



Emergency Landing Trajectory Optimization for a Fixed-Wing UAV under Engine Failure

Xiang Fang* and Florian Holzapfel†
Technical University of Munich, 85748 Garching, Germany

Neng Wan‡, Hamidreza Jafarnejadsani§, Donglei Sun¶, and Naira Hovakimyan||
University of Illinois at Urbana-Champaign, Urbana, IL 61801

With the growing popularity of autonomous unmanned aerial vehicles (UAVs), the improvement of safety for UAV operations has become increasingly important. In this paper, a landing trajectory optimization scheme is proposed to generate reference landing trajectories for a fixed-wing UAV with accidental engine failure. For a specific landing objective, two types of landing trajectory optimization algorithms are investigated: i) trajectory optimization algorithm with nonlinear UAV dynamics, and ii) trajectory optimization algorithm with linearized UAV dynamics. An initialization procedure that generates an initial guess is introduced to accelerate the convergence of the optimization algorithms. The effectiveness of the proposed scheme is verified in a high-fidelity UAV simulation environment, where the optimized landing trajectories are tracked by a UAV equipped with an $\mathcal{L}_1$ adaptive altitude controller in both the offline and online modes.

I. Nomenclature

$\mathbb{N}$	=	$\{0, 1, 2, \dots\}$, set of natural numbers with 0
$\mathbb{N}^+$	=	$\{1, 2, \dots\}$, set of natural numbers without 0
V	=	speed
X	=	horizontal flight distance
h	=	UAV height above ground
H	=	UAV altitude above sea level
α	=	angle of attack
β	=	angle of sideslip
γ	=	flight-path inclination angle
ϕ, θ, ψ	=	attitude Euler angles for roll, pitch, and yaw
p, q, r	=	roll, pitch, and yaw rates
$\delta_e, \delta_r, \delta_a$	=	elevator, rudder, and ailerons deflections
δ_t	=	thrust level

II. Introduction

ENGINE failure is one of the most fatal accidents for aircraft systems [1]. US Airways Flight 1549's landing over Hudson River in 2009 was because of an engine-out failure [2]. Thanks to the professional response of Captain Chesley Sullenberger, the Airbus A320 carrying 155 passengers and crew members landed safely on the Hudson River after striking a flock of birds and losing thrust in both engines. For a civil airplane, the safety of passengers is always the priority at all costs, and engine-out-failures are typically not common. However, for a commercial fixed-wing UAV,

*Doctoral Student, Institute of Flight System Dynamics, Student Member AIAA; xiang.fang@tum.de.

†Professor, Institute of Flight System Dynamics, Associate Fellow AIAA; florian.holzapfel@tum.de.

‡Doctoral Student, Dept. of Mechanical Science and Engineering; nengwan2@illinois.edu.

§Doctoral Student, Dept. of Mechanical Science and Engineering, Student Member AIAA; jafarne2@illinois.edu.

¶Doctoral Student, Dept. of Aerospace Engineering, Student Member AIAA; dsun13@illinois.edu.

||Professor, Dept. of Mechanical Science and Engineering, Fellow AIAA; nhovakim@illinois.edu.

which is preferably designed as a low-cost aircraft, an engine-out accident is more probable. In this paper, we propose to equip autonomous UAVs with a safety feature for emergency landing, which otherwise requires an experienced human operator.

Autonomous UAVs are playing an important role in the modern world. For example, UAVs have already been employed to deliver commercial packages [3, 4] and medications [5, 6]. Consequently, the safety concerns of using UAVs are increasingly drawing the attention of researchers and companies. When a UAV flies over a populous urban area, failures in the UAV systems can even endanger the safety of humans and facilities on the ground. Engine failure is undoubtedly one of the most hazardous UAV failures [1, 7, 8]. An engine-out aircraft can be landed via gliding [9–12]. An adaptive trajectory generation scheme with a certain presumed best glide ratio and bank angle for turns was proposed in [11]. Trajectory planning for an engine-out aircraft towards a specified airstrip was investigated in [12]. In [10], the reachable set for auto-landing was calculated by using the numerical method in optimal control theory. The maximum-range trajectory problem was solved in [9] by implementing the singular perturbation theory. However, when generating a reference gliding trajectory for a UAV that usually flies over an urban area in low altitude, more requirements should be considered in addition to avoiding densely populated areas or providing enough time for evacuation on the ground [9]. Since most of the UAV autopilot software is executed on some low-cost embedded platform, the desired UAV landing trajectory optimization scheme should generate a trajectory quickly and without costing substantial computation resources, such that the engine-out UAV can have enough time and resources to follow the reference trajectory. Meanwhile, since the urban surroundings along the flight path can be versatile, the optimization scheme should be able to generate the trajectories with different landing requirements.

Direct and indirect methods are two categories of approaches in solving the trajectory optimization problem [13]. The fundamental basis of indirect methods is given by the analytical solution of the Hamilton–Jacobi–Bellman equation and by Pontryagin’s maximum principle. Nevertheless, for most practical and complicated systems, these analytical solutions can hardly be derived [14]. In contrast, direct methods are more applicable to solving real-world problems. Direct methods discretize the optimization problem into a nonlinear numerical optimization problem and solve it. Some well-known algorithms in the direct methods are multiple shooting [13, 15], full collocation, and pseudo-spectral method [16]. In general, large amounts of computations are demanded by these algorithms to recursively search for the optimal solution in an offline mode. Nonetheless, a wise choice of initial value can accelerate the optimization process remarkably.

Subject to the requirements mentioned above, an emergency landing trajectory optimization scheme for engine-out UAV is investigated. Two types of landing trajectory optimization algorithms are proposed: i) optimization with nonlinear UAV dynamics, and ii) optimization with linearized UAV dynamics. By defining different cost functions, we can generate landing trajectories with different properties, such as maximal horizontal distance or maximal flight time. For a specific objective function, the optimization algorithm with nonlinear UAV dynamics can generally obtain a trajectory with a lower cost value, while the algorithm with linearized UAV dynamics can find comparable trajectory with an average 40% less running time. Initialization procedure that generates an initial guess is introduced to accelerate the convergence of the optimization algorithms. With this well-selected initial guess, the running time of both algorithms is limited within 1.2 s in the emergency landing examples, presented at the end of this paper. Incorporated with a sampled data $\mathcal{L}_1$ adaptive altitude tracking controller, the optimized emergency landing trajectories are verified in a high-fidelity UAV simulation environment.

This paper is organized as follows: Section III introduces the preliminary knowledge and the problem formulation; Section IV presents the main results on UAV emergency landing trajectory optimization. High-fidelity UAV emergency landing simulation examples are provided in Section V. Finally, Section VI concludes the paper.

III. Preliminaries and Problem Formulation

Two basic assumptions of this paper are stated first.

Assumption 1 *The longitudinal and lateral dynamics and control of fixed-wing UAV are weakly coupled.*

Remark 1 *This assumption is widely adopted in the existing literature on fixed-wing aircraft and UAV [17, 18]. With this assumption, we only consider the UAV longitudinal dynamics when optimizing the landing trajectory, while the lateral dynamics are assumed to be free of constraints. In simulations, the optimized results are applied to the longitudinal dynamics only, while the lateral dynamics are assumed to be stabilized by some preexisting controllers or commanded by some independent reference signals.*

Assumption 2 *The flight states of UAV are available to the trajectory generation software, when the engine fails.*

Remark 2 *Since the objective of the emergency landing trajectory optimization scheme is to quickly generate an optimal landing trajectory for an engine-out UAV to follow, it is crucial for the trajectory generation software to access the most updated UAV flight states as the initial condition in optimization. This information can be retrieved from the measurement units on UAV operated with a fault-detection system.*

A. UAV Motion Models

Nonlinear and linearized UAV dynamics are considered to optimize the emergency landing trajectories, and the effectiveness of the optimized trajectories is verified in a high-fidelity UAV simulation environment. All the model parameters used in this paper are found in [19], where an *Ultra StickTM 25e* high-fidelity model was developed from wind tunnel experiments. Since the nonlinear and linearized UAV models have been extensively discussed in literature [18–20], we only represent them in general terms.

1. Nonlinear Model

A nonlinear six degrees of freedom UAV model can generally be expressed as

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

where $\mathbf{x}$ and $\mathbf{u}$ are the states and control vectors, respectively. Without considering disturbances, the nonlinear model Eq.(1) is precise enough to illustrate the dynamics of fixed-wing UAV flight, and hence it provides an accurate model for the trajectory optimization algorithms to find a landing trajectory that satisfies the flight constraints. More detailed descriptions about nonlinear UAV models are provided in [18, 20]. The parameters and software package of the nonlinear UAV model can be found in [19].

