AUTONOMOUS EMERGENCY LANDING MANEUVER OF UNMANNED ROTORCRAFT FOR ENGINE INOPERATIVE CONDITIONS

Jun-Young AHN, Department of Aerospace Engineering, Konkuk University, Seoul, Corée du Sud
Chang-Joo KIM, Department of Aerospace Engineering, Konkuk University, Seoul, Corée du Sud

Abstract

Three phases—Entry, Steady-descent, and Flare—of a unique technique for autorotation trajectory development for rotorcraft are presented in this research study. In the Entry and Flare stages, nonlinear optimal control programming is used to build trajectories, which are then stored in a library for selection according to flight conditions. The Steady-descent phase integrates the conditions of the Entry and Flare phases using spline interpolation. After that, a single trajectory is created by integrating the resulting trajectories from each phase. The behavior of the rotorcraft during autorotation is simulated using high-fidelity rotorcraft flight dynamic models, and an incremental backstepping control law is selected to track the trajectory that has been constructed. The suggested method, despite certain drawbacks, performs well in sample situations and offers prospective direction for further study.

1. INTRODUCTION

The motivation of the current research begins with the research about the multiple Rotorcraft Unmanned Aerial Vehicles (RUAVs) operation. During the research, the author found out that excessive research about autonomous flight has been performed from the past to the present. Meanwhile, research about autonomous flight, including emergency landing, especially all engine failure situations, is hard to find based on the author's limited knowledge [1–6]. Despite their successful accomplishment, however, some limitations like the lack of vehicle states used for the optimization [1], and the requirement of the additional computation power [3]. During the previous research of the current research like Ref.[7], some points are found to be important topics to be considered in the emergency landing of autonomous rotorcrafts for future research: 1) complex dynamic behaviors, 2) flyable steady-descent trajectory generation, 3) consideration of the obstacles, and 4) robust control design. Ref.[7] tried to consider those effects, however, the decision-making process for landing site identification is not considered and One Engine Inoperative (OEI) conditions are considered which remain-powers are available. As shown in Ref.[8], in OEI conditions, rotorcraft could climb out to find the landing point via performing 'Category A' takeoff and landing procedures.

In the meanwhile, All Engine Inoperative (AEI) conditions have no remain-power which means no further maneuvers are possible and rotorcraft should land safely within a very short allowable time than OEI conditions. Therefore, the range of flight will be limited from the engine failure points. In AEI conditions, autorotation is the only option for the rotorcraft to safe landing. According to the FAA[9], however, autorotation has many chances of failure during the autorotation maneuver. It implies that the autorotation maneuver has difficulties. Therefore, developing autonomous flight systems for autorotation is necessary. In this research, as an expansion of the previous research(Ref. [8]), All Engine Inoperative (AEI) conditions are considered instead. As mentioned above, guidance laws for landing site selection were also developed and integrated.

In this research, the entire autorotation trajectory is divided into three phases as presented in Ref.[7]: Entry, Steady-descent, Flare. For the entry and flare phase, the Direct Dynamic Simulation Approach (DDSA) [8,10] optimized the trajectory using flight dynamic models proposed by NASA[11]. For steady descent, a spline-interpolation-based approach will be used to generate the trajectory between the entry and flare phases to smoothly connect those two phases. And proposed trajectory will be validated based on the simulation environment. A high-fidelity flight dynamic model, HETLAS [12], with engine rpm dynamics is used to represent the response of the autorotation during the simulation. And Incremental Backstepping Control (IBSC) laws are applied to track the generated trajectory.

2. FRAMEWORK

2.1. Problem Definition

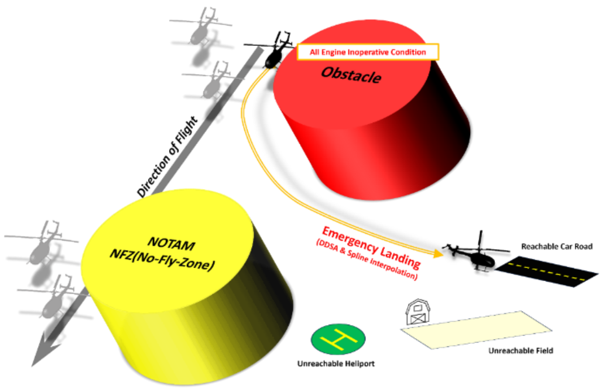

Figure 1: Possible Scenarios for Emergency Landing Scenarios for Unmanned Aerial Vehicles Operation

As mentioned in the introduction, current research is in the context of the multiple RUAVs' mission autonomy. However, in this research, interaction with other RUAVs are not considered here to simplify the research. To perform research, AEI condition occurs to the single RUAV which operates automatically as shown in Figure 1. After the AEI situation occurs, RUAV automatically selects the landing site and establishes autorotation maneuvers toward the selected landing site. Detailed explanations will be presented in the corresponding sections or subsections below.

2.2. 3 Phase Approach for Trajectory Generation

Therefore, the engine-failed RUAV then automatically searches the possible landing site from the current position stabilizing the attitude and recovering RPM. This will be assumed as the 'Entry Phase'. Then, it continually descends to the selected landing site with steady descent. This will be assumed as the 'Steady-Descent Phase'. After the RUAV reaches a certain altitude near the selected landing site, the RUAV will perform landing maneuvers to establish the required attitude for a safe landing. This will be assumed as the 'Flare Phase'. The overall Summary of the 3 phases is shown in Table 1, and visualization of the total trajectory integration is shown in Figure 2.

