



Damaged Airplane Trajectory Planning Based on Flight Envelope and Motion Primitives

Davood Asadi* and Mehdi Sabzehparvar†

Amirkabir University of Technology, 15875-4413 Tehran, Iran

Ella M. Atkins‡

University of Michigan, Ann Arbor, Michigan 48109

and

Heidar A. Talebi§

Amirkabir University of Technology, 15875-4413 Tehran, Iran

DOI: 10.2514/1.C032422

This paper presents an efficient approach for safe landing trajectory generation of an airplane with structural damage to its wing flying in proximity to local terrain. A damaged airplane maneuvering flight envelope is estimated by analyzing the local stability and flying quality at each trim condition. Trim state distance to a flight envelope boundary provides a novel criterion defined as the safety value index to prioritize choice of trim conditions for postdamage flight. A library of trajectory segments including trim states and transition maneuvers between trim conditions in the local neighborhood are generated using a linear quadratic regulator controller. A potential field strategy is used to rapidly define emergency landing trajectories composed as trim state sequences based on criteria including terrain avoidance, safety value index, and safe landing requirements such as touchdown heading and position, airspeed, glide slope, and bank angle. Cost criteria are balanced by adjusting weights. Landing scenarios are presented to demonstrate the capability of the proposed approach to autonomously plan safe landing trajectories.

Nomenclature

A	= state matrix
B	= input matrix
C	= controllability matrix
D	= matrix of variables in data library
DR	= Dutch roll mode
d_T	= airplane distance to terrain
g	= function of equality constraints
h	= flight altitude, m
J	= cost function
K	= state feedback gain of the controller
k	= gain value
M	= maneuvers
Ph	= Phugoid mode
p, q, r	= roll, pitch, and yaw rates, respectively, rad/s
Q	= weighting matrix
R	= roll mode
S	= spiral mode
s	= inequality constraint function
T	= trim trajectories
T_r	= roll mode time constant
T_2	= minimum time to double the amplitude
V	= total velocity in body axes, m/s
W	= weighting factor in heuristic function
z	= airplane state vector
α, β, γ	= angle of attack, sideslip angle, and flight path angle, rad

$\Delta x, \Delta y, \Delta z$	= variation in airplane position in each trajectory segment, m
Δs	= length of each trajectory segment
$\delta_{th}, \delta_a, \delta_e, \delta_r$	= engine throttle (%), aileron, elevator, and rudder deflections, deg
μ	= vector of control inputs
ξ	= damping ratio
ψ, θ, φ	= yaw, pitch, and roll angles, respectively, rad

Subscript

L	= landing condition
-----	---------------------

I. Introduction

IN MODERN aviation, flight safety and risk reduction are important issues for manufacturers and passengers. Although triple-redundancy architectures have reduced the likelihood of incidents induced by damage or failures, loss of control remains the leading cause of accidents in transport aircraft. Damage or in-flight failures in a transport aircraft can lead to significant performance degradation, which can in turn cause loss of control. To prevent such situations, researchers are working on automation enhancement to safely recover a damaged airplane. Automation can assist by characterizing and adapting to the failure or damage situation. Once postfailure performance is characterized, a new landing flight plan can be generated that respects new constraints associated with the airplane's degraded performance and stability characteristics.

There are several challenges in safely landing an aircraft after a failure or damage event. The first challenge is the requirement to characterize the airplane's new flight envelope and kinematic constraints, which requires identification of the damaged airplane and estimation of the resulting flight envelope based on the effects of system degradation on airplane performance and stability. The next challenge is to safely guide and control the damaged airplane to a safe landing. An enhanced automation strategy addressing these challenges is illustrated in Fig. 1. In this architecture, first, the control subsystem helps the pilot retain control of the airplane. While this is happening, a terrain elevation map and a motion primitives database containing reference trim states are loaded based on failure or damage scenario identification. Fault detection and identification (FDI) identifies degraded aircraft behavior along with parameters used by

Received 7 May 2013; revision received 22 June 2014; accepted for publication 18 July 2014; published online 31 October 2014. Copyright © 2014 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Copies of this paper may be made for personal or internal use, on condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 1542-3868/14 and \$10.00 in correspondence with the CCC.

*Ph.D. Student, Department of Aerospace Engineering; davoodasadi@aut.ac.ir.

†Associate Professor, Department of Aerospace Engineering; sabzeh@aut.ac.ir.

‡Associate Professor, Department of Aerospace Engineering; ematkins@umich.edu. Associate Fellow AIAA.

§Professor, Department of Electrical Engineering; alit@aut.ac.ir.

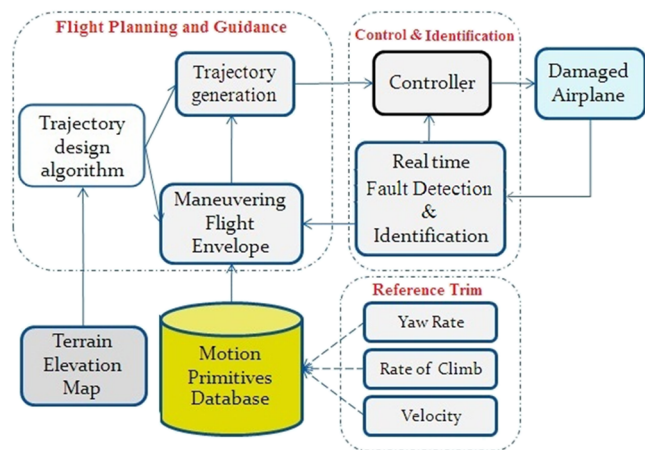


Fig. 1 Motion planning block diagram.

the flight controller to synthesize control commands. In cases where the identified condition can be analyzed a priori (e.g., engine failure or control surface jam), trim state and controller information can be recalled from a database, enabling use of an established and verified situation-appropriate control strategy. If the observed condition cannot be analyzed and captured in an available database (e.g., structural damage or failure), online FDI is required.

Once a degraded condition is identified by FDI and translated to a maneuvering flight, a feasible landing trajectory is computed that does not violate obstacle or control limitations. The tracking controller then follows the designed trajectory to landing. The integrated automation from Fig. 1 enables the automation to maneuver the airplane within the new estimated flight envelope to reach the desired landing site. In this architecture, the maneuvering flight envelope is estimated under the assumption of real-time information from an onboard FDI system. It is assumed that the real-time FDI subsystem identifies the type and performance impact of the wing tip damage fault for case studies described in this paper.

This paper describes a robust method by which a safe landing trajectory can be generated for a damaged or otherwise disabled aircraft that respects reduced flight envelope constraints. NASA has studied such methods as part of its Aviation Safety program targeting safe recovery of an airplane following adverse conditions including failures or damage [1]. Wind-tunnel tests for a variety of damage conditions have been performed by NASA on a generic transport model (GTM), which is a 5.5% scale of a commercial aircraft and is adopted as the desired case study in our research [2]. Initial research focused on development of an adaptive controller tested for a wing-damaged airplane case study. For this damage scenario, feasible trim states were derived from nonlinear equations of motion considering c.g. shift as well as aerodynamic surface reduction [3]. Real-time fault detection and identification must be incorporated in fault-tolerant flight control systems. There are many identification algorithms available in the literature, some of which are applicable to online use. Laban [4] introduces a two-step method and shows that this method is a practical approach to real-time identification of aerodynamic models. The two-step method consists of an aircraft state estimation step followed by an aerodynamic model identification step. Lombaerts [5] implements fault-tolerant flight control by making use of adaptive nonlinear dynamic inversion and real-time identification for damaged airplane using the two-step method described previously. A number of different path planning methods including cell decomposition, visibility graph, probabilistic road maps, and local search techniques were investigated for the landing path generation of a damaged airplane. Algorithm performance characteristics are compared in [6]. Numerous trajectory generation methods have been devised for piloted and automated vehicle operations. A comprehensive review of numerical flight-trajectory optimization algorithms is investigated in [7].

A randomized motion planning algorithm employs an obstacle-free guidance system in a probabilistic roadmap framework [8]. A

discrete search algorithm is presented for real-time generation of four-dimensional trajectories for a single autonomous air vehicle with known obstacles and conflicts [9]. Trim analysis is used to estimate the new flight envelope for an aircraft with a jammed elevator to design a safe and feasible landing trajectory to the desired landing site [10]. In [10], most closely related to the work presented here, an exhaustive search algorithm is applied over a limited subset of available trim states connected by transition segments; the trim state set used in search was necessarily downsampled due to the high complexity of the exhaustive algorithm. The distinctions of this paper from [10] are its use of a computationally tractable local search algorithm allowing consideration of a more extensive trim state set and shorter segments, its expanded cost function that considers terrain and the novel safety value index (SVI) described next, and its variation of cost function weights based on the environment (i.e., far from versus near the runway or an obstacle).

An adaptive flight planning approach for online trajectory generation with variable autonomy was presented in [11]. This adaptive flight planner is intended to enhance a flight management system with improved cockpit situational awareness. An emergency flight management architecture was presented in [12] to compute feasible flight plans in response to events that degrade aircraft performance in real time. The architecture in [12] includes a two-step landing site selection/trajectory generation process, which enables the aircraft to autonomously select and land at an alternate airport. Landing sites are selected after determination of the aircraft's reachable footprint and are ranked according to numerous features, including runway characteristics and current surface wind and weather conditions. An analytical landing trajectory planning algorithm based on the minimum-length Dubins path is applied to a total loss of thrust case study. The landing site search algorithm is complementary to this work, which assumes that a landing runway is specified; the search-based trajectory planning algorithm in this work is more computationally intensive but also more general than a Dubins-based path planner.