2. Trim Point and Linearized Model

Although the nonlinear models can describe the UAV dynamics accurately, a precise nonlinear UAV model demands considerable computation resources in each update. The extensive computation contradicts our requirements for fast trajectory generation and reduced computational burden. One approach to mitigate this problem and accelerate the optimization process is to replace the nonlinear UAV model with a linearized UAV model, which is less complex and less accurate. The linearized UAV dynamics are obtained around trim points. A trim point is a steady-state flight operation point of UAV. For an engine-out UAV, the trim point for a steady-state glide without turn or sideslip can be retrieved by solving the following conditions

$$[\dot{V}_s, \dot{\alpha}_s, \dot{\beta}_s, \dot{\phi}_s, \dot{\theta}_s, \dot{\psi}_s, \dot{p}_s, \dot{q}_s, \dot{r}_s]^\top = \mathbf{f}_{\text{trim}}(\mathbf{p}_s, \mathbf{z}_s) = \mathbf{0}, \quad (2)$$

where $\mathbf{f}_{\text{trim}}$ is a subset of $\mathbf{f}$ in Eq.(1); $\mathbf{p}_s = [V_s, \beta_s, \delta_{t,s}, H]^\top$ is the parameters vector with $(\beta_s, \delta_{t,s}) = (0, 0)$, and H denotes the UAV altitude above sea level; the vector $\mathbf{z}_s = [\alpha_s, \phi_s, \theta_s, p_s, q_s, r_s, \delta_{e,s}, \delta_{r,s}, \delta_{a,s}]^\top$ can be obtained by solving Eq.(2) at the trim point, and the subscript s denotes the value retrieved from the trim. For convenience, we define the function $\mathbf{x}_s = \text{trim}(V_s, H)$, in which $\mathbf{x}_s = [V_s, \gamma_s, \alpha_s, q_s, \delta_{e,s}]$ with $\gamma_s = \theta_s - \alpha_s$. More comprehensive introduction on UAV trim algorithms can be found in [20].

Once the parameters are determined, we can linearize the nonlinear dynamics in Eq.(1) around the trim point $\mathbf{x}_s$. The linearized model is given as follows

$$\begin{aligned} \dot{\mathbf{x}}_{\text{lin}} &= \mathbf{A}(\mathbf{x}_{\text{lin}} - \mathbf{x}_s) + \mathbf{B}\dot{\delta}_e, \\ \dot{X} &= V \cos \gamma, \\ \dot{h} &= V \sin \gamma, \end{aligned} \quad (3)$$

where $\mathbf{x}_{\text{lin}} = [V, \gamma, \alpha, q, \delta_e]^\top$, $\mathbf{A} := \partial \mathbf{f}(\mathbf{x}_s) / \partial \mathbf{x}_{\text{lin}}$, and $\mathbf{B} := [0, 0, 0, 0, 1]^\top$. The kinetic update laws of X and h are kept in nonlinear form. More detailed investigation on longitudinal motion analysis and linearization is available in [18].

3. High-Fidelity Simulation Environment

After generating the reference landing trajectories using the nonlinear and the linearized models, we verify the effectiveness of the optimized trajectories in a high-fidelity UAV simulation platform developed by the UAV Lab at the University of Minnesota*. This high-fidelity platform incorporates a nonlinear UAV model, actuator models, measurement noise, and models of environmental factors. Position limits, rate limits, and bandwidths of actuators are considered in the actuator model. The measurement noise is modeled as a Gaussian process. The environmental model considers steady winds, wind gusts, a precise gravity field, a magnetic field, and a precise atmospheric model. More detailed information and applications on this high-fidelity simulation platform for *Ultra StickTM 25e* can be found in [19].

B. Landing Trajectory Optimization

The generation of UAV emergency landing trajectory is formulated as an optimization problem. Safety concerns and landing requirements are formulated in the cost function. For example, when we desire the longest time to evacuate an area before the engine-out UAV crashes, we can choose the cost function such that the terminal time t_f is maximized, and the cost function can be designed to maximize the horizontal distance $X(t_f)$, when we want to deviate an engine-out UAV from a densely populated area. In the following sections, both scenarios will be considered. The UAV models, actuator limitations, and flight envelope are formulated as the constraints of the optimization problem. Mathematical formulae of this optimization problem will be given in Section IV.

To solve the optimization problem, this paper utilizes the optimal control tool, FALCON.m[†], developed by the Institute of Flight System Dynamics at the Technical University of Munich. This optimization solver is based on the direct methods, e.g., the trapezoidal collocation scheme, and has been applied to various trajectory optimization tasks [21–24]. FALCON.m can automatically convert the general optimal control problem into a standard nonlinear programming (NLP) problem, and then solve the NLP problem with mature third-party NLP solvers embedded in FALCON.m, such as IPOPT [25] and SNOPT [26]. In this paper, the NLP problem is solved numerically by IPOPT, which is based on a primal-dual interior-point algorithm with a filter line-search method.

C. Altitude Tracking Controller

To verify the effectiveness of the reference trajectories in a high-fidelity UAV simulation environment, an altitude tracking control framework is developed to follow the landing trajectories generated by the optimization problem. The control framework consists of an altitude tracker that generates a reference command for pitch angle [19], and a pitch angle tracker using $\mathcal{L}_1$ adaptive control structure. $\mathcal{L}_1$ adaptive controllers have been successfully implemented and verified on numerous aerial vehicles [27–30], including F-16, Learjet, and AirSTAR. For conciseness, the formulation of the control scheme is provided in the Appendix. One can also implement other altitude tracking controllers here to follow the reference landing trajectories.

IV. UAV Emergency Landing Trajectory Optimization Algorithms

The mathematical formulation of UAV emergency landing optimization problem is given in this section. An initialization technique is introduced to accelerate the convergence of the landing trajectory optimization. At the end of this section, the optimization algorithms using the nonlinear and linearized UAV models are provided.

The trajectory optimization problem can be formulated as an optimal control problem with the goal to determine the optimal control history $\mathbf{u}(t)$ and the corresponding state history $\mathbf{x}(t)$ on a time interval $t \in [0, t_f]$ to minimize a given cost function J subject to various constraints. A general mathematical formulation is given as follows

$$\begin{aligned}
 \min_{\mathbf{u}(t)} \quad & J = \Phi(\mathbf{x}(t_f), t_f) + \int_0^{t_f} \mathcal{L}(\mathbf{x}(t), \mathbf{u}(t), t) dt \\
 \text{s.t.} \quad & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \\
 & \Psi_{0,\text{lb}} \leq \Psi_0(\mathbf{x}(0)) \leq \Psi_{0,\text{ub}}, \\
 & \Psi_{f,\text{lb}} \leq \Psi_f(\mathbf{x}(t_f)) \leq \Psi_{f,\text{ub}}, \\
 & \mathbf{g}_{\text{lb}} \leq \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{g}_{\text{ub}},
 \end{aligned} \tag{4}$$

*UAV Lab, University of Minnesota Twin-Cities, URL: <https://www.uav.aem.umn.edu>.

[†]FALCON.m - Fast and Free Optimal Control for MATLAB[®], URL: <http://www.fsd.mw.tum.de/software/falcon-m/>.

where the cost function J is in the Bolza form with Φ and $\mathcal{L}$ being the Mayer and Langrange terms, respectively; Ψ_0 and Ψ_f are respectively the initial and final boundary constraints with $\Psi_{\cdot,lb}$ and $\Psi_{\cdot,ub}$ being the lower and upper bounds, and $\mathbf{g}$ represents the path constraints that are limited by $\mathbf{g}_{lb}$ and $\mathbf{g}_{ub}$. In this paper, the notation $\mathbf{v}(t) \in \mathbb{R}^{N_v \times N_t}$ may also be used to represent the discretized time history of a vector $\mathbf{v}$, where N_v is the number of elements in $\mathbf{v}$, and N_t is the number of discretization points. Meanwhile, the notation $\mathbf{v}(t_i) \in \mathbb{R}^{N_v}$ denotes the value of vector $\mathbf{v}$ at the specific discretized time point t_i .