Table 1 Summary of 3-Phases (I~III) of Autorotation Maneuver

Phase		Goal	Method
I	Site selection	Find a possible landing site	Path planning
	Entry	RPM recover	Trajectory optimization
II	Steady-descent	1) Maintain velocity & RPM 2) Toward the landing site	Path planning
III	Flare	Safe landing	Trajectory optimization

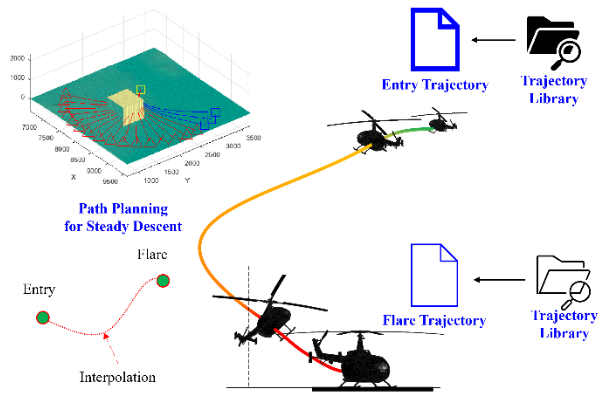

Figure 2 Visualization of the Total Trajectory Integration for 3-Phase Approach.

2.3. Trajectory Generation for Entry and Flare

The trajectory for the entry and flare phases are calculated using the nonlinear optimal control problem. In this research, the DDSA [8,10] method which is mentioned in the introduction is used to solve the formulated optimal control problem. Here, the flight dynamic model proposed by NASA [11] will be imposed as dynamic constraints during the calculation for time duration $t \in [t_0, t_f]$. Here,

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{1}$$

For detailed information about the flight dynamic model used here, see Ref.[11].

2.3.1 Entry Phase Trajectory Design

Figure 3 Illustration of Entry Phase

The purpose of the trajectory optimization for the entry trajectory is to establish a stable attitude and recover the rotor RPM. Therefore, the cost function J is defined as below:

(2) $$J = \frac{1}{2} y_f^2 + \int_{t_0}^{t_f} \frac{1}{2}\left[25\left(\Omega - \Omega_{ref}\right)^2 + \mathbf{x}^T \mathbf{Q}_{entry}\mathbf{x} \right] dt$$

(3) $$\Omega_{ref} = 100\%$$

(4) $$\mathbf{x} = [p, q, r]^T$$

(5) $$\mathbf{Q}_{entry} = 15\mathbf{I}$$

And total flight velocity V_{total} and altitude h_{fail} at the engine failed point are given as the initial conditions.

(6) $$V(t_0) = V$$

(7) $$h(t_0) = h_{fail}$$

And total flight velocity at the engine failed point and desired rate of descent RoD_{steady} are given as the final conditions.

(8) $$V_{total}\left(t_f\right) = V$$

(9) $$\dot{h}\left(t_f\right) = RoD_{steady}$$

2.3.2 Flare Phase Trajectory Design

In the meanwhile, the purpose of the trajectory optimization for the flare trajectory is to establish a stable attitude and decrease the flight speed at the final point.

(10) $$J = y_f^2 + \int_{t_0}^{t_f} \frac{1}{2} \mathbf{x}^T \mathbf{Q}_{flare}\mathbf{x} dt$$

(11) $$\mathbf{x}_1 = [\mathrm{p}, \mathrm{q}, \mathrm{r}]^T$$

(12) $$\mathbf{Q}_{flare} = 5\mathbf{I}$$

Total flight velocity V_{total} and initial flare altitude h_{flare} are given as initial conditions.

(13) $$V(t_0) = V$$

(14) $$\dot{h}(t_0) = RoD_{steady}$$

(15) $$h(t_0) = h_{flare}$$

And final conditions related to the final altitude $h\left(t_f\right)$ and vertical landing speed $\dot{h}_{land}$ are given below:

(16) $$0\,ft \le h\left(t_f\right) \le 3\,ft$$

(17) $$0knot \le \dot{h}\left(t_f\right) \le 2.96knot$$

And final velocity at the landing stage $V_{horizontal}$, and roll angle ϕ_f and pitch angle θ_f are given as inequality conditions to prevent structural damage.

(18) $$\left|V_{horizontal}\right| \le 15knot$$

(19) $$\left|\phi_f\right| \le 30\deg$$

(20) $$\left|\theta_f\right| \le 20\deg$$

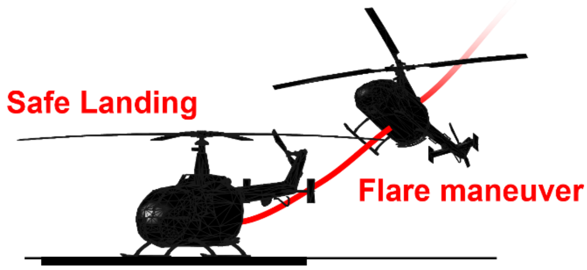

Figure 4 Illustration of Flare Phase

2.4. Trajectory Generation for Steady-Descent

2.4.1 Determination of the Rate of Descent Rate

In the above equations, the determination of the rate of descent speeds are critical issue to make a successful maneuver. To achieve this, autorotation kinematics are developed and used for the autorotation trim analysis. The below equations are developed kinematics for autorotation flight.

(21) $$f_1 = u - V_x$$

(22) $$f_2 = v - V_y$$

(23) $$f_3 = P_{req} - 0$$

After the trim analysis is performed, the rate of descent speed is used for a) the final constraint of the entry phase, and b) the initial constraints of the flare phase.