Safe landing trajectories were designed in accordance with the updated dynamics of a damaged airplane based on trim analysis in [13,14]. Building from [10], these more recent works study wing damage for a generic transport model (GTM), yielding feasible trim state "envelopes" analogous to those formulated in this paper. In [13], the authors use the landing site search algorithm from [12] with the trim-based trajectory planning algorithm from [10] applied to the GTM wing damage case. The distinctions of this work from [13] are similar to the distinctions from [10] cited previously. Yi and Atkins [14] studied the problem of planning a trajectory to explore and identify an unknown flight envelope. The potential field method was used in [14] to explore a path through trim state space as opposed to the more traditional physical space to safely evolve from an initial trim state to a final trim state; such in-flight trim state exploration provides a means to identify feasible trim states but is complementary to rather than a replacement for landing trajectory planning.

Alternate strategies for landing trajectory optimization have been published. In [15], a forced landing trajectory optimization strategy is presented using the genetic algorithm. Different forced landing scenarios with preselected touchdown locations and final headings were simulated in the presence of obstacles and unknown atmospheric turbulence. A flight trajectory generation method for distressed-aircraft-recovery was presented in [16], to maximize safe outcome probability after a distressed event by incorporating an abort airport. In-flight failures or damage in emergency path planning were investigated via probabilistic methods using a simplified kinematic model of damaged aircraft in the presence of obstacles [17]. The robust generation and optimization of three-dimensional landing trajectories were investigated for fixed-wing aircraft subjected to prescribed boundary conditions as well as maneuverability and collision avoidance constraints [18].

This work builds on the idea of a maneuver automaton (MA) first formalized in [19]. The MA is a sequence of vehicle motion primitives, which consists of trim states and maneuvers. Trajectories can be generated by "stitching" together compatible maneuvers and trim trajectories. Frazzoli et al. [19] implemented a path planning MA

database using a rapidly exploring random trees search approach. Bottaso et al. [20] presented a planning strategy based on a pre-computed library of motion primitives in three-dimensional (3-D) space. An optimization approach was used to ensure compatibility with vehicle dynamics for smoothing transitions between trim flight conditions. In [21], two-dimensional Dubins curves along with precomputed sets of motion primitives derived from the vehicle dynamic model were used for 3-D motion planning, similar to the 3-D Dubins extension presented in [12] but applied to more general motion primitives applicable to small unmanned aerial vehicle flight. Online motion planning for an agile vehicle was studied using a heuristic search algorithm that incorporates some features of greedy search with backtracking, based on a library of prespecified maneuvers [22].

A family of trim trajectories is used in many of the previous references to construct segmented paths. In our previous research [23], a wing-damaged airplane maneuvering flight envelope was estimated for different portions of wing tip loss. In this paper, it is assumed that a real-time identification algorithm identifies the fault, limited in this work to wing damage or wing tip loss. To facilitate damaged aircraft aerodynamic model specification, the flight envelopes used in this work are derived based on known percentage of wing tip loss. Local stability and flying quality are analyzed for each trim state inside the estimated flight envelope. The damaged airplane approach to the runway depends on many factors including a new metric based on trim state distance to the degraded flight envelope boundary, stability and handling quality characteristics of states in the new flight envelope distance to the landing site, runway heading, airplane airspeed, above-ground altitude, and glide slope limitations on touchdown and terrain en route. One of the innovations of this paper is that the emergency landing planner takes all these factors into consideration according to the airplane's distance from the landing runway. Another contribution is that it accounts for obstacles as constraints while locally minimizing trajectory cost based a weighted combination of these metrics. To do this in a manner applicable to real-time implementation, a potential field search algorithm over prespecified trajectories or motion primitives (i.e., a feasible trim state set) is used to rapidly generate a safe landing trajectory. The novel SVI cost term based on local stability and flying quality characteristics allows optimization to prioritize safety margin in generated landing trajectories, distinct from strictly time- or length-based cost functions that often yield optimal solutions with states near flight envelope boundaries (e.g., maximum or minimum flight path angle, minimum turn radius or rate).

As in previous work [10,13], a linear quadratic regulator (LQR) controller is designed for maneuvering (transitioning) between trim states. Distinct from previous work, a cost-weighted library of trim states and maneuvers is generated based analysis of damaged airplane stability, flying quality, and flight envelope characteristics.

The remainder of this paper is organized as follows. We first outline previous studies of damaged airplane flight envelope evaluation and extend the results to generate the library of motion primitives and transition maneuvers using an LQR controller. Next, the landing trajectory planning algorithm is described. Finally, the performance of this motion planning algorithm is explored in a variety of landing scenarios.

II. Flight Envelope Evaluation

A. Governing Equations and Aerodynamics

The damage to an aircraft can cause a shift in the c.g., which complicates the traditional equations of motion. The damaged airplane c.g. position variation " $\Delta x_{c.g.}$, $\Delta y_{c.g.}$, $\Delta z_{c.g.}$ " is calculated with respect to the c.g. position of the undamaged aircraft. The new equations of motion according to the new c.g. location of the damaged aircraft are derived as reported in [3,24]. The effect of wing tip loss on the aerodynamic stability derivatives of aircraft was explored on the GTM [2]. Oullette et al. [25] used an extended vortex lattice code to model quasi-steady aerodynamics of the damaged aircraft. A reduced-order model of aerodynamics was introduced in [26], described by a Taylor series expansion with respect to aircraft

trim state values. The formulation of [26] is used in this paper. GTM aerodynamic coefficients vary according to the percentage of wing tip loss, and new nonzero coefficients are generated due to the asymmetrical configuration of the damaged airplane that has been described in the authors' previous work [23]. The GTM is a twin-engine airplane with the capability to support differential throttle settings. The thrust from right and left engines are a function of throttle settings δ_{TR} , δ_{TL} , respectively, and are modeled as a third-order polynomial in accordance with [27].

B. Trim Analysis

The damaged airplane trim states are derived using the airplane nonlinear equations of motion:

$$\dot{z} = f(z, \mu), \quad z \in \mathfrak{R}^9, \quad \mu \in \mathfrak{R}^5 \quad (1)$$

In Eq. (1), f is a vector of nonlinear six-degree-of-freedom state equations. $z = [V, \alpha, \beta, p, q, r, \varphi, \theta]$ represents the airplane state vector, while $\mu = [\delta_{TR}, \delta_{TL}, \delta_a, \delta_e, \delta_r]$ is the column of control inputs with the following constraints [10]:

$$\begin{aligned} 0 \leq \delta_{TR}, \quad \delta_{TL} \leq 1, \quad -30 \leq \delta_a \leq 30 \\ -30 \leq \delta_e \leq 30, \quad -30 \leq \delta_r \leq 30 \end{aligned} \quad (2)$$

The equilibrium or trim state is a steady-state flight condition in which the airplane angular and linear velocities are constant, which satisfies:

$$(\dot{V}, \dot{\alpha}, \dot{\beta}) = (\dot{p}, \dot{q}, \dot{r}) = 0, \quad (\dot{\varphi}, \dot{\theta}) = 0, \quad \dot{\psi} = \dot{\psi}^*, \quad \gamma = \gamma^* \quad (3)$$

where $\dot{\psi}$ and γ are the desired turn rate and path angle, respectively. The asterisk denotes the desired trim condition. Thus, a desired trim condition represents a constant speed V^* , desired turn rate $\dot{\psi}^*$, and desired flight path angle γ^* for a reference altitude h^* .

C. Computational Procedure

One method for deriving the previous trim conditions is to solve the nonlinear constrained optimization problem that minimizes the following cost function [10]:

$$J_{\text{trim}}(z, \mu) = \frac{1}{2} \dot{z}^T Q \dot{z} \quad (4)$$

In this equation, Q is the weighting matrix describing each state's priority with respect to maintaining the trim state; Q also enables scaling to account for different units for the state parameters. The nonlinear airplane equation $\dot{z} = f(z, \mu)$ is subjected to equality constraints $G(z^*, h^*, \gamma^*, \dot{\psi}^*, V^*) = 0$ and the inequality constraint ($S(\mu) \geq 0$), which define kinematic and control input limitations, respectively. G is a nonlinear function that enforces the desired flight condition and can be expressed by

$$G = \begin{pmatrix} h = h^* \\ V = V^* \\ \tan \theta - \frac{ab + \sin \gamma^* \sqrt{a^2 - \sin^2 \gamma^* + b^2}}{a^2 - \sin^2 \gamma^*} \\ p = -\dot{\psi}^* \sin \theta \\ q = \dot{\psi}^* \cos \theta \sin \varphi \\ r = \dot{\psi}^* \cos \theta \cos \varphi \end{pmatrix} \quad (5)$$

where $a = \cos \alpha \cos \beta$, $b = \sin \varphi \sin \beta + \cos \varphi \sin \alpha \cos \beta$, and

$$S(\mu) = \begin{pmatrix} 1 - \mu_{TR} \\ 1 - \mu_{TL} \\ 30 - |\mu_a| \\ 30 - |\mu_e| \\ 30 - |\mu_r| \end{pmatrix} \quad (6)$$

D. Local Stability Analysis

A perturbation model is used to linearize the nonlinear equations about a trim state z_k^* :

$$\dot{x}_k = A_k x_k + B_k u_k \quad x_k = z_k - z_k^*, \quad u_k = \mu - \mu_k^* \quad (7)$$

Jacobian matrices are approximated numerically by deriving a set of first-order differences [10,28]. Based on root locus classification and the eigenvalue's position at each trim condition, several different markers have been defined to demonstrate the local stability at the desired equilibrium point as shown in Fig. 2 [29,30]. Stable trim conditions are preferred due to the fact that small disturbances intuitively decay to zero about these trim states. A local unstable trim state is acceptable as a feasible trim condition, if it is controllable. The airplane is considered controllable about a trim state if the controllability matrix C is of full rank n

$$C = [B_k \quad A_k B_k \quad A_k^2 B_k \quad \dots \quad A_k^{n-1} B_k] \quad (8)$$

E. Computational Results

The trim database is a discrete representation of the continuing postdamage flight envelope that defines the performance features of an airplane after a specific damage event. Based on the previous procedures, fixed trim state points along with their local stability and controllability can be generated over the spectrum of flight conditions that yields a four-dimensional database $(V, \gamma, \dot{\psi}, h)$ defined in terms of local stability and controllability. Indeed, if the controllability condition is satisfied, the trim condition is considered to be acceptable. To visualize the results, a three-dimensional slice of the original trim database can be illustrated graphically as in [13] by fixing the variable h . This depicts the maneuvering flight envelope at a fixed altitude. Figure 3 displays a trim database for 30% of the left-wing-damaged airplane at sea level. Each trim condition is characterized by its local stability as introduced in Fig. 2. The front view of the damaged airplane database at sea level is represented in Fig. 4. Although a variety of left turn rates are achievable by the airplane due to lift asymmetry, only limited right turns are feasible.