For the UAV trajectory optimization problem, when an emergency landing trajectory with maximal horizontal distance $X(t_f)$ is desired, we can design the cost function of Eq.(4) in the following form

$$J = -X(t_f) + W \int_0^{t_f} \dot{\delta}_e^2 dt. \quad (5)$$

When the longest flight time is desired, we can choose the cost function in Eq.(4) as follows

$$J = -t_f + W \int_0^{t_f} \dot{\delta}_e^2 dt, \quad (6)$$

where the Lagrange term $W \int_0^{t_f} \dot{\delta}_e^2 dt$ in Eq.(5) and Eq.(6) prevents the oscillations in control commands and improves the convergence, and the weighting factor W should be chosen as a small positive constant such that the modified cost function is close enough to the original one. Elevator rate $\dot{\delta}_e$ instead of the elevator position δ_e is considered as the control input, which allows the formulations of Lagrange terms in Eq.(5) and Eq.(6), as well as a latter path constraint on actuator rate. The dynamic constraints $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ in Eq.(4) can be either the nonlinear UAV model or linearized UAV model in Eq. (3), with $\mathbf{x} = [X, h, V, \gamma, \alpha, q, \delta_e]^\top$ and $\mathbf{u} = \dot{\delta}_e$. Some boundary conditions and path constraints are given as follows

$$\Psi_0 = \mathbf{x}(0) - \mathbf{x}_{ini} = \mathbf{0}, \quad (7a)$$

$$\Psi_f = h(t_f) = 0, \quad (7b)$$

$$h(t) \geq 0, \quad (7c)$$

$$V(t) \geq V_{lb}, \quad (7d)$$

$$\alpha_{lb} \leq \alpha(t) \leq \alpha_{ub}, \quad (7e)$$

$$q_{lb} \leq q(t) \leq q_{ub}, \quad (7f)$$

$$\delta_{e,lb} \leq \delta_e(t) \leq \delta_{e,ub}, \quad (7g)$$

$$\dot{\delta}_{e,lb} \leq \dot{\delta}_e(t) \leq \dot{\delta}_{e,ub}. \quad (7h)$$

Eq.(7a) implies that when the engine fails the states of the reference landing trajectory $\mathbf{x}(t)$ at the initial time $t = 0$ should take the value of the UAV states $\mathbf{x}_{ini}$, i.e. the initialization is given by $\mathbf{x}(0) = \mathbf{x}_{ini}$. According to Assumption 2, the initial flight states of UAV $\mathbf{x}_{ini}$ are available to the landing trajectory generation software. Eq.(7b) and Eq.(7c) present respectively the final boundary conditions and the path constraint on the height $h(t)$. Path constraints Eq.(7d) and Eq.(7e) prevent UAV from stalling. The path constraint Eq.(7f) on pitch rate prevents UAV from drastic maneuvers. Path constraints Eq.(7g) and Eq.(7h) reflect the position and the rate limitations of the elevator. In order to recursively search for an optimal landing trajectory with the formulation above, an initial guess for the landing trajectory is required to start the optimization iterations. For most optimization problems a well-selected initial guess that is close to the optimal trajectory can significantly reduce the amount of iteration times and running time, which is highly desired for the low-cost embedded environment of a UAV [31].

A. Initial Guess Generation Strategy

For UAV emergency landing trajectory optimization problem, a feasible initial guess $[\mathbf{x}^0(t), \mathbf{u}^0(t), t_f^0]$, which consists of a feasible state history $\mathbf{x}^0(t)$, a control history $\mathbf{u}^0(t)$, and a landing time t_f , can be selected to be an arbitrary point from a steady-state gliding trajectory of the UAV from its initial position to the ground, where the superscript 0 denotes the index of optimization iteration. For a fixed altitude H , we can find an optimal steady-state glide velocity V_s^* , which can further be used to determine the states history $\mathbf{x}^0(t)$ by $\mathbf{x}_s = \text{trim}(V_s, H)$, introduced in Section III.A. More specifically, for the nonlinear UAV model discussed in Section III.A with $H = 100$ m and a lift-drag polar shown in Fig. 1(a), Fig. 1(b) gives the corresponding speed polar for different steady-state glides. Among these steady-state

glides, the point with minimal descent rate, marked by a circle in Fig. 1(b) with $V_{s,t_f}^* = 9.516$ m/s, is the optimal steady-state glide for maximal flight time; and the point with minimal path inclination angle $|\gamma_s|$, marked by an asterisk with $V_{s,X}^* = 11.534$ m/s, denotes the steady-state glide with maximal horizontal distance.

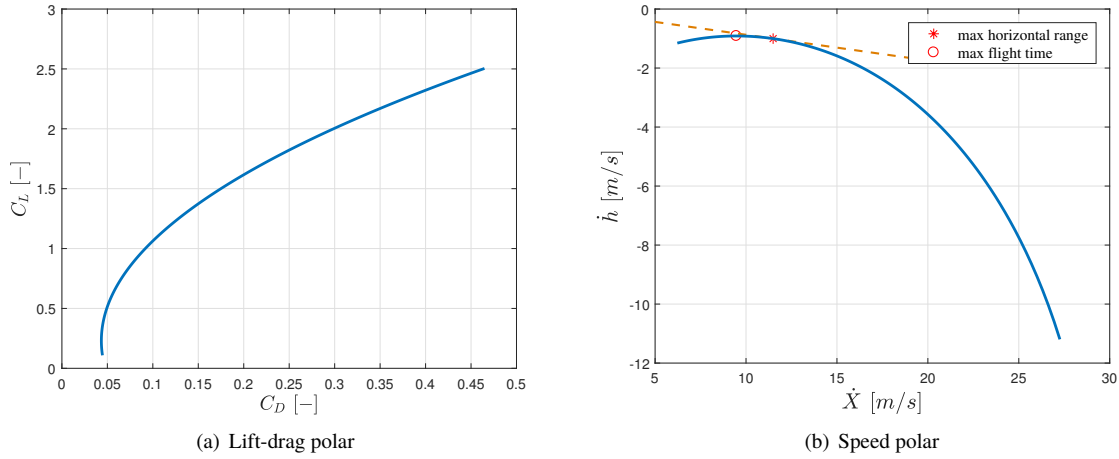


Fig. 1 Polars of steady-state glide conditions for nonlinear UAV model.

With the optimal steady-state glide velocity V_s^* , we can determine the corresponding trim condition $\mathbf{x}_s^*$ by $\mathbf{x}_s^* = \text{trim}(V_s^*, H)$. The straight-line glide trajectory $\mathbf{x}^0(t)$ with $\mathbf{x}^0(0) = [0, h_{\text{ini}}, V_s^*, \gamma_s^*, \alpha_s^*, q_s^*, \delta_{e,s}]^\top$ and $\mathbf{x}^0(t_f) = [-h_{\text{ini}}/\tan(\gamma_s^*), 0, V_s^*, \gamma_s^*, \alpha_s^*, q_s^*, \delta_{e,s}]^\top$ is selected as the state history in the initial guess, where h_{ini} denotes the UAV height above ground, when the engine fails. The control history in initial guess is assigned as $\mathbf{u}^0(t) = \delta_e^0(t) = 0$, and the flight time or terminal time in the initial guess satisfies $t_f^0 = -h_{\text{ini}}/(V_s^* \sin \gamma_s^*)$. In practice, a look-up table or an interpolation function of $\mathbf{x}_s^* = \text{trim}(V_s^*, H)$ can be pre-calculated to fast generate an initial guess $\mathbf{x}^0(t)$ for different heights H . Without the initialization procedure discussed in this section, the following optimization algorithms may not be able to find an optimal trajectory.

B. Optimization Algorithm with Nonlinear Model

With the optimization formulation and the initial guess introduced above, the algorithm of UAV emergency landing trajectory optimization with nonlinear model is summarized as follows:

Algorithm 1 Emergency Landing Trajectory Optimization with Nonlinear Model

Formulation:

- 1: Choose the objective function J satisfying the landing requirement;
- 2: Choose the nonlinear UAV model $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ as dynamic constraint;
- 3: Measure UAV flight states $\mathbf{x}_{\text{ini}}$, and formulate boundary constraints $\mathbf{x}(0) - \mathbf{x}_{\text{ini}} = \mathbf{0}$ and $h(t_f) = 0$;
- 4: Formulate all path constraints, and determine the values of the bounds, such as lower bound of speed V_{lb} , bounds of AoA ($\alpha_{\text{lb}}, \alpha_{\text{ub}}$), bounds of pitch rate ($q_{\text{lb}}, q_{\text{ub}}$), bounds of elevator ($\delta_{e,\text{lb}}, \delta_{e,\text{ub}}$), and bounds of elevator rate ($\dot{\delta}_{e,\text{lb}}, \dot{\delta}_{e,\text{ub}}$).

Initialization:

- 5: Measure UAV initial altitude above sea level H , and determine optimal steady-state glide velocity V_s^* from a look-up table or using an interpolation function;
- 6: Find the optimal steady-state glide velocity $\mathbf{x}_s^* = \text{trim}(V_s^*, H)$ from the look-up table or using an interpolation function;
- 7: Generate initial guess $[\mathbf{x}^0(t), \mathbf{u}^0(t), t_f^0]$ with initial height above ground h_{ini} and $\mathbf{x}_s^*$.