2.4.2 Trajectory Generation using Interpolation.

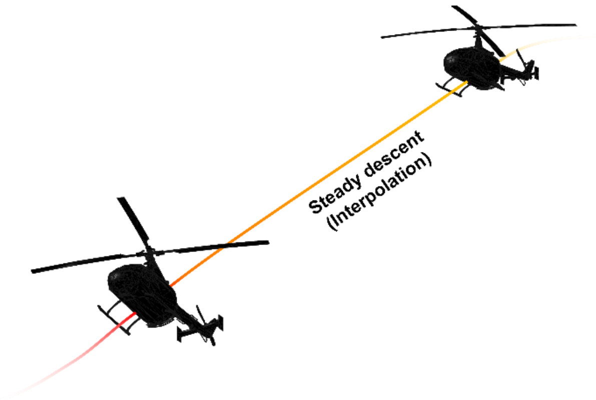

Figure 5 Illustration of the Steady-Descent Phase

For Steady-Descent trajectory generation, the final condition of the entry phase trajectory and the initial condition of the flare phase are used as the initial and final conditions for the spline interpolation to generate the steady descent trajectory. In this research, coordinates of the position, and corresponding roll and pitch angle and their derivatives are used for the conditions of interpolation. At initial point,

(24) $$\mathbf{x}(t_0) = \begin{bmatrix} x_0 & y_0 & z_0 & \phi_0 & \theta_0 \end{bmatrix}^T$$

(25) $$\dot{\mathbf{x}}(t_0) = \begin{bmatrix} V_{x_0} & V_{y_0} & \dot{h}_0 & p_0 & q_0 \end{bmatrix}^T$$

(26) $$\ddot{\mathbf{x}}(t_0) = \begin{bmatrix} 0 & 0 & \ddot{h}_0 & 0 & 0 \end{bmatrix}^T$$

At final point,

(27) $$\mathbf{x}(t_f) = \begin{bmatrix} x_f & y_f & z_f & \phi_f & \theta_f \end{bmatrix}^T$$

(28) $$\dot{\mathbf{x}}(t_f) = \begin{bmatrix} V_{x_f} & V_{y_f} & \dot{h}_f & p_f & q_f \end{bmatrix}^T$$

(29) $$\ddot{\mathbf{x}}(t_f) = \begin{bmatrix} 0 & 0 & \ddot{h}_f & 0 & 0 \end{bmatrix}^T$$

However, as flare phase trajectories were generated by assuming the initial flare point as the initial point, conversion work is required to generate the smooth path. Therefore, the below equations are used to determine the steady descent time and horizontal steady descent distance.

(30) $$\Delta t_{steady} = \frac{\left(h_{entry} - h_{flare}\right)}{\left(\dfrac{\dot{h}_{entry} + \dot{h}_{flare}}{2}\right)}$$

(31) $$d_{steady} = \frac{\left(V_{flare}^2 - V_{entry}^2\right)}{2\left(\dfrac{V_{flare} - V_{entry}}{\Delta t_{steady}}\right)}$$

3.5.3 Landing Site Selection

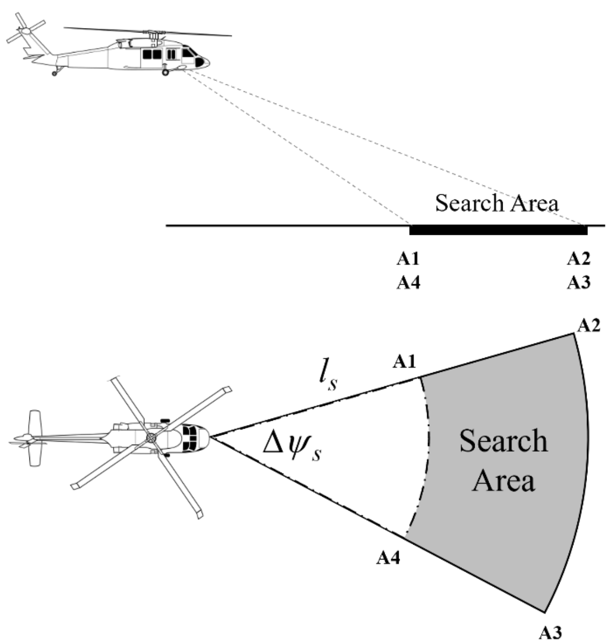

Figure 6 Visualization of the Concept for landing site search

For selecting the landing site, as shown in Figure 6, possible points are surveyed within the predefined search area. This predefined area is defined with search length l_s and search angle $\Delta\psi_s$. Figure 7 shows the detailed concepts of the grid search inside the search area. In Figure 7, two types of grid points are defined: 'Candidate Landing Site' (Diamond shapes) and 'Obstacle Check for Flare' (Square Shapes). Here, 3 strategies are applied. First, the obstacle collision check of the 'Candidate Landing Site' is checked. Second, obstacle collision check of the 'Obstacle Check for Flare'. Finally, the obstacle collision check of the trajectory is checked first. The trajectory used here will be the result of the interpolation described in the previous subsection. As a result, final candidates for the landing sites are selected.

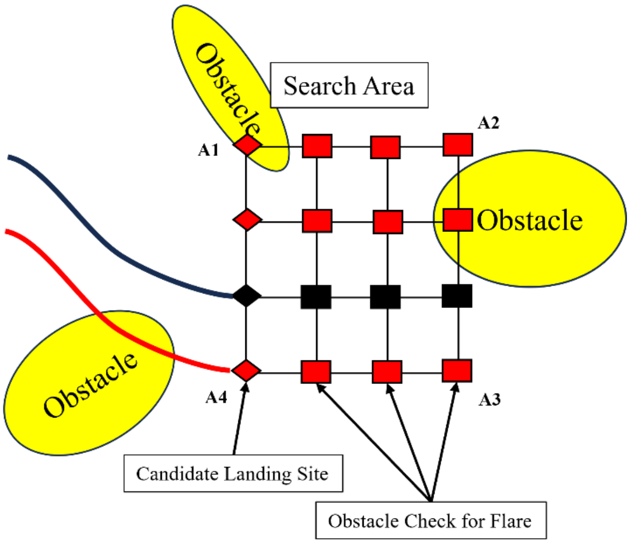

Figure 7 Grid Search Approaches for Landing Site Selection

After the candidates are selected, the load factor of the trajectory will be checked and the trajectory that has the minimum load factor is selected as the final trajectory. To check the load factor, the curvature of the trajectory $\boldsymbol{\kappa}$ is calculated based on the geometry of the trajectory like position $\mathbf{r}$ and their derivatives.