The damaged airplane maintains a wide range of potential descending and climbing options that easily enables the airplane to achieve the required altitude for landing. The left-wing-damaged airplane local stability varies from stable states at small negative turn rates to unstable saddle focus and then saddle equilibrium points at small-magnitude positive turn rates. The trim states' local stability changes to an unstable focus at large left turn rates according to Fig. 4.

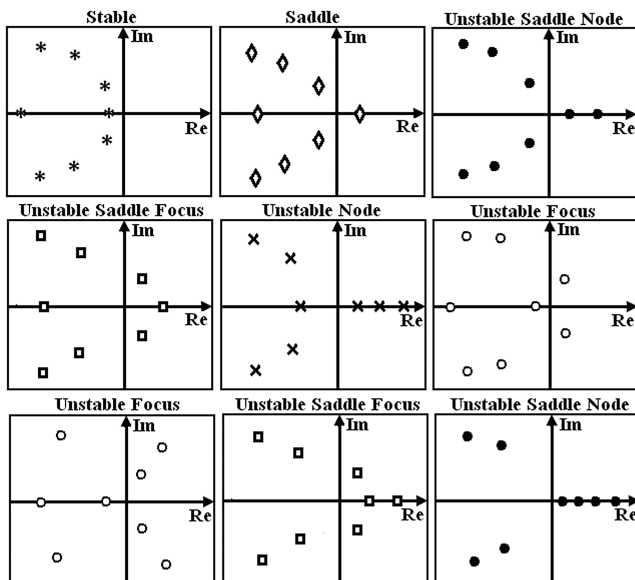


Fig. 2 Local stability based on root locus classification.

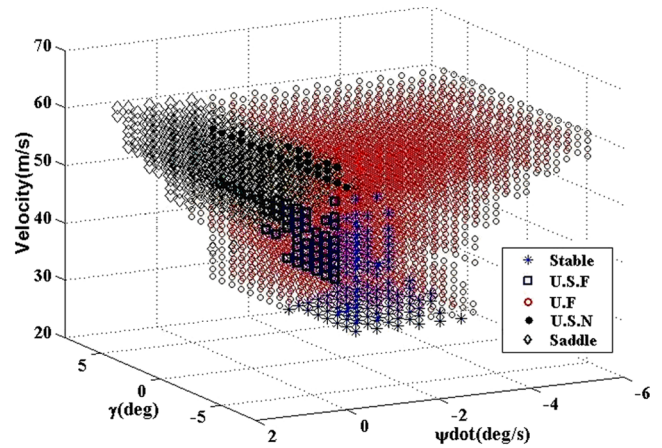


Fig. 3 Trim database for 30% left-wing damage.

Figure 5 shows the turning capability for a damaged airplane with 40% wing tip loss. Flight envelope contraction is obvious as wing tip loss percentage increases. Figure 6 represents the effect of altitude increase on the flight envelope contraction. In fact, the airplane can be trimmed at negative turn rates and velocities near the maximum airplane airspeed at high altitudes.

Although all the trim conditions are controllable, the flight envelope boundary includes the saddle and the unstable focus states in positive and negative turn rates, respectively. The saddle states represent the spiral mode instability, and the "unstable focus" represents the phugoid mode instability. The eigenvalues of steady descending/climbing turns are illustrated in Figs. 7a–7d at sea level and $V = 50$ m/s. Figure 7a represents the eigenvalues for $\dot{\psi} = -1$ deg/s, which are all stable. The dominant effect of damage is a slower response of the roll mode and stability reduction. There is also a reduction in the frequency and increase in the damping ratio of the short period mode. The damage also has a slight influence on the phugoid, spiral, and Dutch roll modes. Damping reduction of the phugoid and Dutch roll modes due to the damage is obvious. The spiral mode time constant decreases as the damage increases. Although the roll mode stability degrades substantially as the damage increases, the roll and short period modes remain stable in the flight envelope as long as the airplane can be trimmed.

Figure 7b illustrates the effect of turn rate on the eigenvalues in each motion mode. As shown, each mode's stability characteristics depend on the direction and magnitude of the turn rate. For the left-wing-damaged airplane, spiral mode stability increases in negative turn rates and reduces in positive turn rates. Roll and phugoid mode stability decreases in negative turn rates as the phugoid moves toward instability. Indeed, the phugoid and spiral eigenvalues move toward

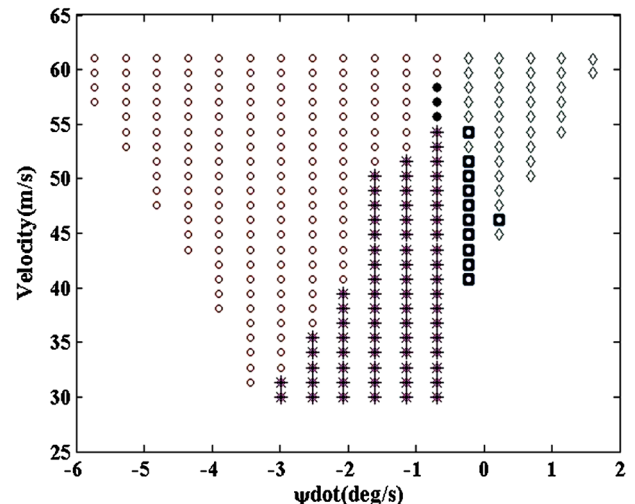


Fig. 4 View of trim states for 30% of damage.

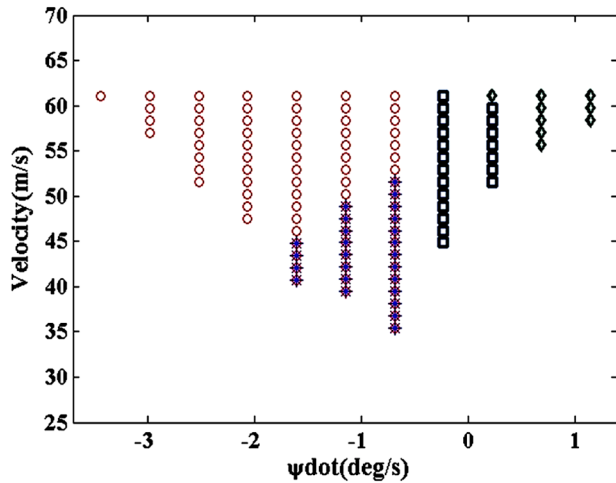


Fig. 5 Trim states for 40% of damage at sea level.

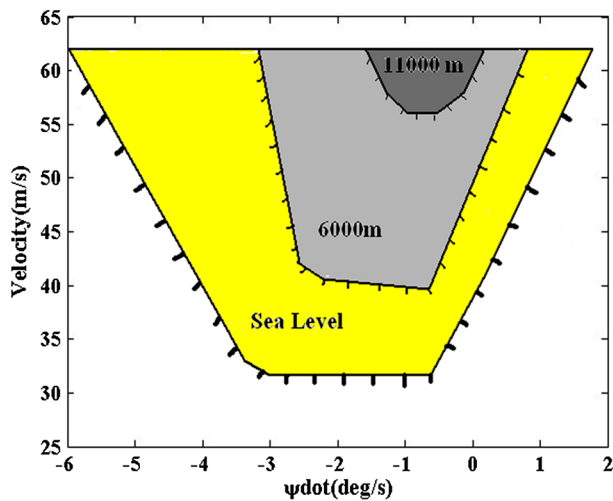


Fig. 6 Effect of altitude on flight envelope.

instability by increasing the turn rates to the left and right direction, respectively. Figure 7c illustrates the effect of flight path angle γ , which is not considerable in any of the illustrated trim states. The roll and spiral modes stability characteristics improve slightly with negative γ . As displayed in Fig. 7d, velocity highly impacts the stability characteristics in all the modes, particularly in the short period and roll modes for which stability is enhanced significantly as the velocity increases. The spiral and phugoid modes degrade slightly once the velocity increases. According to the previous analysis, airplane flying quality in all motion modes inside the maneuvering flight envelope is explored using their relevant damping ratio and frequency for each mode.

Per [31], the flying quality for each mode is categorized over three levels depending on the airplane type and flight phase. Handling quality requirements for class III aircraft and the category C phase of flight (approach, landing) are considered according to [31]. These classifications are consistent with the presented case study using a damaged business jet (class III) aircraft tasked with planning then following an approach and landing (category C) trajectory. The flying quality metric enables the landing trajectory planning to consider stability characteristics as a metric favoring solutions that are relatively easy to follow prioritization of the trim states for post-damage flight.