Optimization:

- 8: Solve trajectory optimization problem with FALCON.m, and return the optimized trajectory $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$ and t_f^* .
-

C. Landing Trajectory Optimization with Linear UAV Dynamics

Algorithm 1 can be modified and accelerated for computations by incorporating the linearized UAV model. As the UAV model is called and evaluated in each iteration of the optimization, replacing the nonlinear UAV model in **Algorithm 1** with a linearized UAV model can significantly speed up the optimization process. An accurate linearized model can also be generated by linearizing the UAV model around the optimal steady-glide point $\mathbf{x}_s^*$, which is determined by $\mathbf{x}_s = \text{trim}(V_s^*, H)$. In practice, the linearized system matrix $\mathbf{A} = \partial \mathbf{f}(\mathbf{x}_s^*) / \partial \mathbf{x}_{\text{lin}}$ can also be retrieved from a pre-calculated look-up table or an interpolation function with respect to UAV's initial altitude H . The optimization algorithm with linearized UAV model is summarized as follows:

Algorithm 2 Emergency Landing Trajectory Optimization with Linearized Model

Formulation:

- 1: Choose objective function J satisfying the landing requirement;
- 2: Choose linearized UAV model [Eq.\(3\)](#) as dynamic constraint;
- 3: Measure UAV initial flight states $\mathbf{x}_{\text{ini}}$, and formulate boundary constraints $\mathbf{x}(0) - \mathbf{x}_{\text{ini}} = \mathbf{0}$ and $h(t_f) = 0$;
- 4: Formulate all path constraints, and determine the values of bounds, such as lower bound of speed V_{lb} , bounds of AoA $(\alpha_{\text{lb}}, \alpha_{\text{ub}})$, bounds of pitch rate $(q_{\text{lb}}, q_{\text{ub}})$, bounds of elevator $(\delta_{e,\text{lb}}, \delta_{e,\text{ub}})$, and bounds of elevator rate $(\dot{\delta}_{e,\text{lb}}, \dot{\delta}_{e,\text{ub}})$.

Initialization:

- 5: Measure UAV initial altitude above sea level H , and determine optimal steady-state glide velocity V_s^* by look-up table or interpolation function.
- 6: Find the optimal steady-state glide velocity $\mathbf{x}_s^* = \text{trim}(V_s^*, H)$ from look-up table or interpolation function.
- 7: Generate matrix $\mathbf{A} := \partial \mathbf{f}(\mathbf{x}_s^*) / \partial \mathbf{x}_{\text{lin}}$ and $\mathbf{B}$ in [Eq.\(3\)](#) from look-up table or interpolation function.
- 8: Generate initial guess $[\mathbf{x}^0(t), \mathbf{u}^0(t), t_f^0]$ with initial height above ground h_{ini} and $\mathbf{x}_s^*$.

Optimization:

- 9: Solve trajectory optimization problem with FALCON.m, and return the optimized trajectory $\mathbf{x}^*(t)$, $\mathbf{u}^*(t)$ and t_f^* .
-

V. Illustrative Examples

The effectiveness of the UAV emergency landing trajectory optimization scheme is verified in this section with a high-fidelity UAV simulation environment. Both algorithms are employed to generate the emergency landing trajectories with the longest flight time t_f and maximal horizontal distance $X(t_f)$. Incorporated with a sampled-data $\mathcal{L}_1$ adaptive altitude tracker, the optimized trajectories are then applied to a high-fidelity UAV model in both online and offline modes. A comparison between [Algorithm 1](#) and [Algorithm 2](#) is given at the end of this section.

A. Landing without Reference Trajectory

We first consider an example of an engine-out UAV without any landing trajectory generation scheme. Throughout this section, we assume that a UAV is cruising on a steady level flight, when the engine fails. The initial flight states when engine fails, $\mathbf{x}_{\text{ini}} = [X_{\text{ini}}, h_{\text{ini}}, V_{\text{ini}}, \gamma_{\text{ini}}, \alpha_{\text{ini}}, q_{\text{ini}}, \delta_{e,\text{ini}}]^T$ in [Eq.\(7a\)](#) satisfies $X_{\text{ini}} = 0$ m, $h_{\text{ini}} = 100$ m, $V_{\text{ini}} = 17.025$ m/s, $\gamma_{\text{ini}} = 0$ deg, $\alpha_{\text{ini}} = 1.771$ deg, $q_{\text{ini}} = 0$ deg/s and $\delta_{e,\text{ini}} = -5$ deg. A feasible set of control parameters (see [Appendix](#) for details) is: Proportional gain $K_p = 0.08$, integral gain $K_i = 0.001$, desired dynamics $M(s) = (-0.2067s - 20.67)/(s^2 + 4s + 5)$, and low-pass filter $C(s) = 4/(s + 4)$. For landing without optimized reference trajectory, we assume that the UAV is trying to maintain its altitude at 100 m after the engine fails. The UAV trajectory in the vertical plane and the altitude information with respect to time are shown in [Fig. 1](#).

Without any reference landing trajectory, the maximum horizontal position that the UAV can reach is 1008.9 m, and the flight time before landing is 113.08 s. During this landing process, the range of speed V is [7.7, 17.4] m/s; the range of elevator deviation δ_e is [-4.92, -17.3] deg; the range of pitch angle θ is [2.46, 9.76] deg; the range of pitch rate is [-4.15, 4.86] deg/s, and the range of angle of attack is [3.1, 16.2] deg. Although this trajectory falls inside the flight envelope of UAV [\[19\]](#), better landing conditions can be obtained with more aggressive trajectories and landing strategies, which exploit more area inside the flight envelope.

B. Landing Trajectories with Longest Flight Time t_f

An emergency landing trajectory with maximal flight time is desired, when time is needed to evacuate the humans and facilities on the ground. Correspondingly, the cost function is selected as [Eq.\(5\)](#) with $W = 0.01$. The bounds of

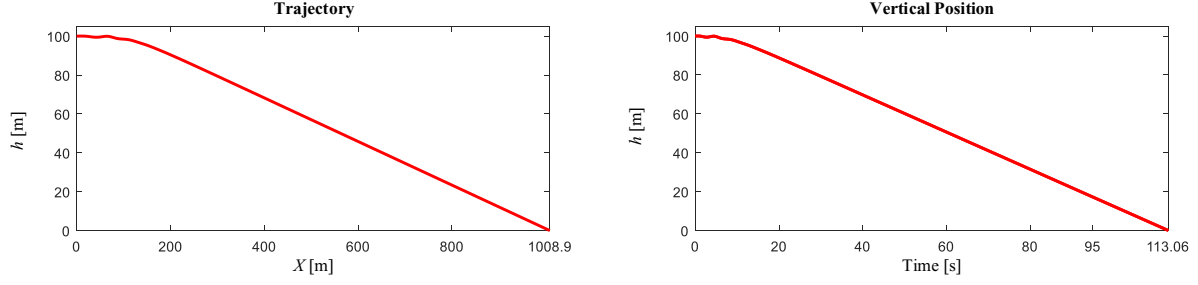
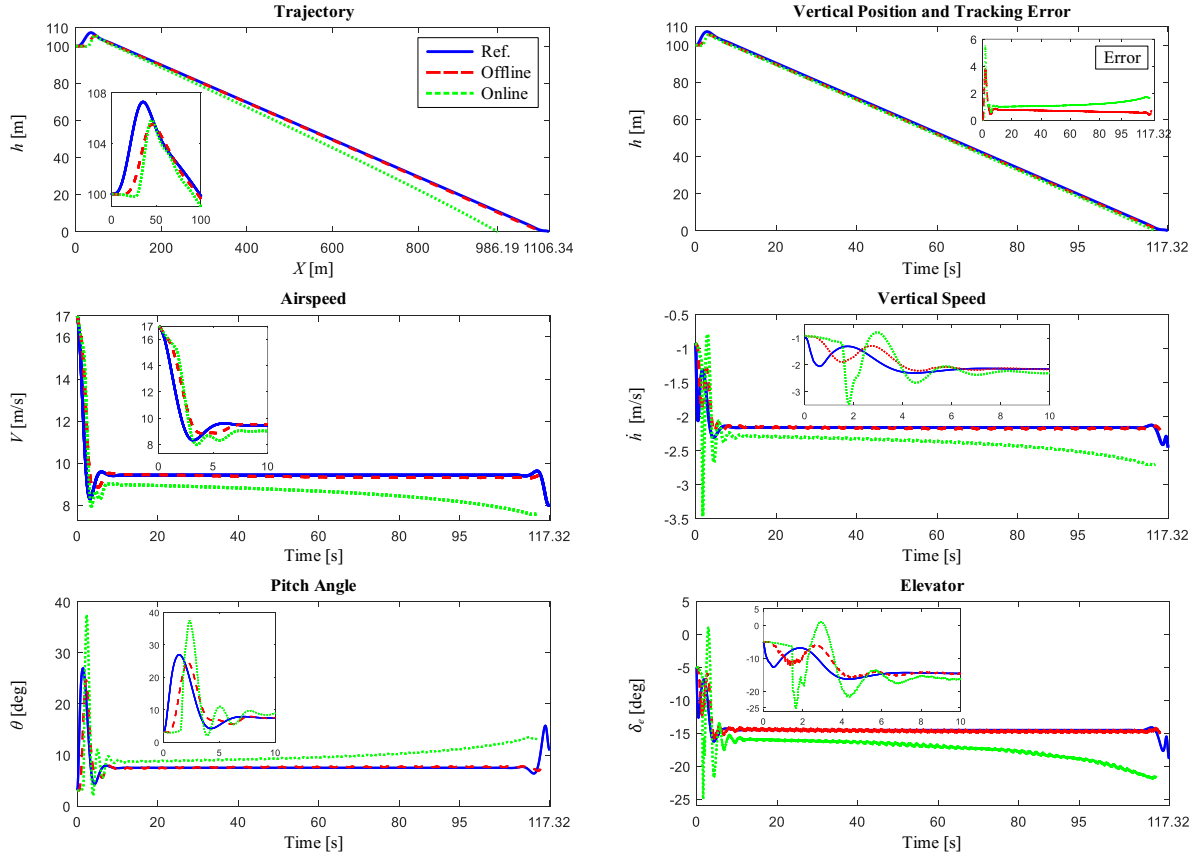