(32) $$\boldsymbol{\kappa}(t) = \frac{d^2\mathbf{r}}{ds^2} = \frac{\ddot{\mathbf{r}}}{V_{Fs}^2}$$

(33) $$\mathbf{r}(t) = \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix}$$

(34) $$\dot{\mathbf{r}}(t) = V_{Fs}\mathbf{u}_t$$

(35) $$\ddot{\mathbf{r}}(t) = \dot{V}_{Fs}\mathbf{u}_t + V_{Fs}\dot{\mathbf{u}}_t = \dot{V}_{Fs}\mathbf{u}_t + \kappa V_{Fs}^2\mathbf{u}_n$$

(36) $$\mathbf{u}_t = \frac{\dot{\mathbf{r}}(t)}{V_{Fs}}$$

(37) $$\mathbf{u}_n = \frac{\boldsymbol{\kappa}}{\|\boldsymbol{\kappa}\|}$$

(38) $$\kappa = \|\boldsymbol{\kappa}\|$$

Therefore, the final load factor will be:

(39) $$\mathbf{n} = \frac{\kappa V_{Fs}^2}{g}\mathbf{u}_n + \mathbf{u}_g$$

In current research, trajectories that have less than 1.5 will be selected as feasible trajectories.

3. SIMULATION RESULTS

3.1. Trajectory Tracking Control Law Design

3.1.1 Development of the Incremental Dynamics

The flight dynamic of the rotorcraft can be represented as the following equation. $\mathbf{f}_b$ and $\mathbf{m}_b$ are forces and moments in a body-fixed frame.

(40) $$\dot{\mathbf{v}}_b = \frac{1}{m}\mathbf{f}_b - \boldsymbol{\omega}\times\mathbf{v}_b$$

(41) $$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}(\mathbf{m}_b - \boldsymbol{\omega}\times\mathbf{J}\boldsymbol{\omega})$$

Their states for linear and angular velocity will be:

(42) $$\mathbf{v}_b = \begin{pmatrix} u & v & w \end{pmatrix}^T$$

(43) $$\boldsymbol{\omega} = \begin{pmatrix} p & q & r \end{pmatrix}^T$$

The above equations are rewritten as:

(44) $$\ddot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}_p\right)$$

(45) $$\mathbf{x} = \left(x, y, z, \phi, \theta, \psi\right)^T$$

(46) $$\mathbf{u}_p = \left(\delta_0, \delta_{1c}, \delta_{1s}, \delta_{TR}\right)^T$$

To design the trajectory tracking control law, incremental dynamics are derived. By assuming that control update times are Δt at specific times t_k, incremental dynamics are shown as:

(47) $$\mathbf{x} = \mathbf{x}_k + \Delta\mathbf{x}$$

(48) $$\dot{\mathbf{x}} = \dot{\mathbf{x}}_k + \Delta\dot{\mathbf{x}}$$

(49) $$\mathbf{u}_p = \mathbf{u}_{pk} + \Delta\mathbf{u}_p$$

(50) $$\ddot{\mathbf{x}}_k = \mathbf{f}_k = \mathbf{f}(\mathbf{x}_k, \dot{\mathbf{x}}_k, \mathbf{u}_{pk})$$

RPM dynamics also included to consider the autorotation. RPM dynamics are presented as:

(51) $$\dot{\Omega} = \frac{1}{I_R\Omega}\left(P_S - P_R\right)$$

(52) $$P_S = P_R + K_{FF}\Delta\delta_{col} - K_p\Delta\Omega - K_I\int\Delta\Omega dt$$

Therefore, the entire dynamics will be presented as below equation. Here, RPM Ω is explicitly represented to clarify it more clearly.

$$\begin{pmatrix} \ddot{\mathbf{x}} \\ \dot{\Omega} \end{pmatrix} = \mathbf{f}_{auto}(\mathbf{x}, \dot{\mathbf{x}}, \Omega, \mathbf{u}_p) = \begin{pmatrix} \mathbf{f}_k(\mathbf{x}, \dot{\mathbf{x}}, \Omega, \mathbf{u}_p) \\ f_\Omega(\mathbf{x}, \dot{\mathbf{x}}, \Omega, \mathbf{u}_p) \end{pmatrix} \tag{53}$$

Approximation of dynamics using the Taylor series expansion, dynamics will be rewritten as below equation. $\mathbf{h}_k$ is higher order term of the Taylor series expansion.

$$\begin{aligned} \begin{pmatrix} \ddot{\mathbf{x}} \\ \dot{\Omega} \end{pmatrix} &\approx \begin{pmatrix} \mathbf{f}_k \\ f_\Omega \end{pmatrix} + \begin{pmatrix} \mathbf{F}_k & \mathbf{F}_\Omega \\ \mathbf{f}_\Omega & f_{\Delta\Omega} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\dot{\mathbf{x}} \\ \Delta\Omega \end{pmatrix} \\ &+ \begin{pmatrix} \mathbf{G}_k \\ \mathbf{g}_k^T \end{pmatrix} \Delta\mathbf{u}_p + \mathbf{h}_k \end{aligned} \tag{54}$$

$$\mathbf{F}_k = \begin{pmatrix} \dfrac{\partial \mathbf{f}_k}{\partial \mathbf{x}} & \dfrac{\partial \mathbf{f}_k}{\partial \dot{\mathbf{x}}} \end{pmatrix} \tag{55}$$

$$\mathbf{F}_\Omega = \frac{\partial \mathbf{f}_k}{\partial \Omega} \tag{56}$$

$$\mathbf{f}_\Omega = \begin{pmatrix} \dfrac{\partial f_\Omega}{\partial \mathbf{x}} & \dfrac{\partial f_\Omega}{\partial \dot{\mathbf{x}}} \end{pmatrix} \tag{57}$$