Based on the eigenvalues variations in different damages and flight conditions inside the maneuvering flight envelope, the levels of flying quality in each mode are defined based on the standard definitions from [31] as follows:

$$\text{Shortperiod : } \begin{cases} 0.55 < \xi < 0.85 & \text{level 1a} \\ 0.45 < \xi < 1.1 & \text{level 1b} \\ 0.35 < \xi < 1.3 & \text{level 1c} \\ 0.25 < \xi < 2 & \text{level 2} \\ \xi > 0.15 & \text{level 3} \end{cases} \quad (9a)$$

$$\text{Phugoid : } \begin{cases} \xi > 0.04 & \text{level 1} \\ \xi \geq 0 & \text{level 2} \\ T_2 > 55 & \text{level 3} \\ T_2 > 25 & \text{level 4} \\ \text{else} & \text{level 5} \end{cases} \quad (9b)$$

$$\text{Roll : } \begin{cases} T_r < 0.45 & \text{level 1a} \\ T_r < 0.9 & \text{level 1b} \\ T_r < 1.4 & \text{level 1c} \\ T_r < 3 & \text{level 2} \\ T_r < 10 & \text{level 3} \end{cases} \quad (10a)$$

$$\text{Dutchroll : } \begin{cases} \xi \geq 0.11 & \text{level 1a} \\ \xi \geq 0.08 & \text{level 1b} \\ \xi \geq 0.05 & \text{level 1c} \\ \xi \geq 0.02 & \text{level 2} \\ \xi \geq 0 & \text{level 3} \end{cases} \quad (10b)$$

$$\text{Spiral : } \begin{cases} T_2 > 12 & \text{level 1} \\ T_2 > 8 & \text{level 2} \\ T_2 > 4 & \text{level 3} \\ T_2 > 0 & \text{level 4} \end{cases} \quad (10c)$$

In Eqs. (9) and (10), handling quality range and trends [31] are considered to define new levels or sublevels. For example, according to the definition from [31], damping ratio ranges from 0.35 to 1.3 in handling quality level 1 ($0.35 < \xi < 1.3$) in short period mode. Short period damping ratio variations occur in this level 1 range for our case study. A consistent level 1 quality might suggest removal of this metric from consideration in this paper's case studies because there is no distinction using the three-level standard from the literature. However, in other scenarios, multiple handling quality levels would be expected. Additionally, although assigning few handling quality levels (e.g., three) is necessary to ensure perceptible distinction between levels in human pilot studies, a more refined categorization is posited to be appropriate to enable a more refined handling quality metric for the purpose of the landing trajectory planning task studied in this work. Therefore, as shown in Eq. (9), the baseline level 1 handling quality metric from the literature is refined by proposing level 1 decomposition into sublevels 1a, 1b, and 1c. Although not needed for case studies presented in this paper, additional decomposition could be performed for levels 2 and 3.

For the phugoid, levels 1–3 are consistent with those defined in [31]. Levels 4 and 5 are defined based on the parameter T_2 , minimum time to double the amplitude, defined in [31]. Level 4 is defined according to the trend of damping ratio reduction of 0.04 per [31]. The criterion in level 3 is time T_2 ($T_2 = 55$), which is equivalent to $\xi = -0.04$. Therefore, for level 4, the time T_2 is derived for $\xi = -0.08$, which leads to $T_2 = 25$. A new level 5 captures T_2 values less than 25. Decompositions of handling quality levels similar to that described in detail for the previous short period and phugoid modes have also been defined for other modes.

Trim states inside the maneuvering flight envelope are derived for a 33% of left-wing damage with the constraint of zero side slip ($\beta = 0$) according to the previous definition of flying or handling quality. As shown in Fig. 8, handling quality in the short period reduces slightly with right turn rates. Phugoid handling degrades with left turn rates.

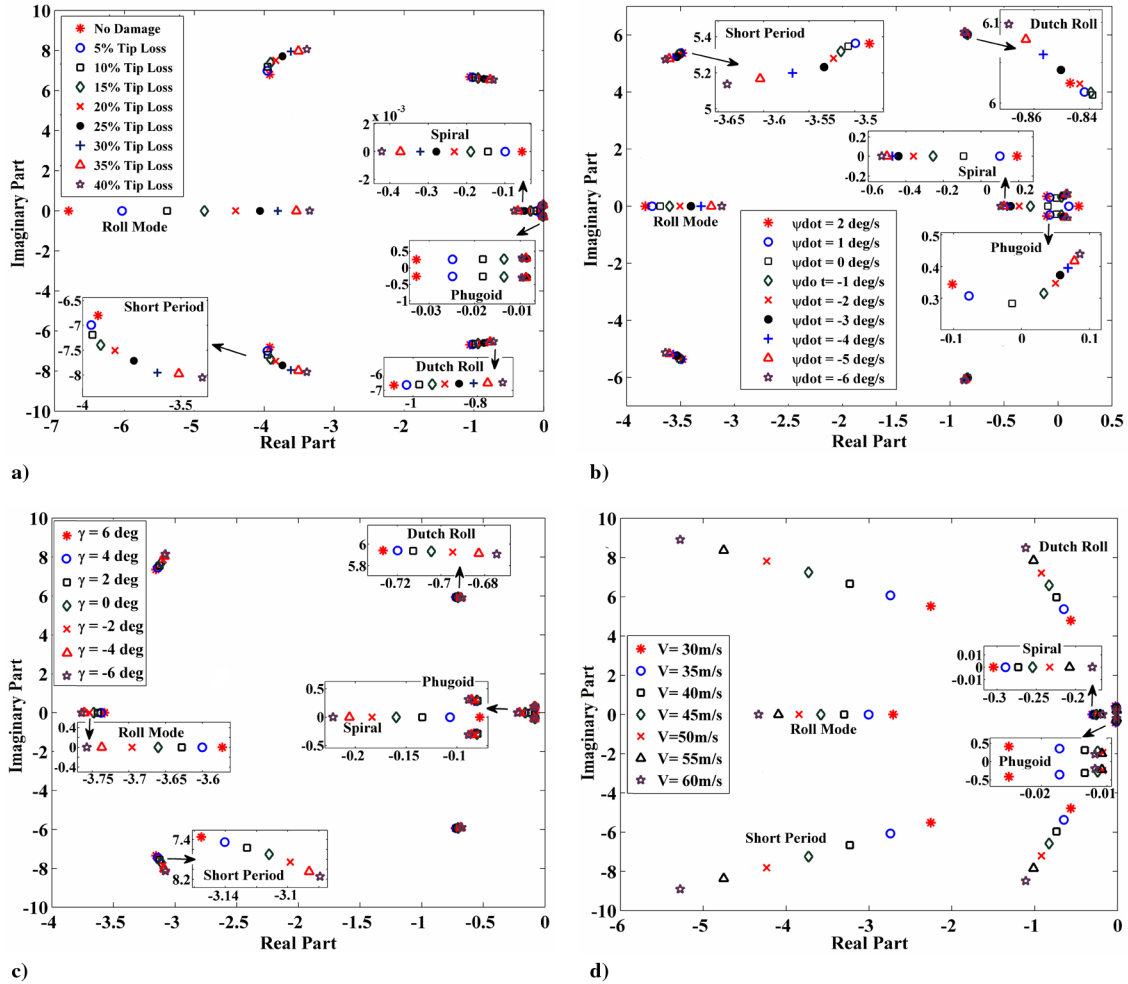


Fig. 7 Locus of eigenvalues: a) damage effect at $\dot{\psi} = -1$ deg/s, b) effect of $\dot{\psi}$ variations, c) effect of γ variations, and d) effect of V variations.

Roll handling degrades in left turn rates, but spiral handling improves in left turn rates. Handling quality in the Dutch roll mode does not change considerably at appreciable airspeeds but decreases in low-speed flight states.

A level value LV is defined, which represents a cost function weight for handling quality at the different levels. The LV is assigned in a [0.1–1] range for five levels of handling quality according to Eq. (11). In defining LV , a logical difference value of 0.2 has been considered between the five main levels of handling quality (level 1, level 2, level 3, level 4, and level 5). Because the handling qualities in sublevels are approximately half the main levels, a difference value of 0.1 is considered for the short period sublevels (level 1a, level 1b, and level 1c).

$$LV = \begin{cases} 1 & \text{level 1a} \\ 0.9 & \text{level 1b, level 1} \\ 0.8 & \text{level 1c} \\ 0.7 & \text{level 2} \\ 0.5 & \text{level 3} \\ 0.3 & \text{level 4} \\ 0.1 & \text{level 5} \end{cases} \quad (11)$$

To estimate an overall stability and handling quality characteristic for each trim state to be collectively applied as criteria for prioritized use of trim states in post damage motion planning, a safety value index (SVI) is defined for each trim state inside the maneuvering flight envelope. The SVI value derived from stability and handling quality analysis is denoted by SVI_{SHQ} and is defined as follows:

$$SVI_{SHQ} = \frac{k_{SP}LV_{SP} + k_{Ph}LV_{Ph} + k_{DR}LV_{DR} + k_RLV_R + k_SLV_S}{k_{SP} + k_{Ph} + k_{DR} + k_R + k_S} \quad (12)$$

Because the wing damage more specifically influences the lateral-directional modes, the gain values k_i , which indicate the weight of each mode in the overall estimation of SVI, are larger for these modes. In addition, the roll, short period, and Dutch roll modes are more important relative to the slow modes including the phugoid and spiral, and therefore, the gain k_i is considered larger for the fast modes. Thus, k_i value for each mode is defined as

$$k_{SP} = 2.5, \quad k_{Ph} = 1, \quad k_{DR} = 2, \quad k_R = 3, \quad k_S = 1.5 \quad (13)$$

A sensitivity analysis on selecting the k_i values relating handling qualities to SVI is presented in Fig. 9. The sensitivity analysis provides insight into how the gains affect SVI and assists in selecting appropriate gain values. Because of the previous sensitivity analysis, it is concluded that SVI_{SHQ} sensitivity to the short period and Dutch roll gains is not considerable and thus can be ignored. Increasing the phugoid gain leads to larger SVI_{SHQ} values in negative turn rates, but inversely increasing the roll and spiral gains results in larger SVI_{SHQ} values in small negative and positive turn rates. Figure 10 illustrates the SVI_{SHQ} over a range of turn rates and velocities within the flight envelope.