Fig. 2 Emergency landing without optimized reference trajectory.

path constraints in Eqs.(7) are selected as $V_{lb} = 8$ m/s, $\alpha_{lb} = -20$ deg, $\alpha_{ub} = 30$ deg, $q_{lb} = -30$ deg/s, $q_{ub} = 30$ deg/s, $\delta_{e,lb} = -25$ deg, $\delta_{e,ub} = 25$ deg, $\dot{\delta}_{e,lb} = -40$ deg/s, and $\dot{\delta}_{e,ub} = 40$ deg/s. By running Algorithm 1 and Algorithm 2 with the amount of discretization points $N_t = 501$, the reference landing trajectories with longest flight time are plotted as the blue solid lines in Fig. 3 and Fig. 4, respectively. Incorporated with a high-fidelity UAV simulation environment and a sampled-data $\mathcal{L}_1$ adaptive altitude tracker, the reference trajectories are investigated in two different modes, offline mode and online mode. In offline mode, we consider the scenario when the reference landing trajectories are stored on UAV or sent to UAV by the ground station, such that the UAV starts to track the reference landing trajectory immediately after the engine failed. In online mode, we first execute Algorithm 1 or Algorithm 2 to generate a landing trajectory, while the UAV is trying to maintain its initial altitude, and the UAV begins to track the reference landing trajectory after the optimization scheme returned the optimized trajectory. The landing trajectories of UAV in offline mode and online mode are respectively plotted in red dashed lines and green dotted lines.



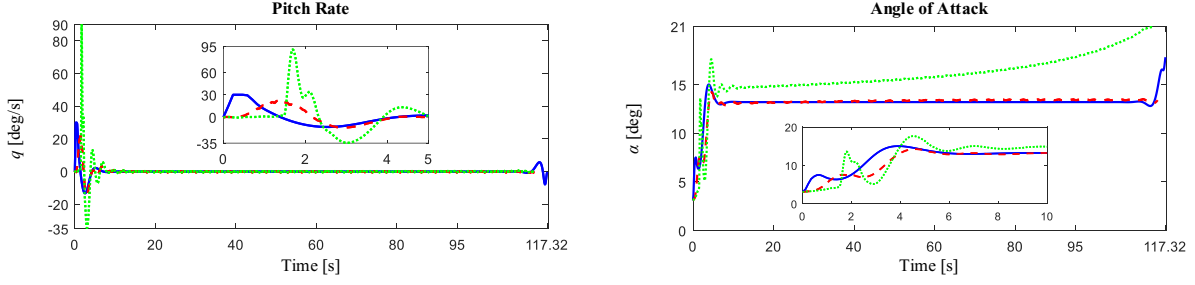


Fig. 3 Emergency landing trajectory generated by **Algorithm 1** with maximal flight time t_f .

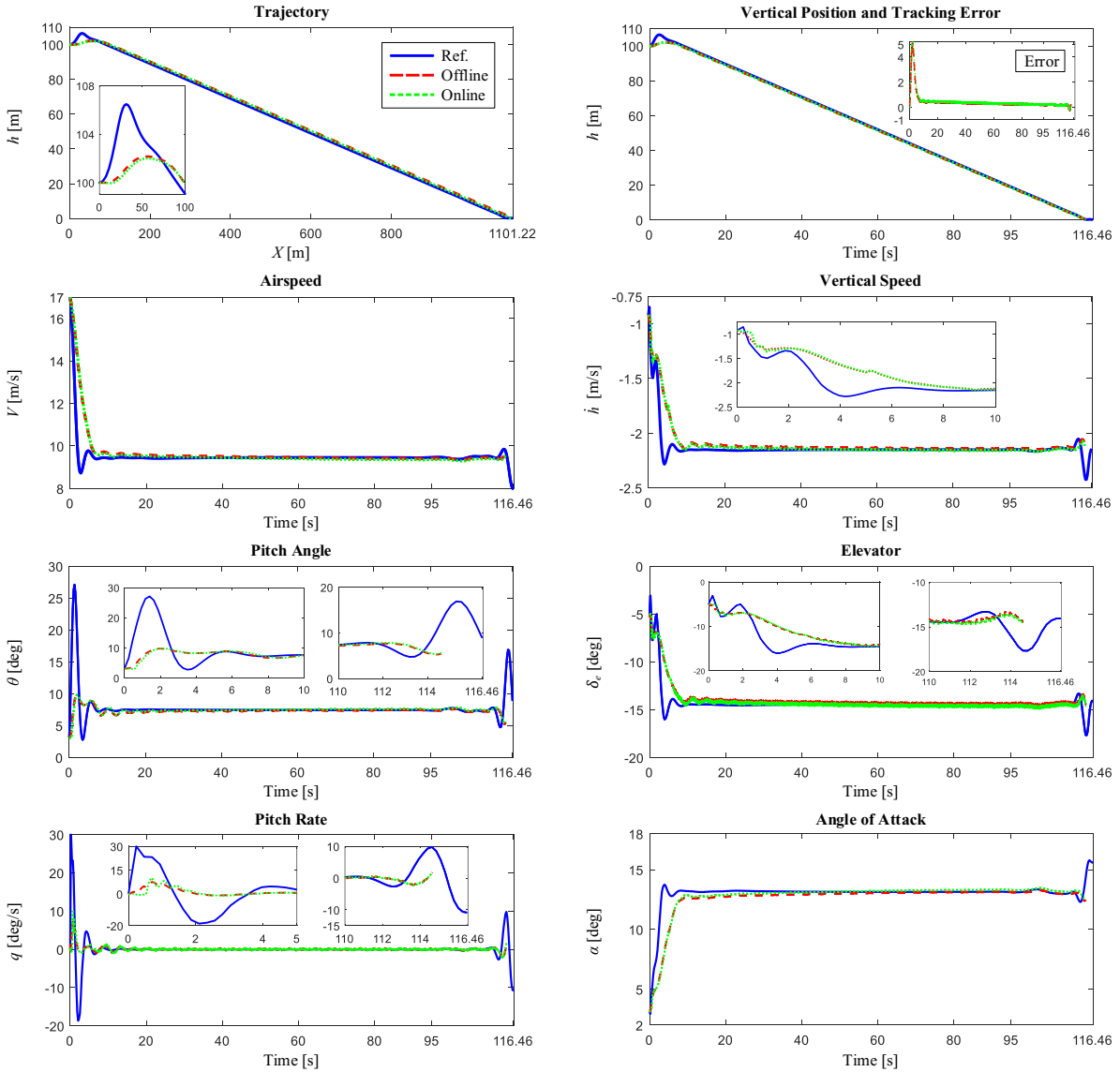


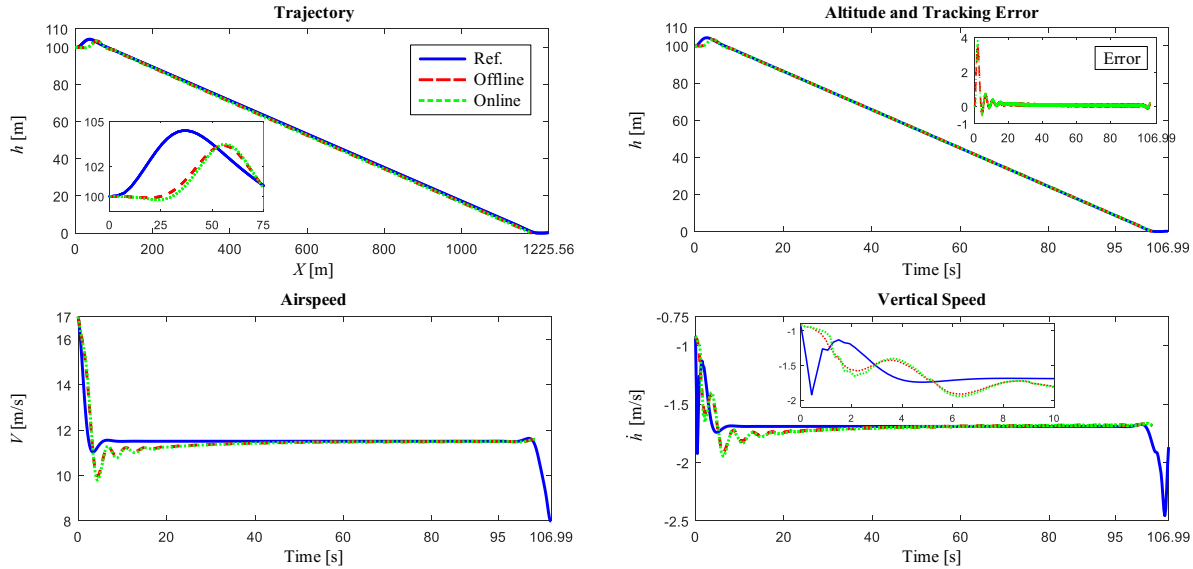
Fig. 4 Emergency landing trajectory generated by **Algorithm 2** with maximal flight time t_f .