$$f_{\Delta\Omega} = \frac{\partial f_\Omega}{\partial \Omega} \tag{58}$$

The corresponding control effective matrix will be:

$$\mathbf{G}_k = \frac{\partial \mathbf{f}_k}{\partial \mathbf{u}_p} \tag{59}$$

$$\mathbf{g}_k = \frac{\partial f_\Omega}{\partial \mathbf{u}_p} \tag{60}$$

3.1.2 Incremental Dynamic Inversion

By assuming Δt is small enough, incremental dynamics are estimated as:

$$\begin{pmatrix} \ddot{\mathbf{x}} \\ \dot{\Omega} \end{pmatrix} = \begin{pmatrix} \ddot{\mathbf{x}}_k \\ \dot{\Omega}_k \end{pmatrix} + \begin{pmatrix} \mathbf{G}_k \\ \mathbf{g}_k^T \end{pmatrix} \Delta\mathbf{u}_p \tag{61}$$

To cope with the critical assumption for Δt, angular acceleration prediction techniques proposed in Ref. [13].

To design the control law, the inverse of the total control effective matrix should be required to apply dynamic inversion. However, as the matrix is a non-square matrix, additional methods should be required. Therefore, the slack-variable approach proposed in Ref. [14,15] is applied. Therefore, dynamics will be transformed as below:

$$\begin{pmatrix} \ddot{\mathbf{x}} \\ \dot{\Omega} \end{pmatrix} = \begin{pmatrix} \ddot{\mathbf{x}}_0 \\ \dot{\Omega}_0 \end{pmatrix} + \begin{pmatrix} \mathbf{G}_{auto} \\ \mathbf{g}_{auto}^T \end{pmatrix} \Delta\mathbf{u}_{auto} + \begin{pmatrix} \Delta\mathbf{v} \\ \Delta\nu_\Omega \end{pmatrix} \tag{62}$$

The detailed form of the control effective matrix will be:

$$\mathbf{G}_{auto} = \begin{pmatrix} \mathbf{G}_k & \mathbf{G}_s \end{pmatrix} \tag{63}$$

$$\mathbf{g}_{auto} = \begin{pmatrix} \mathbf{g}_k & \mathbf{g}_s \end{pmatrix} \tag{64}$$

$$\mathbf{G}_s = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \tag{65}$$

$$\mathbf{g}_s = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \tag{66}$$

Control input due to the slack variable will be:

$$\mathbf{u}_s = (u_1, u_2, u_3)^T \tag{67}$$

$$\Delta\mathbf{u}_{auto} = \begin{pmatrix} \Delta\mathbf{u}_p \\ \Delta\mathbf{u}_s \end{pmatrix} \tag{68}$$

Additional terms and ν_Ω will be:

$$\Delta\mathbf{v} = -\mathbf{G}_s \Delta\mathbf{u}_s \tag{69}$$

$$\Delta\nu_\Omega = -\mathbf{g}_s \Delta\mathbf{u}_s \tag{70}$$

3.1.3 Incremental Backstepping Control Design

To design the backstepping control, error dynamics will be defined as below. For trajectory tracking errors,

$$\mathbf{z}_1 = \mathbf{x} - \mathbf{x}_d \tag{71}$$

$$\mathbf{z}_2 = \dot{\mathbf{x}} - \boldsymbol{\alpha} \tag{72}$$

The first derivatives of those error dynamics will be:

$$\dot{\mathbf{z}}_1 = \mathbf{z}_2 + \boldsymbol{\alpha} - \dot{\mathbf{x}}_d \tag{73}$$

$$\dot{\mathbf{z}}_2 = \ddot{\mathbf{x}}_0 + \mathbf{G}_k \Delta\mathbf{u}_p + \Delta\boldsymbol{\xi} - \dot{\boldsymbol{\alpha}} \tag{74}$$

To stabilize the RPM, error dynamics and its derivatives will be:

$$z_\Omega = \Omega - \Omega_d \tag{75}$$

$$\dot{z}_\Omega = \dot{\Omega} - \alpha_\Omega \quad (76)$$

Here, subscription *d* is the desired trajectory or the reference value to be tracked or stabilized by control law. And $\boldsymbol{\alpha}$ and α_Ω indicates the pseudo-control which will be determined to satisfy the Lyapunov stability [16].

To make control laws to track the trajectory and to stabilize the RPM, the Control Lyapunov Function (CLF) will be defined as:

$$\begin{aligned} V &= \frac{1}{2}\mathbf{z}_1^T\mathbf{Q}^{-1}\dot{\mathbf{z}}_1 + \frac{1}{2}\mathbf{z}_2^T\mathbf{z}_2 + \frac{1}{2}k_\Omega z_\Omega^2 \\ &+ \frac{1}{2}\Delta\tilde{\mathbf{v}}^T\mathbf{\Lambda}_v^{-1}\Delta\dot{\tilde{\mathbf{v}}} + \lambda\frac{1}{2_\Omega}\Delta\tilde{v}_\Omega^2 \end{aligned} \quad (77)$$

By taking the derivatives to the CLF,

$$\begin{aligned} \dot{V} &= \mathbf{z}_1^T\mathbf{Q}^{-1}\dot{\mathbf{z}}_1 + \mathbf{z}_2^T\dot{\mathbf{z}}_2 + k_\Omega z_\Omega \dot{z}_\Omega \\ &+ \Delta\tilde{\mathbf{v}}^T\mathbf{\Lambda}_v^{-1}\Delta\dot{\tilde{\mathbf{v}}} + \lambda_\Omega\Delta\tilde{v}_\Omega\Delta\dot{\tilde{v}}_\Omega \end{aligned} \quad (78)$$