A safety value index relating trim state distance (in respective flight envelope dimensions) to the flight envelope boundary is defined. Denoted by SVI_{FEB} , this metric is calculated based on the normalized distance of each trim state to the flight envelope

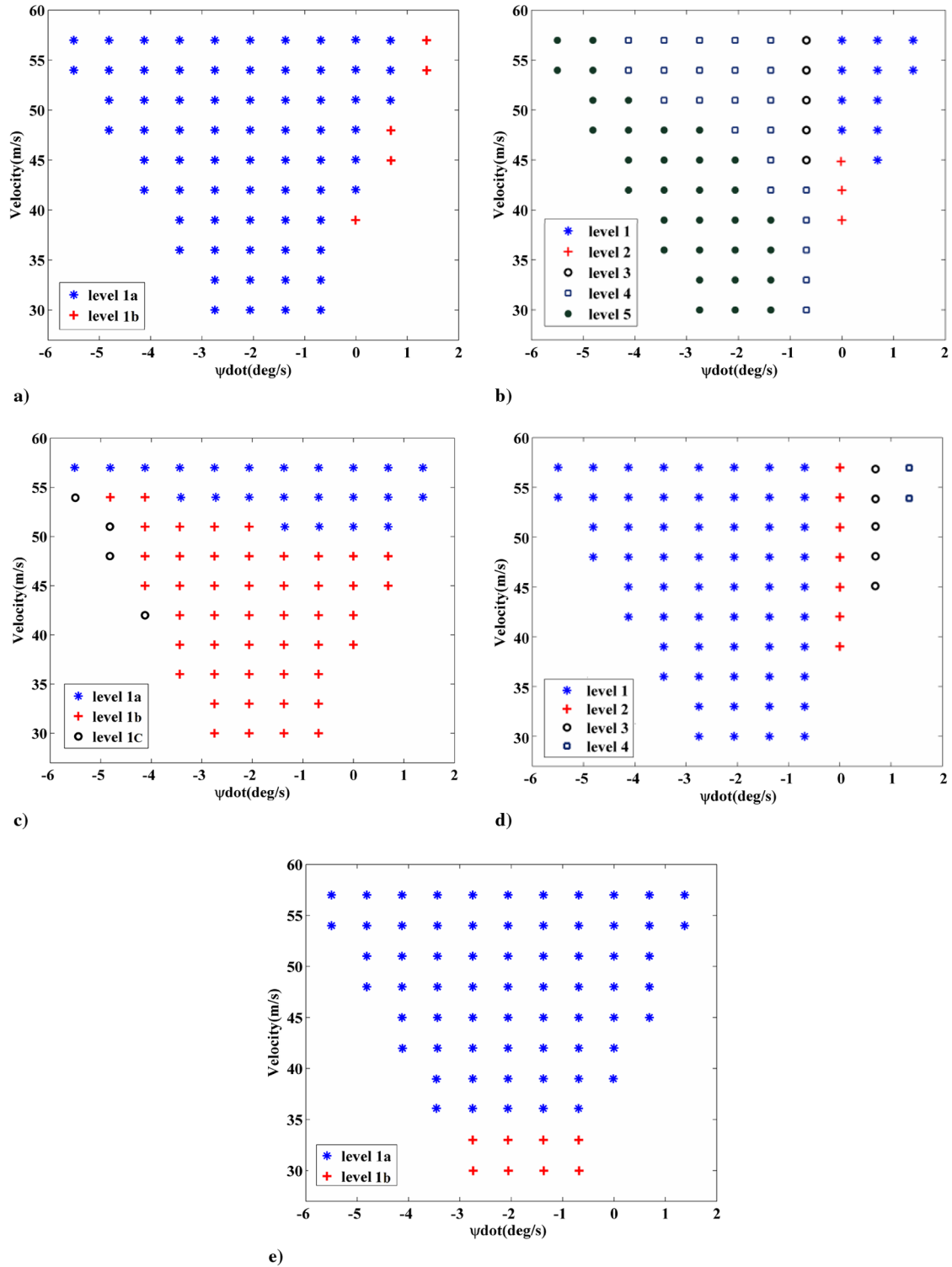


Fig. 8 Handling quality of each mode inside the maneuvering flight envelope at sea level: a) short period, b) phugoid, c) roll mode, d) spiral, and e) Dutch roll.

boundary. The flight envelope boundary and parameters indicating distance to flight envelope boundary are illustrated in Fig. 11.

Trim states distance to the flight envelope boundary are represented by an SVI denoted by SVI_{FEB} , calculated as follows:

$$SVI_{FEB} = 0.5 \left(\frac{D_{\dot{\psi}}}{D_{\dot{\psi}_{max}}} + \frac{D_V}{D_{V_{max}}} \right) \quad (14)$$

where $D_{\dot{\psi}_{max}}$ and $D_{V_{max}}$ are the maximum distances to the flight envelope boundary over all identified trim conditions with respect to turn rate and velocity, respectively. $D_{\dot{\psi}}$ and D_V are calculated for each

trim condition according to Fig. 11. As shown in Fig. 12, trim states nearer to $\dot{\psi} = -2$ deg/s are assigned the highest SVI_{FEB} values due to the preference for turning induced by the asymmetric wing damage, while increasing velocity increases the SVI_{FEB} value because aerodynamic force application capability improves with increased flow speeds.

The total SVI value (SVI_T), combining SVI_{SHQ} and SVI_{FEB} , is calculated as

$$SVI_T = k_{SHQ} SVI_{SHQ} + SVI_{FEB} \quad (15)$$

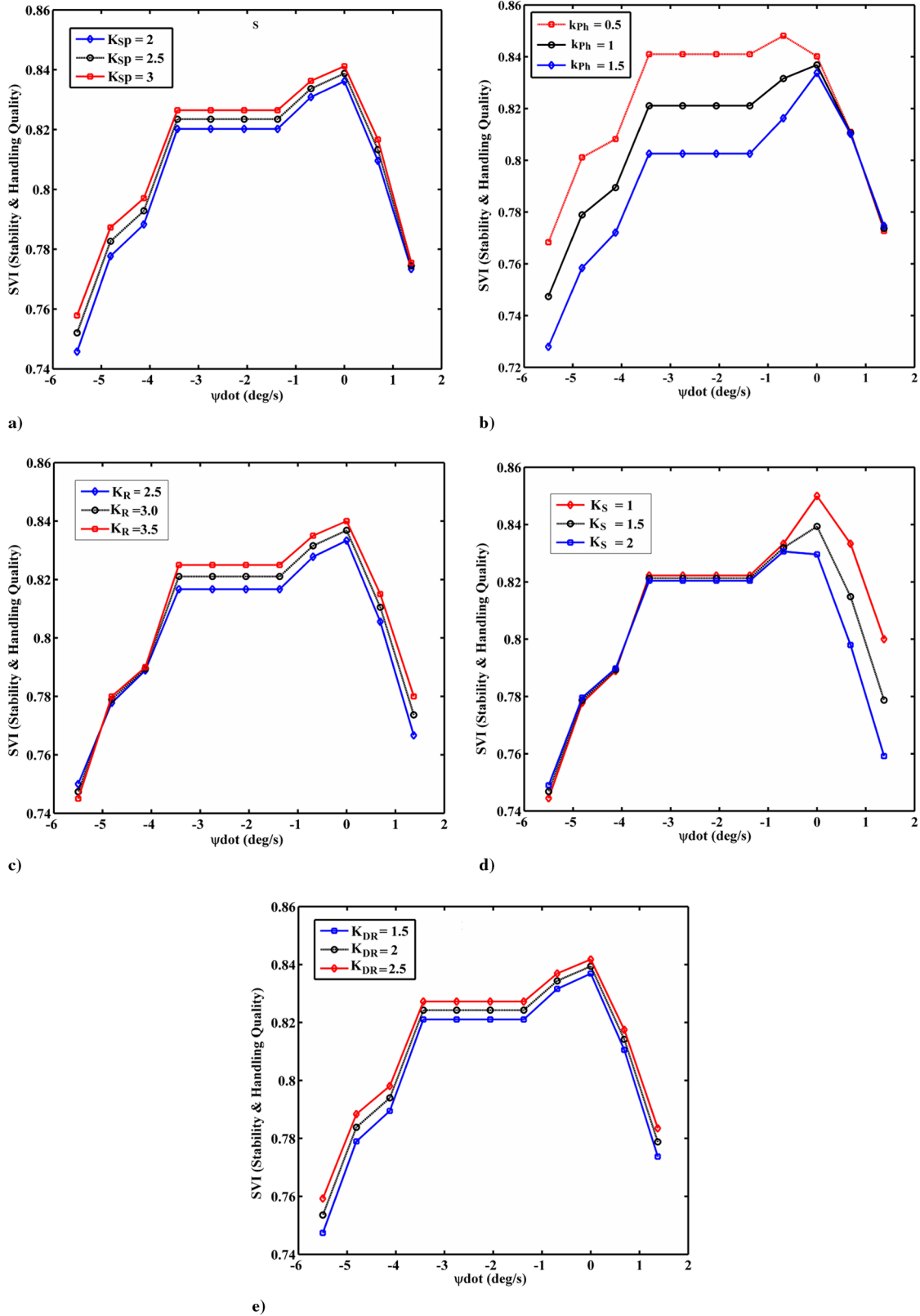


Fig. 9 SVI sensitivity to each mode gain variation for turn rate changes inside the flight envelope: a) short period, b) phugoid, c) roll mode, d) spiral, and e) Dutch roll.

Because the range of SVI_{FEB} is almost four times the range of SVI_{SHQ} , in this work, $k_{SHQ} = 4$ to balance the two SVI criteria. Total SVI sensitivity SVI_T to k_{SHQ} selection is illustrated in Fig. 13. According to Fig. 13, increasing k_{SHQ} slightly increases SVI_T at turn

rates away from that with the maximum SVI and shifts the maximum SVI_T from $\dot{\psi} = -2$ deg/s toward $\dot{\psi} = 0$ deg/s.

Total SVI (SVI_T) is illustrated in Fig. 14. SVI_T is used as the single SVI metric in the trajectory planner described further.