With an Intel® Core™ i7-6700K @ 4.00GHz CPU, the running time of **Algorithm 1** in this scenario is 1.151 s with 18 iterations in the IPOPT solver. The running time of **Algorithm 2** is 0.669 s, which is 41.9% faster than the

running time of [Algorithm 1](#), with 21 iterations in the IPOPT solver. The maximal flight time of landing trajectory generated by [Algorithm 1](#) is 117.32 s with terminal horizontal position $X(t_f) = 1106.34$ m, while the maximal flight time of trajectory generated by [Algorithm 2](#) is 116.46 s with landing horizontal position $X(t_f) = 1101.22$ m. These results validate our previous proposition that due to the introduction of the linearized UAV model, optimization scheme based on [Algorithm 2](#) is faster than the scheme based on [Algorithm 1](#). However, with a more precise UAV model and longer running time, [Algorithm 2](#) is able to find the trajectory with lower objective value. Both of the reference landing trajectories generated by [Algorithm 1](#) and [Algorithm 2](#) can be divided into three major phases. The first phase is a transition phase, during which the UAV changes from the initial condition $\mathbf{x}_{ini}$ to the optimal steady-state glide $\mathbf{x}_s^*$. In [Fig. 3](#) and [Fig. 4](#), since initial speed $V_{ini} = 17.025$ m/s is larger than the optimal glide speed $V_{s,tf}^* = 9.516$ m/s, the UAV first climbs to gain potential energy and then transits to the steady-state glide condition $\mathbf{x}_s^*$. The second phase is the optimal steady-state glide phase, in which UAV continuously descends with optimal glide velocity $V_{s,tf}^*$. For the simulations shown in [Fig. 3](#) and [Fig. 4](#), this phase is approximately between 10 s and 110 s with $V_s^* = 9.516$ m/s and $\alpha, \theta, q, \delta_e$ being stable. The last phase is the landing phase, when UAV leaves steady-state glide and lands on the ground. In [Fig. 3](#) and [Fig. 4](#), we can tell that the optimizer tries to exploit all the mechanical energy to maximize t_f and $X(t_f)$ in this phase. The online trajectories in [Fig. 3](#) and [Fig. 4](#) imply that the long running time of [Algorithm 1](#) deteriorates the performance of altitude tracking controller, while the running time of [Algorithm 2](#) has little impact on the control system. The UAV altitude can follow both reference signals well in the offline mode, and the values of other flight parameters are within small intervals around the reference signals. By introducing some compensations or thresholds in the tracking control law, we can further diminish the tracking errors in the landing phase.

C. Landing Trajectories with Maximal Horizontal Distances $X(t_f)$

In the end, we consider the emergency landing trajectory with maximal horizontal distance $X(t_f)$, which will be desired when the regulators want to deviate the engine-out UAV from a critical area. Cost function is correspondingly selected as [Eq.\(6\)](#) with $W = 0.01$. Initial conditions, boundary conditions, path constraints values, and control parameters are set to be the same as the previous example. Reference landing trajectories obtained from [Algorithm 1](#) and [Algorithm 2](#) are respectively presented in [Fig. 5](#) and [Fig. 6](#).



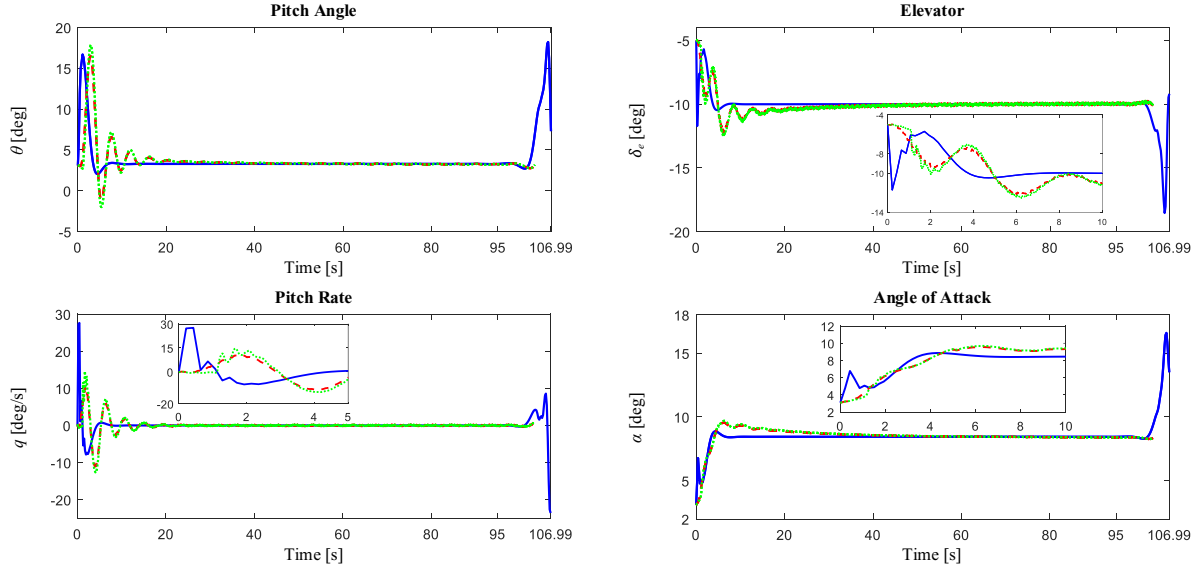
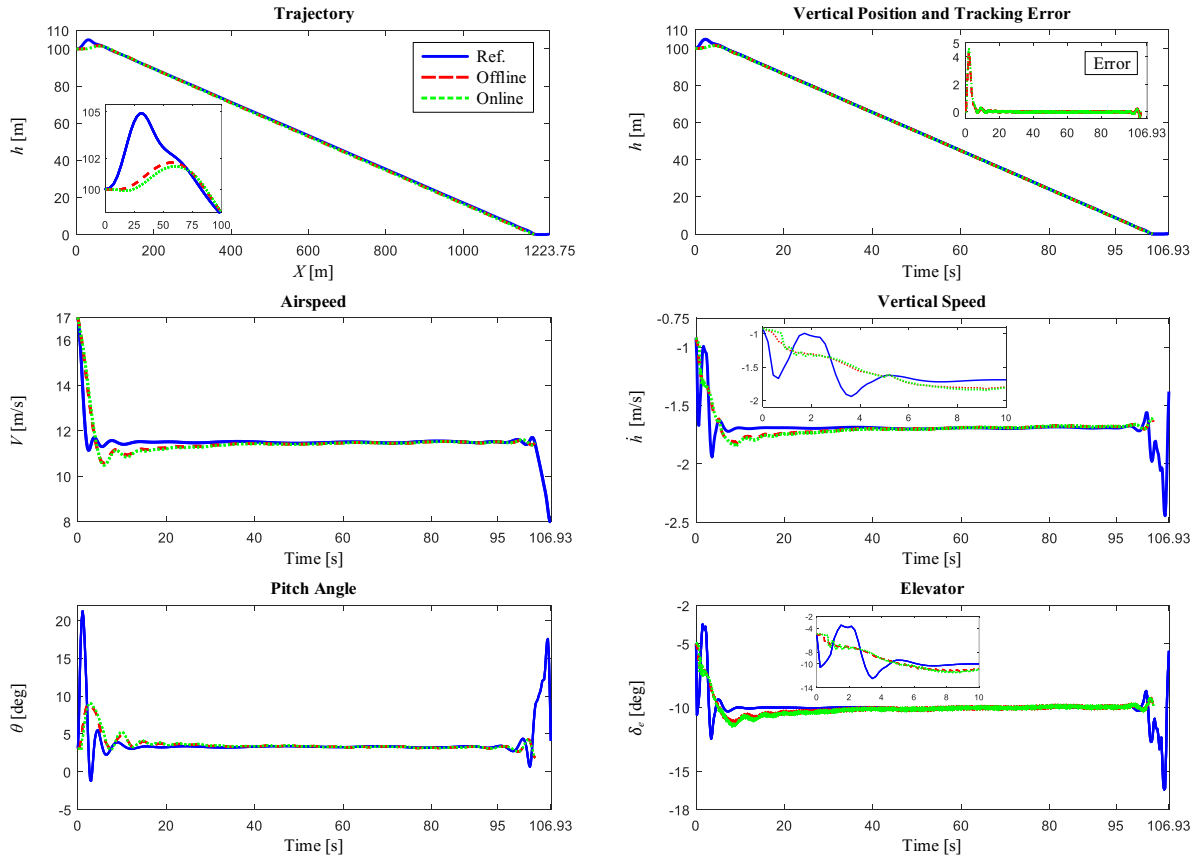


Fig. 5 Emergency landing trajectory generated by [Algorithm 1](#) with maximal horizontal distance $X(t_f)$.