By substituting the above from eq.(71) to eq.(76),

$$\begin{aligned} \dot{V} &= \mathbf{z}_1^T\mathbf{Q}^{-1}(\boldsymbol{\alpha} - \dot{\mathbf{x}}_d) + k_\Omega z_\Omega(\dot{\Omega}_0 + \mathbf{g}^T\Delta\mathbf{u}_p - \alpha_\Omega) \\ &+ \mathbf{z}_2^T(\mathbf{Q}^{-1}\mathbf{z}_1 + \ddot{\mathbf{x}}_0 + \mathbf{G}\Delta\mathbf{u}_p - \dot{\boldsymbol{\alpha}}) \\ &+ \Delta\mathbf{v}^T(\mathbf{z}_2 - \mathbf{\Lambda}_v^{-1}\Delta\dot{\tilde{\mathbf{v}}}) + \Delta v_\Omega(k_\Omega z_\Omega - \lambda_\Omega\Delta\dot{\tilde{v}}_\Omega) \end{aligned} \quad (79)$$

By introducing the positive definite matrix $\mathbf{K}_1 > 0$ and $\mathbf{K}_2 > 0$,

$$-\mathbf{K}_1\mathbf{z}_1 = \mathbf{Q}^{-1}(\boldsymbol{\alpha} - \dot{\mathbf{x}}_d) \quad (80)$$

$$-\mathbf{K}_2\mathbf{z}_2 = \mathbf{Q}^{-1}\mathbf{z}_1 + \ddot{\mathbf{x}}_0 + \mathbf{G}_k\Delta\mathbf{u}_p - \dot{\boldsymbol{\alpha}} \quad (81)$$

$$-z_\Omega = (\dot{\Omega}_0 + \mathbf{g}_k^T\Delta\mathbf{u}_p - \alpha_\Omega) \quad (82)$$

$$0 = \mathbf{z}_2 - \mathbf{\Lambda}_v^{-1}\Delta\dot{\tilde{\mathbf{v}}} \quad (83)$$

$$0 = k_\Omega z_\Omega - \lambda_\Omega\Delta\dot{\tilde{v}}_\Omega \quad (84)$$

From eq.(80), eq.(81), and eq.(82), shapes of the pseudo-controls are determined by making arrangements. Therefore, Lyapunov stability will be satisfied as below:

$$\dot{V} = -\mathbf{z}_1^T\mathbf{K}_1\mathbf{z}_1 - \mathbf{z}_2^T\mathbf{K}_2\mathbf{z}_2 - k_\Omega z_\Omega^2 < 0 \quad (85)$$

Therefore, the designed control input will be:

$$\Delta\mathbf{u} = -\mathbf{G}^{-1}(\mathbf{K}_2\mathbf{z}_2 + \mathbf{Q}^{-1}\mathbf{z}_1^T + \ddot{\mathbf{x}}_0 - \dot{\boldsymbol{\alpha}}) \quad (86)$$

3.2. Example Scenarios

Table 2 Coordinates of Engine Failure Point

X (m)	Y (m)	Z (m)
8000	2200	914.4

Table 3 Inserted No Fly Zone (NFZ) for Emergency Landing Scenario

NFZ	X (m)	Y (m)	Height (m)	Width (m)	Length (m)
1	4250	2250	260	500	500
2	5150	5200	240	300	400
3	6250	1150	180	500	700
4	8250	1750	1500	500	500
5	7250	4700	180	500	400

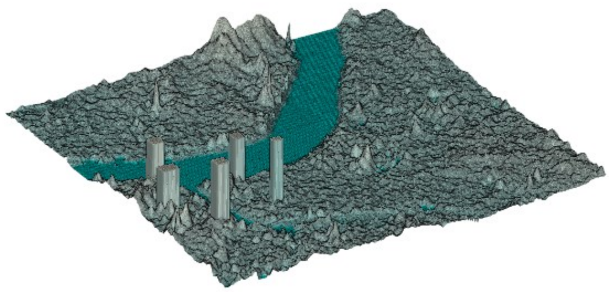

Figure 8 Visualized 3D Terrain used as Example Scenario

Example scenarios defined here are described in this subsection. Engine failure point information is presented in Table 2 and inserted No Fly Zone information is presented in Table 3. Figure 8 shows the actual terrain and inserted NFZ which will be used in the simulation. Figure 9 shows the example rotorcraft, Bo-105, which is implemented in the HETLAS.

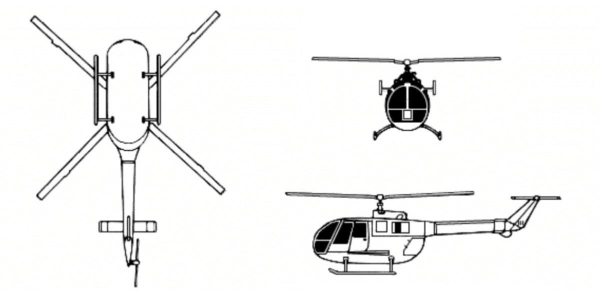

Figure 9 Bo-105 Rotorcraft as an Example Rotorcraft for the Validation of the Proposed Framework

3.3. Analysis Results

In this subsection, some analysis results are represented.

3.3.1 Landing Site Selection Results

Figure 10 shows the load factor of the generated candidate trajectories. The 'Best Path' presented in Figure 10 means the selected final path which satisfies the 'Geometric Feasible' which means that the trajectory is the obstacle-free trajectory and 'Dynamic Feasible' which means that the trajectory satisfies the maximum load-factor limits. Detailed information about those limits is shown in Figure 11. Figure 12 shows the selected landing site and integrated trajectory. It was found that the finally generated trajectory successfully avoided NFZ (or obstacle) and reached the landing area.