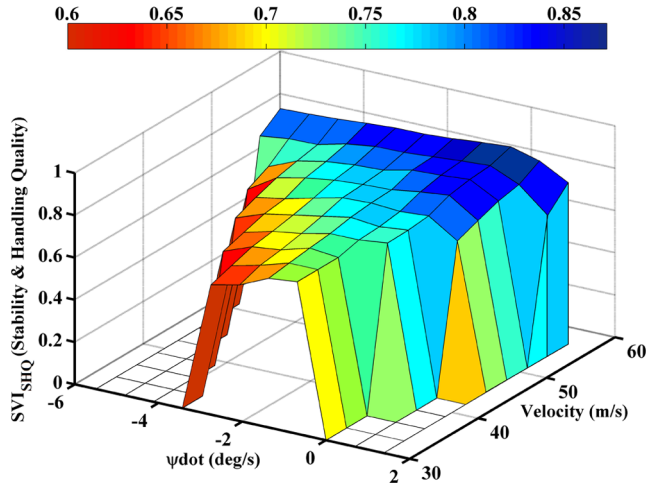


Fig. 10 Trim states SVI_{SHQ} due to stability and handling quality.

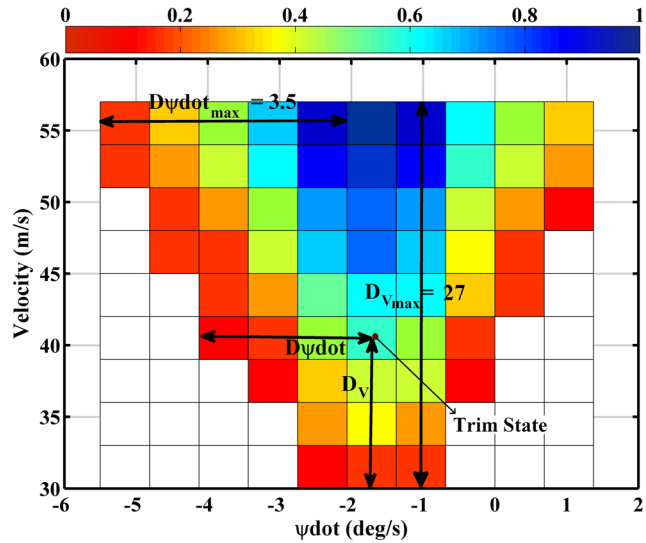


Fig. 11 Distance to flight envelope boundary.

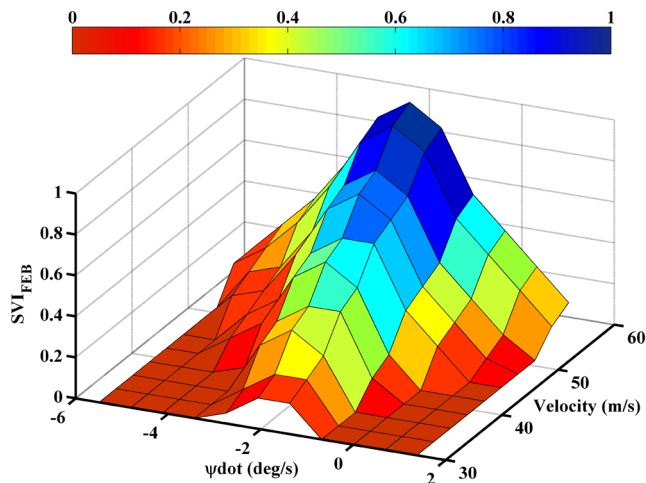


Fig. 12 Trim state SVI_{FEB} .

III. Flight Envelope Evaluation

The task of the flight planner is to identify an appropriate sequence of trim states from the initial airplane state where damage occurs to the desired landing site position and heading. A trajectory consists of

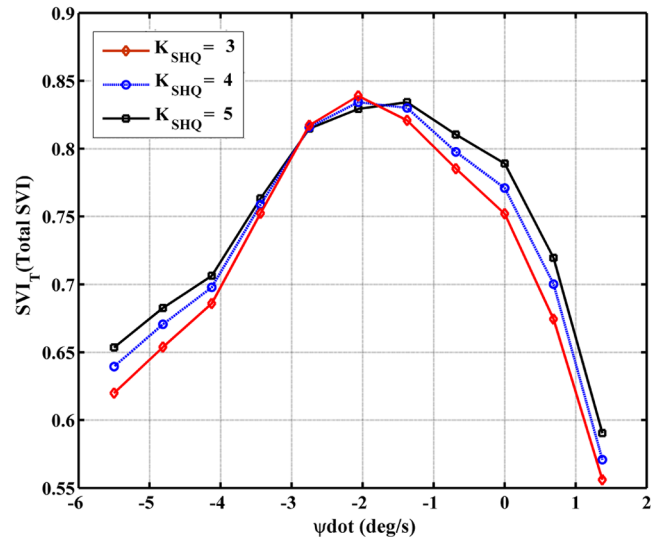


Fig. 13 Total SVI sensitivity to K_{SHQ} gain at constant velocity.

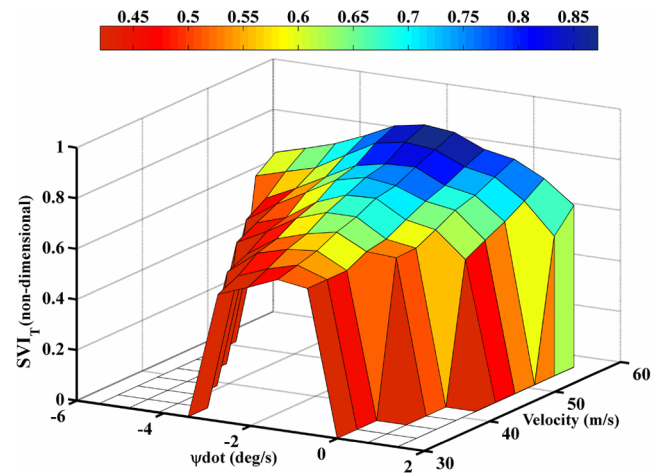


Fig. 14 Trim states total safety value index SVI_T for the wing damage case study.

a sequence of states or motion primitives that can be divided into two types. The first class of motion primitives is the steady state trajectory of the system known as the trim trajectory (T). The second class of motion primitives is the finite time transition between the trim states known as the maneuver (M). A neighborhood set is defined for each trim condition and maneuvers are generated based on the neighboring states of each trim. Fixed time segmented trajectories are derived for each trim state and its pertinent neighborhood using a linear quadratic regulator (LQR), controller and the variations in states ($\Delta x, \Delta y, \Delta z, \Delta \psi, \Delta \gamma$) are saved in a data structure to be used for real-time motion planning. The data structure also contains a safety value index (SVI) for each trim and maneuver segmented trajectory.

A. Successor States

A full trajectory will consist of N trim states and $N - 1$ maneuvers. To identify a trajectory based on motion primitives, a time duration of 5 s is considered for each trimmed flight segment and each maneuver. Motion planning takes place in discrete space by considering a fixed time step for trim and maneuver trajectory segments. The generation of maneuver trajectories requires specifying successor trim states.

Consider a discrete grid $\{[i\Delta V, j\Delta \gamma, k\Delta \psi]\}$, where $\Delta V = 3$ m/s, $\Delta \gamma = 0.5$ deg, $\Delta \psi = 0.7$ deg/s determine the grid spacing, and i, j, k take integer values.

Each trim condition in the maneuvering flight envelope has a set of neighboring grid points. Let T represent the current trim state. The neighborhood states (N) of T are defined as all trim states accessible

from T in one step. In the current study, i, j, k are considered in the immediate neighborhood (subscripts of ± 1) for generating the set N for each T . Thus, the successor states of each T are the set N of the 26 adjacent states from T . While generating segmented trajectories for each trim condition, 26 maneuvers are possible, and therefore each trim state and its neighboring states constitute 27 possible trajectory segments.

To minimize the error of tracking, an LQR controller is used to generate the maneuvers that transition between neighboring trim states from T . The segmented trajectories of trim states and maneuvers created by the controller are saved in the database library for the purpose of motion planning.

B. Tracking Controller

Based on the linearized perturbation model of the damaged aircraft, an LQR controller that operates on airplane states $x = [V \ \alpha \ \beta \ p \ q \ r \ \phi \ \theta]^T$ is designed to generate the transition maneuver between each pair of neighboring trim states [32]. Let x_e and u_e be the error between airplane state and the desired trim state; the error equation is derived as the following:

$$x_e = x - x^*, \quad u_e = u - u^* \quad \dot{x}_e = Ax_e + Bu_e \quad (16)$$

To incorporate the integral of output error, an additional vector $z \in \mathcal{R}^3$ state is defined such that

$$y_e = y - y^*, \quad y = [V \ \gamma \ \dot{\psi}]^T \quad \dot{z} = y_e = Cx_e \quad (17)$$

The matrix $C_{3 \times 8}$ is calculated numerically like matrices A and B , using the following equation:

$$C = \left. \frac{\partial y}{\partial x} \right|_{x^*, u^*} \quad (18)$$

A new augmented state vector $X = [z \ x_e]^T$ is defined, and the new augmented state equation becomes

$$\dot{X} = \begin{bmatrix} \dot{z} \\ \dot{x}_e \end{bmatrix} = \begin{bmatrix} 0 & C \\ 0 & A \end{bmatrix} X + \begin{bmatrix} 0 \\ B \end{bmatrix} u_e = A_{\text{Aug}}X + B_{\text{Aug}}u_e \quad (19)$$

The closed-loop tracking error dynamics are calculated using the following equation:

$$u_e = -KX \quad \dot{X} = (A_{\text{Aug}} - B_{\text{Aug}}K)X \quad (20)$$

where K is the state feedback gain and is designed using LQR theory as

$$K = R^{-1}B_{\text{Aug}}^T P \quad (21)$$

In Eq. (19), P is a positive-definite and is obtained by solving the algebraic Riccati equation:

$$0 = A_{\text{Aug}}^T P + PA_{\text{Aug}} + Q - PB_{\text{Aug}}R^{-1}B_{\text{Aug}}^T P \quad (22)$$

where Q and R are weighting matrices on the states and control actions, respectively, which are selected appropriately to minimize the tracking cost:

$$J = \int (X_e^T Q X_e + u_e^T R u_e) dt \quad (23)$$

Considering Eq. (18), the control is derived as

$$u_e = -K_i z - K_p x_e \quad (24)$$

where K_i and K_p are integral and proportional submatrices of K :

$$u = u^* + K_p x^* - K_p x - K_i z \quad (25)$$

The trajectory time history over all state variables is generated for each trajectory segment whether generated by holding the trim states or the maneuvers between trim states. The airplane states in each maneuver are linearly interpolated between the states of two successor trim states to yield an approximate maneuver states at any time instant as needed. u^* and y^* are thus calculable both for each trim state and by interpolation in the maneuvers between the desired trim states. Note that this maneuver generation controller is not to be confused with the trajectory tracking controller. This controller is used solely to generate maneuvers between pairs of neighboring trim points. The segmented trajectory generated by the previous controller is saved in the data library to be used for trajectory generation.

