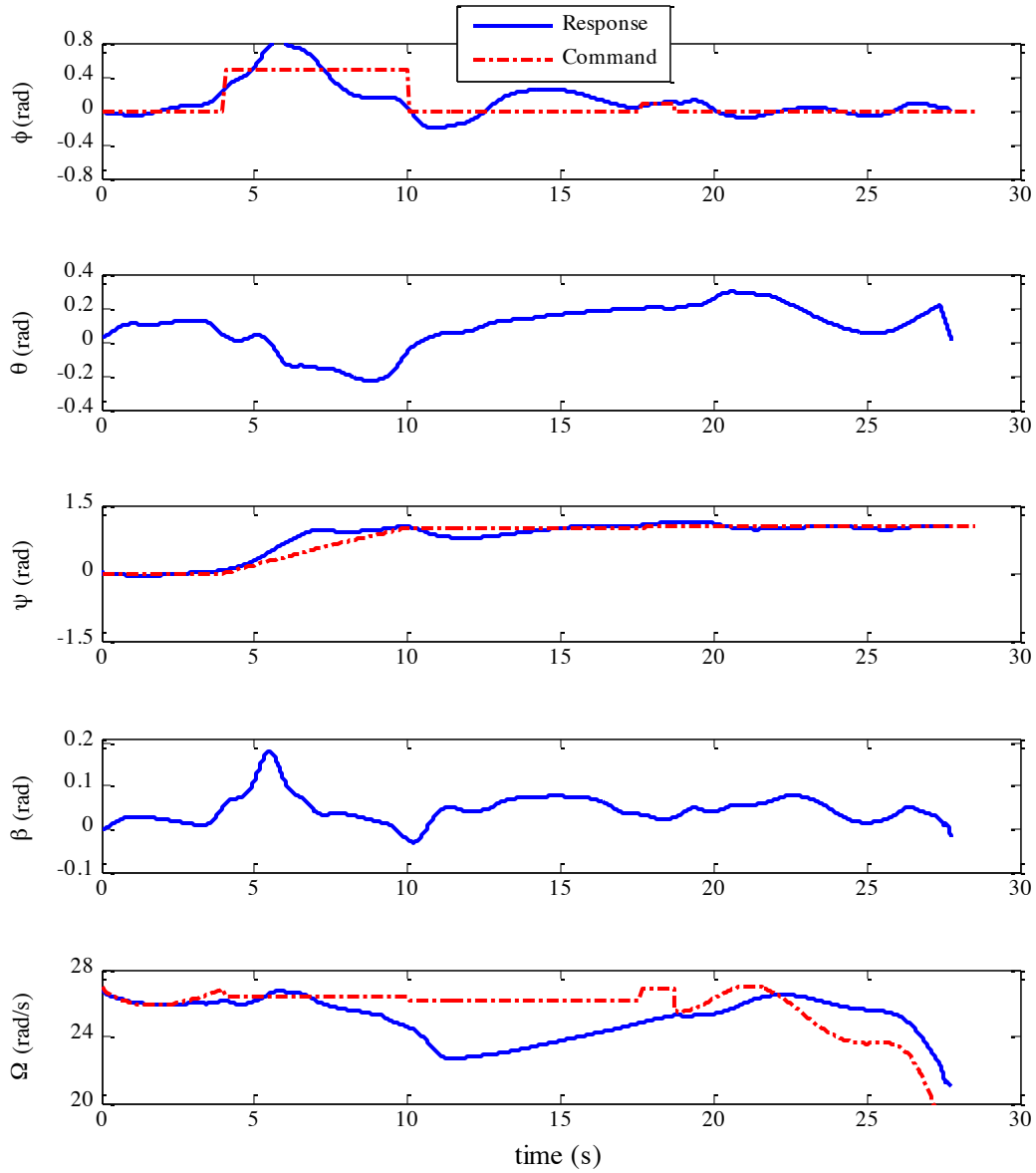
Figure 11. State and control histories for case 2 (continued)
The top three plots show angles (bank, pitch, heading), the fourth plot shows sideslip, the fifth plot shows rotor speed.