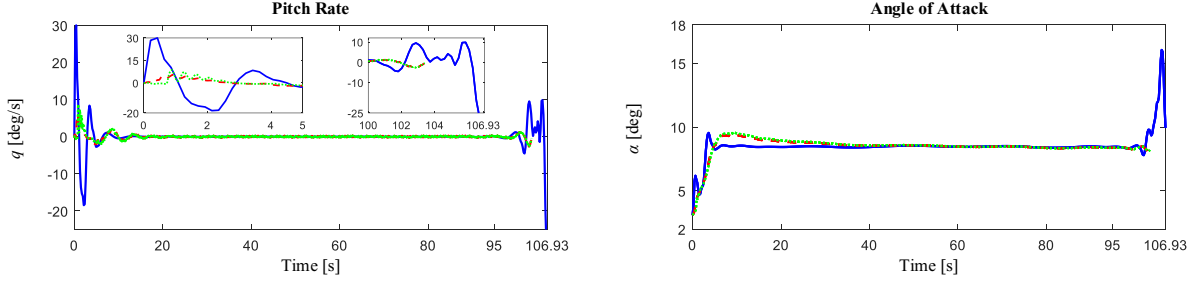


Fig. 6 Emergency landing trajectory generated by Algorithm 2 with maximal horizontal distance $X(t_f)$.

Algorithm 1 finds the solution in 1.047 s with 16 iterations in the IPOPT solver, while Algorithm 2 finds it in 0.721 s, which is 31.1% faster than the former, with 20 iterations in the IPOPT solver. The maximal horizontal distance of trajectory generated by Algorithm 1 is $X(t_f) = 1225.56$ m with flight time $t_f = 106.99$ s. The maximal horizontal distance of trajectory generated from Algorithm 2 is $X(t_f) = 1223.76$ m with flight time $t_f = 106.93$ s. The reference trajectories generated by both algorithms have comparable cost value. Incorporated with the $\mathcal{L}_1$ adaptive altitude tracker, the high-fidelity UAV can track the reference altitude signals well in both online mode and offline mode, and the values of all other flight parameters are within small intervals around the reference signals.

D. Summary

The optimization results of Algorithm 1 and Algorithm 2 with different objective functions are listed in Table 1. In general, Algorithm 1 with nonlinear UAV model can find trajectories with less cost values, while the running time of Algorithm 2 is at least 30% shorter. Comparing with the flight trajectory without reference landing signal, the optimized trajectories have better performance on the optimized parameters.

Cost Function	Algorithm	t_f [s]	$X(t_f)$ [m]	Iterations	IPOPT running time [s]
No optimization	Ref. height = 100 m	113.06	1008.9	-	-
max t_f	Alg. 1 (Nonlinear)	117.30	1106.8	18	1.151
	Alg. 2 (Linearized)	116.46	1101.2	21	0.669 (41.9% faster)
max $X(t_f)$	Alg. 1 (Nonlinear)	106.99	1225.6	16	1.047
	Alg. 2 (Linearized)	106.94	1223.7	20	0.721 (31.1% faster)

Table 1 Landing trajectory optimization results.

VI. Conclusion

An optimization and control scheme for the emergency landing of engine-out UAVs has been proposed in this paper. Two landing trajectory optimization algorithms have been introduced. An initialization procedure that generates an initial guess has been investigated to accelerate the landing trajectory optimization. Incorporated with a sampled-data $\mathcal{L}_1$ adaptive altitude tracking controller, the optimized reference landing trajectories have been simulated and verified in a high-fidelity UAV simulation environment.

Appendix

The altitude tracking controller consists of: i) An outer-loop proportional-integral altitude tracker that generates the reference pitch angle θ_r from the planned altitude h_r and UAV altitude h , and ii) An inner-loop multi-rate $\mathcal{L}_1$ adaptive pitch angle tracker that drives the UAV pitch angle θ to follow the reference signal θ_r generated by the outer-loop tracking law. The formulations of these two tracking control laws are given as follows.

A. Proportional-Integral Altitude Tracker

The altitude tracker is a discrete-time proportional-integral controller with a saturation constraint on the output signal θ_r [19]. The transfer function of the proportional-integral compensator is

$$\theta_r(z) = K_p \cdot h_e(z) + K_i \cdot \frac{T_s}{z-1} h_e(z),$$

where $\cdot(z)$ is the z -transform of variable $\cdot(t)$ in time-domain; h_e is the altitude tracking error, $T_s = 0.04$ s is the sampling period of control systems; K_p and K_i are respectively the proportional gain and the integral gain of the discrete-time PI controller. The reference pitch angle signal θ_r is constrained within $[-15^\circ, 15^\circ]$, which is an effective approach to prevent the actuator saturation in the inner-loop control framework.

B. Multi-Rate Adaptive Pitch Angle Tracker

The multi-rate adaptive pitch angle tracking control law adopted in this paper consists of a pitch angle output tracking control law, an output predictor, and an adaptation law [8, 32].

1. Tracking Control Law

Control input $u(t)$ or the deviation of elevator $\delta_e(t)$ is implemented via a zero-order hold mechanism with control period T_s :

$$u(t) = u_d[i].$$

where $t \in [iT_s, (i+1)T_s)$, $i \in \mathbb{N}$, in which $\mathbb{N} = \{0, 1, 2, \dots\}$ denotes the set of natural numbers with 0, and $u_d[i]$ is a discrete-time control input. The output of inner-loop control system $y(t)$ or UAV pitch angle $\theta(t)$ is sampled $N \in \mathbb{N}^+$ times faster with period T_s/N , where $\mathbb{N}^+ = \{1, 2, \dots\}$ represents the set of natural numbers without 0. In our simulations, we take $N = 2$ and therefore $T_s/N = 0.02$ s. Then the discrete-time output $y_d[j]$ can be expressed as

$$y_d[j] = y\left(j\frac{T_s}{N}, (j+1)\frac{T_s}{N}\right),$$

where $t \in [jT_s/N, (j+1)T_s/N]$ and $j \in \mathbb{N}^+$. Next, we consider the desired dynamics as follows

$$M(s) = C_m(sI - A_m)^{-1}B_m,$$

where the triple $\{A_m, B_m, C_m\}$ is a minimal state-space realization of $M(s)$, with A_m being Hurwitz and (C_mB_m) being nonsingular. The Laplace transform of the desired response $y_m(t)$ is given as

$$y_m(s) = M(s)K_g r(s),$$

where $K_g = -(C_mA_m^{-1}B_m)^{-1}$, $r(s)$ is the Laplace transform of $r(t)$ given by $r(t) = r_d[k]$, $t \in [kT_s, (k+1)T_s]$, and $r_d[k]$ is the discrete-time reference command or the reference pitch angle $\theta_r[k]$ from PI altitude tracker in this scenario. Finally, let $C(s)$ be the low-pass filter on control input [33]. Define $O(s) := C(s)M^{-1}(s)C_m(sI - A_m)^{-1}$, and let $\{A_o, B_o, C_o\}$ be a minimal state-space realization such that $O(s) = C_o(sI - A_o)^{-1}B_o$. The discrete-time output tracking control law is then defined as follows

$$u_d[i] = u_N(iT_s) + K_g r(iT_s), \quad (8)$$

where $i \in \mathbb{N}$, $u_N(t) = u_{Nd}[j]$ with $t \in [jT_s/N, (j+1)T_s/N]$, $u_{Nd}[j] = -C_o x_u[j]$, $x_u[j+1] = \exp(A_o T_s/N)x_u[j] + A_o^{-1}[\exp(A_o T_s/N) - I]B_o \exp(-A_o T_s/N)\hat{\sigma}_d[j]$ with $x_u[0] = 0$ and $j \in \mathbb{N}$, and $\hat{\sigma}_d[\cdot]$ is provided by the adaptation law.

2. Output Predictor

The construction of $\hat{\sigma}_d[\cdot]$ is based on an output predictor governed by

$$\begin{aligned} \hat{x}_d[j+1] &= e^{A_m \frac{T_s}{N}} \hat{x}_d[j] + A_m^{-1} \left(e^{A_m \frac{T_s}{N}} - I \right) + A_m^{-1} (B_m u_p[j] + \hat{\sigma}_d[j]), \\ \hat{y}_d[j] &= C_m \hat{x}_d[j], \end{aligned} \quad (9)$$

where the control input of predictor is defined by $u_p[j] = u(jT_s/N)$ with $j \in \mathbb{N}$.