Real-time flight plan following to accurately execute emergency landing plans defined in this work will require use of a trajectory tracking controller [33]. Careful design of such a controller, however, is not the focus of this work; the simpler LQR controller is sufficient for definition of feasible trim states and their approximate costs. The procedure we propose for defining and using trim states and transitions between these states is generally applicable. The performance of the LQR controller during a trim transition from an initial trim condition $V_T^* = 55$ m/s, $\gamma^* = 3.75$ deg, and $\dot{\psi}^* = 0$ deg/s to a new desired trim flight condition $V_T^* = 60$ m/s, $\gamma^* = 2.35$ deg, and $\dot{\psi}^* = -2$ deg/s is shown in Fig. 15.

All maneuvers between the possible successor trim states are generated with time intervals of 5 s using a typical command signal as presented in Fig. 15. Transients have been observed, as in Fig. 15, to settle within this time interval because the magnitude of the changes in neighboring trim reference states is small. The transition maneuver database contains a matrix $D = [\Delta x, \Delta y, \Delta z, \Delta \psi, \Delta \gamma, \Delta s, \text{SVI}_T]$ and is generated for each trim state and maneuver using the previous controller for which Δs and SVI_T are the length and total safety value index for a trajectory segment, respectively. Note that Δs is defined as

$$\Delta s = \int_0^t \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2} dt \quad (26)$$

According to Fig. 14, SVI_T can be computed for each trim condition. For each maneuver, it is defined as the mean value of the start and end trim state SVI_T values. For the previous maneuver, the D vector is $D = [367.5, 3.2, 3.21, -2, -1.4, 367.53, 0.785]$.

An example family of maneuvers to all 27 neighboring trim states for an initial T ($V_T^* = 55$ m/s, $\gamma^* = 3.75$ deg, and $\dot{\psi}^* = 0$ deg/s) is illustrated in Fig. 16. Each segmented trajectory is illustrated over 5 s in duration. According to the neighboring states for each trim condition, the segmented trajectories are derived and stored in a maneuver database. This allows vehicle dynamics to be captured in the trim and maneuver databases, which eliminates the need to propagate dynamics during online trajectory planning. For this work, the trim and maneuver database is derived assuming zero wind velocity. To compensate for steady wind and gusts, the planned trajectory is then executed in a trajectory-tracking feedback control loop. Steady winds or gusts may be sufficiently large that even a capable trajectory tracking controller cannot compensate resulting in deviation or drift away from the desired inertial path (x, y , and z). Strong winds are challenging for any trajectory planner to handle. Previous work has focused on modeling steady wind (e.g., at different altitudes) in flight planning and guidance laws. For example, Kluever [34] proposes a real-time algorithm for incorporating wind into the computation of flight path angle for recovery of a reusable launch vehicle. Iteration over this single parameter enables computation and update of the trajectory in real-time, during the descent. Bares et al. [35] propose an adaptive guidance strategy capable of tracking Dubins paths in the presence of wind. Techy and Woolsey [36] derive a minimum-time path planning algorithm that accounts for steady uniform winds assuming a “turn-straight-turn” extremal solution form.

For a damaged aircraft, an “optimal” solution will maximize chances of a safe landing; such an optimal solution will not necessarily be characterized by a Dubins-like extremal path, nor will adjustment of one parameter such as flight path angle be possible

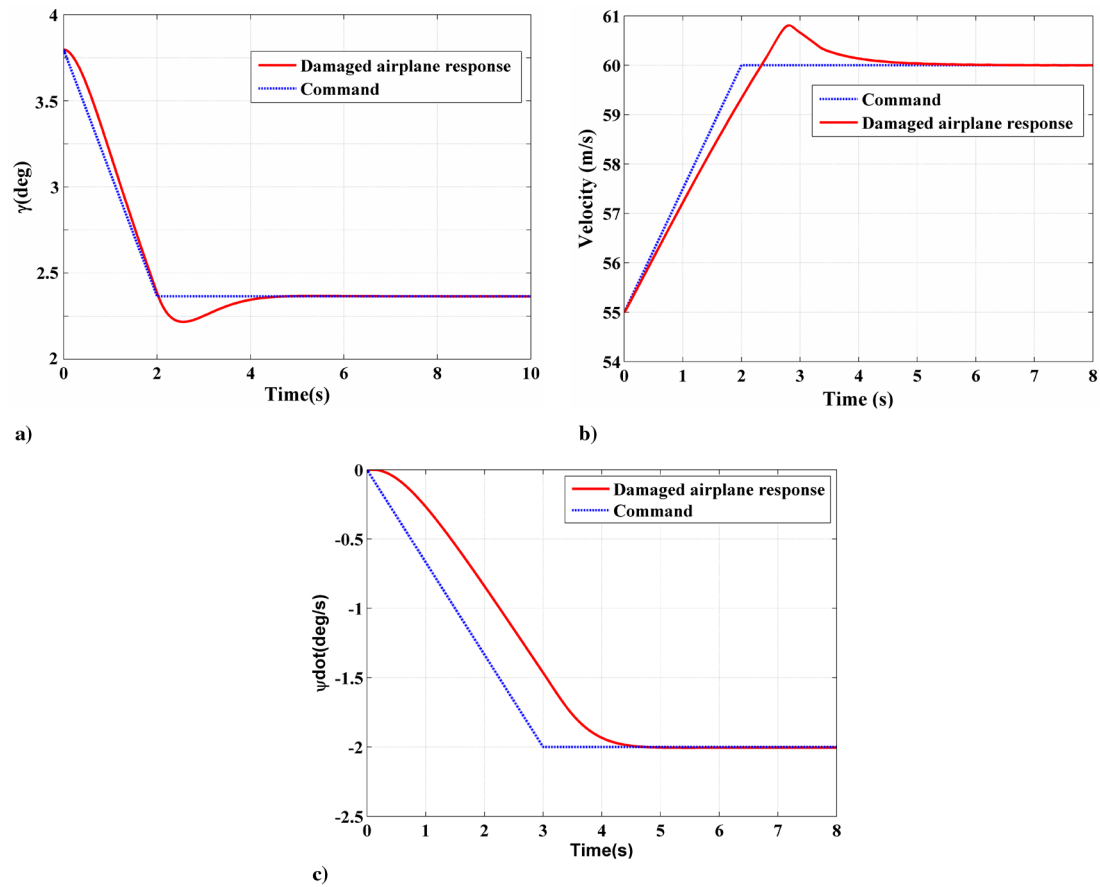


Fig. 15 Maneuver generation using an LQR controller.

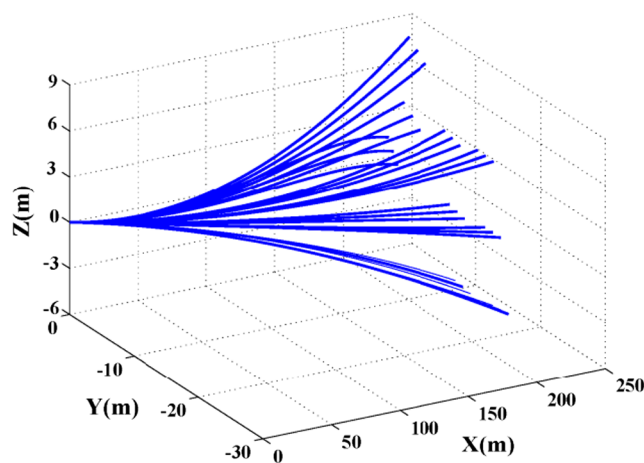


Fig. 16 Maneuver generation using an LQR controller.

given that longitudinal and lateral motion parameters for a damaged aircraft may be more tightly coupled. Therefore, although important, consideration of strong winds beyond levels for which a trajectory tracking controller can compensate without modification to the no-wind flight plan is beyond the scope of this paper and is discussed later as future work.

C. Motion Planning

The motion planning algorithm in this paper uses a potential field path planning algorithm to calculate a trajectory in real-time from a given initial trim condition to a desired landing site considering obstacle avoidance based on the motion library of prespecified trajectories. The data structure used in motion planning defines all segmented trim trajectories along with the D vector for each

maneuver inside the neighborhood of each trim condition. The goal state represents the desired landing runway position, heading, and flight path angle denoted by $(p_{x_g}, p_{y_g}, p_{z_g}, \psi_g, \gamma_g)$, respectively. Because the algorithm uses segmented trajectories with a pre-specified step size corresponding to a fixed Δt , the requirement of precisely reaching the goal state is relaxed by allowing the search algorithm to terminate once a state is found within a neighborhood of the goal state.

To find a motion plan in real time, we adopt a potential field planning strategy to our planning problem. The artificial potential field (APF) method [37] has been widely used as a computationally efficient path planning strategy for ground and aerial robotics applications for more than two decades. In APF planning, a local gradient is followed to guide a vehicle from an initial state to a goal state. An attractor potential function pulls the vehicle toward the goal state while repulsor potential functions push the vehicle away from obstacles. The vehicle follows the maximum local gradient path toward the goal. Although local minima are possible, such minima are improbable in sparse obstacle fields. For this work, possible steps in the gradient direction correspond to the 27 neighboring trajectory segments of the current trim state. In other words, at each step, the trajectory segment producing the minimum total potential is selected.