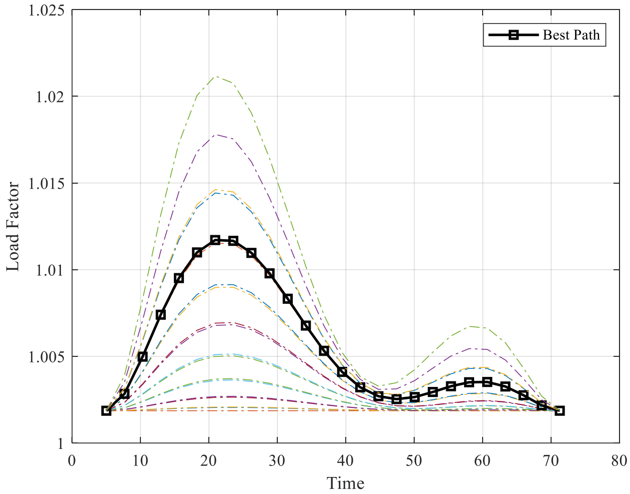

Figure 10 Calculation of the Load Factor of the Generated Trajectory

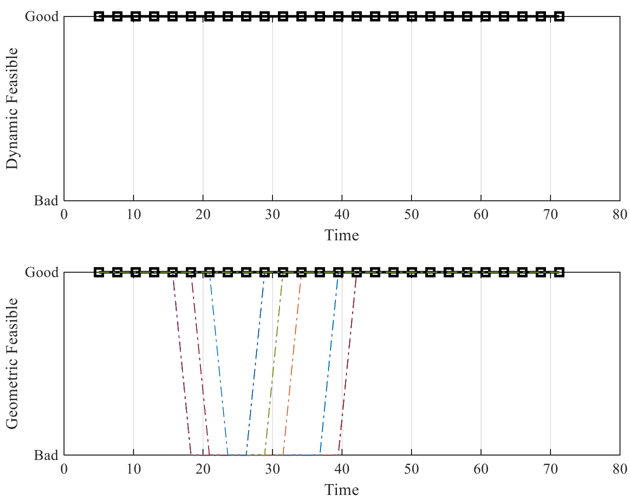

Figure 11 Trajectory Check for Dynamic Feasible and Geometric Feasible.

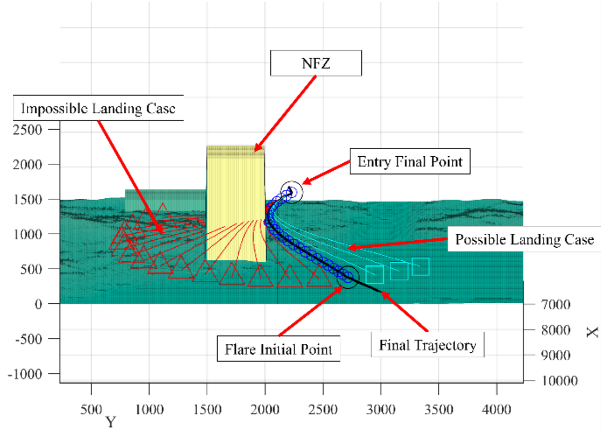

Figure 12 Description of the Integrated Path with 3D Terrain

3.3.2 Trajectory Tracking Results

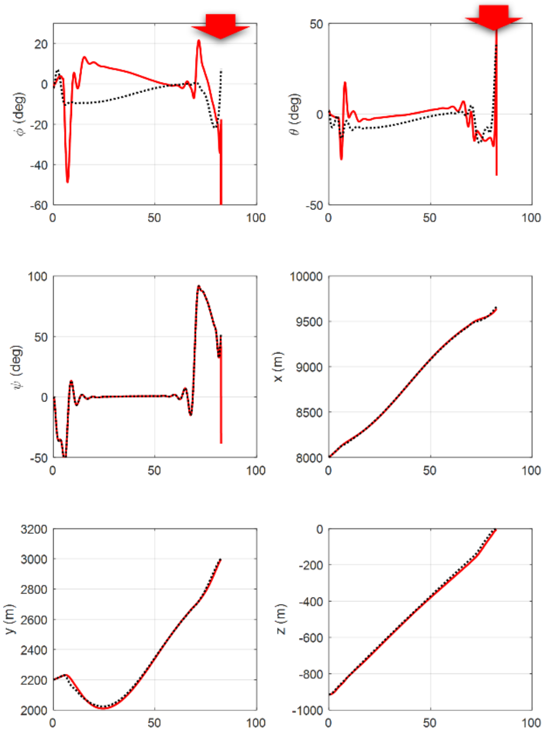

Figure 13 Trajectory Tracking Results using IBSC

Trajectory tracking results using IBSC are presented in Figure 13. In the results, except for the final few seconds, most of them show some good tracking results. One of the reasons for those responses might be explained by Figure 14 which shows the available and required power of the rotorcraft and the corresponding RPM of the rotor. During the entire maneuvers, except for the final few seconds, IBSC maintain the RPM within 90% to 110% successfully. In the final few seconds, however, control laws somehow failed to maintain the RPM within a certain level and it decreased dramatically. Due to that reason, rotorcraft cannot attain enough thrust from the rotor and fail to maintain the attitude required for safe landing. Therefore, further research will be performed to improve the response at the final stage.

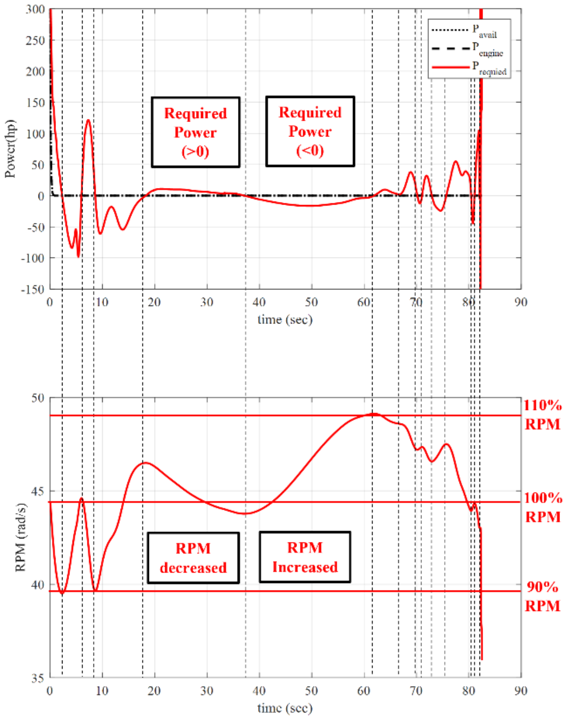

Figure 14 Power and RPM as the results of the Trajectory Tracking Results using IBSC