3. Adaptation Law

Since the matrix A_m is Hurwitz, there exists a positive definite matrix P solving $A_m^T P + P A_m = -Q$ for a given positive definite matrix Q . Define

$$\Lambda = \begin{pmatrix} C_m \\ D\sqrt{P} \end{pmatrix}, \quad (10)$$

where $\sqrt{P}$ satisfies $P = \sqrt{P}^T \sqrt{P}$ and D is a matrix that is in the null space of $C_m (\sqrt{P})^{-1}$. Let $\Phi(\cdot)$ be a matrix such that

$$\Phi(T_s) := \int_0^{T_s} e^{\Lambda A_m \Lambda^{-1} (T_s - \tau)} \Lambda d\tau. \quad (11)$$

The adaptation law is governed by the following equation

$$\hat{\sigma}_d[j] = -\Phi^{-1}(T_s) e^{\Lambda A_m \Lambda^{-1} T_s} \mathbf{1} \tilde{y}_d[j], \quad (12)$$

where $\tilde{y}_d[j] = \hat{y}_d[j] - y_d[j]$, $j \in \mathbb{N}$, and $\mathbf{1}$ is given by

$$\mathbf{1} := \begin{pmatrix} I \\ 0 \end{pmatrix}. \quad (13)$$

Acknowledgments

Xiang Fang would like to acknowledge the support from the Deutsche Forschungsgemeinschaft (DFG) through the TUM International Graduate School of Science and Engineering (IGSSE) on his international travel to attend this conference. Xiang Fang and Donglei Sun would like to thank China Scholarship Council (CSC) for the financial support on their doctoral studies. This work was also partially supported by National Science Foundation (NSF) under Grant Nos. ECCS-1739732, CMMI-1663460 and ECCS-1830639, and Air Force Office of Scientific Research (AFOSR) under Grant No. FA9550-15-1-0518.

References

- [1] Jourdan, D., Piedmonte, M., Gavrilits, V., Vos, D., and McCormick, J., "Enhancing UAV survivability Through Damage Tolerant Control," *AIAA Guidance, Navigation, and Control Conference*, 2010.
- [2] Board, N. T. S., "Loss of Thrust in Both Engines After Encountering a Flock of Birds and Subsequent Ditching on the Hudson River, US Airways Flight 1549, Airbus A320-214, N106US, Weehawken, New Jersey, January 15, 2009," *Aircraft Accident Report NTSB/AAR-10/03*, 2010.
- [3] D'Andrea, R., "Guest Editorial Can Drone Deliver?" *IEEE Transactions on Automation Science and Engineering*, Vol. 11, No. 3, 2014, pp. 647–648.
- [4] Stolaroff, J., "The Need For a Life Cycle Assessment of Drone-based Commercial Package Delivery," Tech. rep., Lawrence Livermore National Laboratory (LLNL), Livermore, CA, 2014.
- [5] Scott, J. E., and Scott, C. H., "Drone Delivery Models for Healthcare," *Proceedings of the 50th Hawaii International Conference on System Sciences*, 2017.
- [6] Lin, C. A., Shah, K., Mauntel, L. C. C., and Shah, S. A., "Drone Delivery of Medications: Review of the Landscape and Legal Considerations," *American Journal of Health-System Pharmacy*, Vol. 75, No. 3, 2018, pp. 153–158.
- [7] Sun, D., Choe, R., Xargay, E., and Hovakimyan, N., "An $\mathcal{L}_1$ Adaptive Backup Flight Control Law for Transport Aircraft with Vertical-tail Damage," *AIAA Guidance, Navigation, and Control Conference*, 2016.
- [8] Jafarnejadsani, H., Lee, H., Hovakimyan, N., and Voulgaris, P., "Dual-rate $\mathcal{L}_1$ Adaptive Controller for Cyber-physical Sampled-data Systems," *IEEE Conference on Decision and Control*, 2017, pp. 6259–6264.
- [9] Shapira, I., and Ben-Asher, J. Z., "Singular Perturbation Analysis of Optimal Glide," *Journal of guidance, control, and dynamics*, Vol. 27, No. 5, 2004, pp. 915–918.

- [10] Bayen, A. M., Mitchell, I. M., Osihi, M. K., and Tomlin, C. J., "Aircraft Autolander Safety Analysis Through Optimal Control-based Reach Set Computation," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, 2007, pp. 68–77.
- [11] Atkins, E., "Emergency Landing Automation Aids: An Evaluation Inspired by US Airways Flight 1549," *AIAA Infotech@Aerospace 2010*, 2010, p. 3381.
- [12] Adler, A., Bar-Gill, A., and Shimkin, N., "Optimal Flight Paths for Engine-out Emergency Landing," *Control and Decision Conference (CCDC), 2012 24th Chinese*, IEEE, 2012, pp. 2908–2915.
- [13] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of guidance, control, and dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.
- [14] Ben-Asher, J. Z., *Optimal Control Theory with Aerospace Applications*, AIAA, 2010.
- [15] Bock, H. G., and Plitt, K.-J., "A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems," *IFAC Proceedings Volumes*, Vol. 17, No. 2, 1984, pp. 1603–1608.
- [16] Herman, A. L., and Conway, B. A., "Direct Optimization Using Collocation Based on High-order Gauss-Lobatto Quadrature Rules," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 592–599.
- [17] Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls*, DARcorporation, 1998.
- [18] Stengel, R. F., *Flight Dynamics*, Princeton University Press, 2015.
- [19] Paw, Y. C., "Synthesis and Validation of Flight Control for UAV," Ph.D. thesis, University of Minnesota, 2009.
- [20] Stevens, B. L., Lewis, F. L., and Johnson, E. N., *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous systems*, John Wiley & Sons, 2015.
- [21] Fisch, F., Lenz, J., Holzapfel, F., and Sachs, G., "On the Solution of Bilevel Optimal Control Problems to Increase the Fairness in Air Races," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 4, 2012, pp. 1292–1298.
- [22] Bittner, M., Fleischmann, B., Richter, M., and Holzapfel, F., "Optimization of ATM Scenarios Considering Overall and Single Costs," *6th International Conference on Research in Air Transportation Istanbul*, 2014.
- [23] Grüter, B., Diepolder, J., Bittner, M., Rieck, M., and Holzapfel, F., "Optimal Control-Based Altitude Profile Envelope for Emergency Landing," *CEAS EuroGNC*, 2017.
- [24] Piprek, P., and Holzapfel, F., "Robust Trajectory Optimization Combining Gaussian Mixture Models with Stochastic Collocation," *2017 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2017, pp. 1751–1756.
- [25] Wächter, A., and Biegler, L. T., "On the Implementation of An Interior-point Filter Line-search Algorithm for Large-scale Nonlinear Programming," *Mathematical programming*, Vol. 106, No. 1, 2006, pp. 25–57.
- [26] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-scale Constrained Optimization," *SIAM review*, Vol. 47, No. 1, 2005, pp. 99–131.
- [27] Hovakimyan, N., Cao, C., Kharisov, E., Xargay, E., and Gregory, I. M., " $\mathcal{L}_1$ Adaptive Control for Safety-Critical Systems," *IEEE Control Systems Magazine*, Vol. 31, No. 5, 2011, pp. 54–104.
- [28] Gregory, I. M., Xargay, E., Cao, C., and Hovakimyan, N., "Flight Test of $\mathcal{L}_1$ Adaptive Control Law: Offset Landings and Large Flight Envelope Modeling Work," *AIAA Guidance, Navigation, and Control Conference*, Portland, OR, 2011. AIAA 2011-6608.
- [29] Ackerman, K., Xargay, E., Choe, R., Hovakimyan, N., Cotting, M. C., Jeffrey, R. B., Blackstun, M. P., Fulkerson, T. P., Lau, T. R., and Stephens, S. S., "L1 Stability Augmentation System for Calspan's Variable-Stability Learjet," *AIAA Guidance, Navigation, and Control Conference*, 2016.
- [30] Ackerman, K. A., Xargay, E., Choe, R., Hovakimyan, N., Cotting, M. C., Jeffrey, R. B., Blackstun, M. P., Fulkerson, T. P., Lau, T. R., and Stephens, S. S., "Evaluation of an L1 Adaptive Flight Control Law on Calspan's Variable-Stability Learjet," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 4, 2017, pp. 1051–1060.
- [31] Bakolas, E., Zhao, Y., and Tsiotras, P., "Initial Guess Generation for Aircraft Landing Trajectory Optimization," *AIAA Guidance, Navigation, and Control Conference*, AIAA Reston, VA, 2011, pp. 8–11.
- [32] Jafarnejadsani, H., Lee, H., Hovakimyan, N., and Voulgaris, P., "A Multi-rate Adaptive Control for MIMO Systems with Application to Cyber-physical Security," *IEEE Conference on Decision and Control*, 2018.
- [33] Hovakimyan, N., and Cao, C., *$\mathcal{L}_1$ Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*, Vol. 21, SIAM, 2010.