4. CONCLUSIONS

The most important results of the current research are summarized below:

1. In this research, integration of the guidance and control for autonomous emergency landing, especially for the AEI condition was proposed and validated.
2. The proposed method shows successful results for most of the autorotation maneuvers including the autonomous landing site selection and successful trajectory tracking results with RPM regulations.
3. However, current research also has some limitations like the failure to maintain a stable attitude in the final few seconds. Therefore, further research should be required to improve those problems.
4. Other factors like external disturbance are also one of the topics to be considered in future research. Therefore, some methodologies like adaptive control and disturbance modeling might be considered in future research.
5. As a result, even though some limitations exist, the proposed method shows some reasonable results. Therefore, the author expects that current research might give some insight.

Author contact:

Jun-Young AHN denix717@konkuk.ac.kr

Chang-Joo KIM cjkim@konkuk.ac.kr

5. ACKNOWLEDGMENTS

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2020R1A6A1A03046811). This work was supported by the NRF grant funded by the Korean government (MSIT) (NRF-2020R1A2C2011955)

6. REFERENCES

[1] S. Tierney, J.W. Langelaan, Autorotation path planning using backwards reachable set and optimal control, Annual Forum Proceedings - AHS International. 3 (2010) 1895–1907.

[2] T. Yomchinda, J. Horn, J. Langelaan, Flight Path Planning for Descent-phase Helicopter Autorotation, in: AIAA Guidance, Navigation, and Control Conference, American Institute of Aeronautics and Astronautics, Reston, Virigina, 2011. https://doi.org/10.2514/6.2011-6601.

[3] T. Yomchinda, J.F. Horn, J.W. Langelaan, Autonomous control and path planning for autorotation of unmanned helicopters, Annual Forum Proceedings - AHS International. 4 (2012) 2941–2955.

[4] J. Holsten, S. Loechelt, W. Alles, Autonomous autorotation flights of helicopter uavs to known landing sites, Annual Forum Proceedings - AHS International. 1 (2010) 387–408.

[5] P. Abbeel, A. Coates, T. Hunter, A.Y. Ng, Autonomous Autorotation of an RC Helicopter, Springer Tracts in Advanced Robotics. 54 (2009) 385–394. https://doi.org/10.1007/978-3-642-00196-3_45.

[6] D.J. Lee, H. Bang, Autonomous autorotation of an unmanned helicopter using a reinforcement learning algorithm, Journal of Aerospace Information Systems. 10 (2013) 98–104. https://doi.org/10.2514/1.48253.

[7] I.-R. Lee, Y.-S. Choi, M.G. Lim, J.W. Woo, C.-J. Kim, Guidance and control for autonomous emergency landing of the rotorcraft using the incremental backstepping controller in 3-dimensional terrain environments, Aerospace Science and Technology. 132 (2023) 108051. https://doi.org/10.1016/j.ast.2022.108051.

[8] Y.H. Nam, C.-J. Kim, S.H. Lee, S.W. Hur, Y.Y. Kwak, Direct Dynamic-Simulation Approach to Trajectory Optimization for Rotorcraft Category-A Maneuver Procedures, International Journal of Aeronautical and Space Sciences. 22 (2021) 648–662. https://doi.org/10.1007/s42405-020-00322-2.

[9] FAA, Helicopter Flying Handbook (FAA-H-8083-21B), Federal Aviation Administration. (2019) 208.

[10] S.W. HUR, S.H. LEE, Y.H. NAM, C.-J. KIM, Direct dynamic-simulation approach to trajectory optimization, Chinese Journal of Aeronautics. 34 (2021) 6–19. https://doi.org/10.1016/j.cja.2021.01.019.

[11] P.D. Talbot, B.E. Tinling, W. Decker, R.T.N. Chen, A mathematical model of a single main rotor helicopter for piloted simulation, NASA Technical Memorandum. 84281 (1982) 46.

[12] J. An, Y.-S. Choi, I.-R. Lee, M. Lim, C.-J. Kim, Performance Analysis of a Conceptual Urban Air Mobility Configuration Using High-Fidelity Rotorcraft Flight Dynamic Model, International Journal of Aeronautical and Space Sciences. (2023). https://doi.org/10.1007/s42405-023-00610-7.

[13] J. An, C.-J. Kim, Prediction of Angular Acceleration for Practical Realization of Incremental Nonlinear Trajectory-tracking Control for Aircrafts, International Journal of Control, Automation and Systems. 20 (2022) 1250–1265. https://doi.org/10.1007/s12555-021-0232-8.

[14] D. Lee, H. Lim, H. Kim, Y. Kim, K.-J. Seong, Adaptive Image-Based Visual Servoing for an Underactuated Quadrotor System, Journal of Guidance, Control, and Dynamics. 35 (2012) 1335–1353. https://doi.org/10.2514/1.52169.

[15] M. Kim, Y. Hong, S. Lee, Y. Kim, Slack Variables Generation via QR Decomposition for Adaptive Nonlinear Control of Affine Underactuated Systems, IFAC-PapersOnLine. 49 (2016) 188–193. https://doi.org/10.1016/j.ifacol.2016.09.033.

[16] Y.N. Raffoul, General theorems for stability and boundedness for nonlinear functional discrete systems, Journal of Mathematical Analysis and Applications. 279 (2003) 639–650. https://doi.org/10.1016/S0022-247X(03)00051-9.