Combinatorial search offers an alternative to gradient-based planning algorithms such as APF. The tradeoff is that APF is computationally tractable, but combinatorial search can guarantee comprehensive search through the solution space and solution optimality. For example, the A* algorithm assigns a total cost function $f(n) = g(n) + h(n)$ for each node which is the sum of the cost from start state to the current node $g(n)$ and the heuristic $h(n)$ estimating the cost from the current node to the destination. As long as $h(n)$ underestimates actual cost-to-go, A* guarantees an optimal solution will be identified. The A* algorithm combines the approaches of Dijkstra's algorithm and the heuristic benefit from a best first (or greedy) search [38].

Dijkstra’s algorithm explores and expands all the existing neighbors of each node in each step. This yields a prohibitively large state space of order 27^N in our planning algorithm, where 27 is the number of neighboring states of each node, and N is the total number of trajectory segments in a desired flight plan. The A^* algorithm does not typically explore all the neighboring states in each step but it explores multiple paths by keeping all frontier nodes in a queue and expand the next best node based on the A^* cost function until it finds what it considers an optimal solution. Therefore, the A^* will typically still generate a state space that is quite large, often too large to support real-time implementation on an embedded (onboard) computing environment.

Several variants of A^* have been developed to reduce the computation effort of the search process. In weighted A^* (WA^*), the total cost function is estimated by $f(n) = g(n) + W^*h(n)$ where W is the weighting factor. The WA^* algorithm relaxes the optimality requirements of A^* by increasing the importance of the heuristic, which leads the algorithm to favor expansion of states closer to the goal [39]. In dynamically weighted A^* (DWA^*), the weighting factor is adjusted based upon the depth of the search, decreasing the importance of the heuristic as the goal is approached to encourage identification of a near-optimal or optimal solution [40]. Because we require a real-time algorithm for safe landing trajectory generation, we adopt the artificial potential field strategy in this work, in conjunction with the library of prespecified trajectories. The APF algorithm identifies a single path from an initial state to a goal state, which is the optimal cost (minimum gradient) node identified at each step. This reduces greatly the search space size to $27 \times N$, the number of steps along the solution trajectory. This algorithm can therefore be executed in real-time. In each step, the algorithm generates the candidate successor points inside the neighboring set and selects the next trajectory segment from among the neighboring trim conditions based on a cost function. In potential field planning, typically the cost function is computed as a sum of a goal-seeking attractor term and obstacle-avoiding repulsor terms. For this work, we use an optimal total cost function $f(n) = g(n) + W^*h(n)$ for both the potential field and A^* algorithms used for trajectory planning. This enables direct comparison of total costs when finding solutions with both methods and does not alter the potential field method because $g(n)$ will be the same for all possible next steps because only one path is explored to the goal. Cost terms $g(n)$ and $h(n)$ are defined next.

The aircraft kinematic states at each step are updated using the following equations:

$$\begin{aligned} \psi_{\text{mean}} &= \frac{\psi_{\text{previous}} + (\psi_{\text{previous}} + \Delta\psi)}{2} \\ \gamma_{\text{mean}} &= \frac{\gamma_{\text{previous}} + (\gamma_{\text{previous}} + \Delta\gamma)}{2} \\ p_x &= p_{x_{\text{previous}}} + \Delta s \cos(\gamma_{\text{mean}}) \cos(\psi_{\text{mean}}) \\ p_y &= p_{y_{\text{previous}}} + \Delta s \cos(\gamma_{\text{mean}}) \sin(\psi_{\text{mean}}) \\ p_z &= p_{z_{\text{previous}}} + \Delta s \sin(\gamma_{\text{mean}}) \end{aligned} \quad (27)$$

Because $\dot{\psi}$ and $\dot{\gamma}$ are fixed constants in each step of algorithm optimization, a mean value is defined for ψ_{mean} , γ_{mean} , and the airplane’s new position is derived.

Safety is prioritized through use of the SVI_T , terrain avoidance, landing velocity, flight time, and final airplane landing requirements (runway position, heading, glide slope). All the previously discussed criteria (SVI_T , etc.) are modeled as cost functions in the potential field planning algorithm to compute a safe landing trajectory in real time. The potential functions map to the heuristic cost-to-go function $h(n)$ used for combinatorial search methods such as A^* .

To specify the potential field planning problem, initial state, final state, possible “steps” (the neighboring trim state database), and total potential (the heuristic function) must all be defined. Intuitively, the goal is to sequence trim state segments that offer higher safety values, reduced flight time, and adequate clearance from terrain. Final landing state consists of runway approach end position P_L , runway heading ψ_L , final approach glide slope γ_L and airspeed V_L . A

heuristic function is presented next, which incorporates safety criteria defined previously as well the as the goal state, which specifies that final airplane position, heading, and flight path angle must be equal to the landing site position, heading, and glide slope. Airplane velocity (airspeed) must decrease while approaching the runway to approximately the stall speed because landing with high speeds increases risk during landing.

Next, we define both cost function terms $g(n)$ and $h(n)$. For simplicity, the cost function $g(n)$ is defined as the sum of all segment path lengths Δs from the initial to current state. The heuristic function $h(n)$ is defined next. In each step, the potential field algorithm identifies the best of the 27 possible choices of the next trim condition. The A^* algorithm expands the space starting from the initial state with a branching factor of 27 representing all possible neighboring trim states. Both search algorithms terminate when the goal state is reached to within an acceptable tolerance.

D. Heuristic Cost Definition

The total heuristic function is of the following form:

$$h(n) = \sum_{k \in \Omega} J_k \quad (28)$$

where $\Omega = \{SVI_T, P_L, \text{altitude, time, } \psi_L, \gamma_L, V_L, \text{terrain}\}$ represents the set of cost criteria defined previously. Each cost function term J_k must be weighted appropriately to balance the relative impact of each criterion on total cost. The next equations define each cost term and its weighting; specific weights for the presented case study are given in Table 1.

The cost term for SVI, specified previously in Eq. (12), is defined as

$$J_{SVI} = w_{SVI}(1 - SVI_T) \quad (29)$$

According to SVI_T analysis, for the left-wing-damaged airplane case study, increasing the SVI weighting factor w_{SVI} leads to trajectories with higher values of SVI_T .

The airplane final position must be equal to the runway position, and thus the P_L cost function is defined as the airplane distance to the desired runway position as follows:

$$J_{P_L} = w_{P_L} \sqrt{\Delta p_x^2 + \Delta p_y^2 + \Delta p_z^2} \quad (30)$$

where $(\Delta p_x, \Delta p_y, \Delta p_z)$ represents airplane distances to the landing site in inertial $x, y,$ and z directions, respectively.

Because fixed-time trajectory segments are sequenced into flight plans, increasing the distance-to-go weighting factor w_{P_L} leads to trajectory segments flying toward the landing runway with larger velocities; this term therefore reduces the total flight plan time as well as tending to reduce maneuvers between different trim conditions that deviate from a direct course to a short approach to landing. This paper proposes a “weighting schedule” based on distance from the landing runway and obstacles. A large w_{P_L} is defined while the airplane is far from the runway and the obstacles based on a user-defined threshold defining the notion of “far”. In our test cases, w_{P_L} has been demonstrated to effectively regulate the total time and the number of switches between different trim conditions in the total flight plan. When the aircraft is far from the runway, priority is given to flying fast and direct to the landing site. When the aircraft approaches landing or an obstacle, priority is given to safety and alignment with goal state constraints, resulting in frequent switching to neighboring trim states.

Table 1 Cost function weights for the two different flight phases

Distance from runway/obstacle	w_{SVI}	w_{P_L}	w_{Alt}	$w_{Terrain}$	w_λ	w_{ψ_L}	w_γ	w_{V_L}
Far	6	0.1	7	0	0	0	0	0
Near	4	0.05	1	7000	2	3	6	0.3

The alignment of the airplane final heading is achieved by the heading cost function presented next:

$$J_{\psi_L} = w_{\psi_L} |\psi_a - \psi_L| \quad (31)$$

The heading weight is set to a large value when the airplane position is classified as “near” the runway based on a user-defined threshold and is zero elsewhere.

A localizer can guide the aircraft toward the runway centerline, and a glide slope indicator can guide the final descent. Ideal approach parameters can be used as cost criteria for the landing flight plan; if the aircraft cannot achieve the target parameters, cost will be minimized by selecting trim states with parameters as close to the ideal values as possible. First, the localizer angle λ is defined according to Fig. 17 so as to guide the airplane toward the runway centerline:

$$J_{\lambda} = w_{\lambda} \tan^{-1} \left(\frac{|\Delta p_y|}{|\Delta p_x|} \right) \quad (32)$$

The cost term J_{γ} is then defined to drive the airplane flight path angle toward the desired glide slope as follows:

$$J_{\gamma} = w_{\gamma} |\gamma_a - \gamma_L| \quad (33)$$

where γ_a is the airplane flight path angle, and γ_L is the desired glide slope. The weight w_{γ} increases on approach to the runway and is adjusted to zero elsewhere. Increasing the weight w_{λ} guides the algorithm to choose segments that, along with the heading cost term, drive the aircraft toward runway heading and centerline. In fact, based on a threshold distance from the runway, the weight switches from zero to a large value.

Airplane landing occurs in a reference speed with no bank angle. Velocity cost is defined to penalize the high velocity landing approaches as follows:

$$J_{V_L} = w_{V_L} |V_a - V_L| \quad (34)$$

The velocity weight w_{V_L} assumes a substantial, nonzero value as the airplane approaches the landing site and switches from zero to a large value. Because all segmented trajectories saved in the example trim library are derived for zero sideslip ($\beta = 0$), segmented trajectories with zero heading rate ($\dot{\psi} = 0$) will lead to the zero bank angle necessary for the final phase of landing. Note that winds are not modeled in this work. In future work, a wind-aware path planner would need to properly account for execution of a final approach with nonzero sideslip, bank, or a combination to manage significant crosswind, and intermediate waypoints would need to be translated in accordance with wind-induced translation as each trim state is executed. A landing controller allowing touchdown and rollout from an approach/flare state with unconventional attitude and/or high speed will also be required.

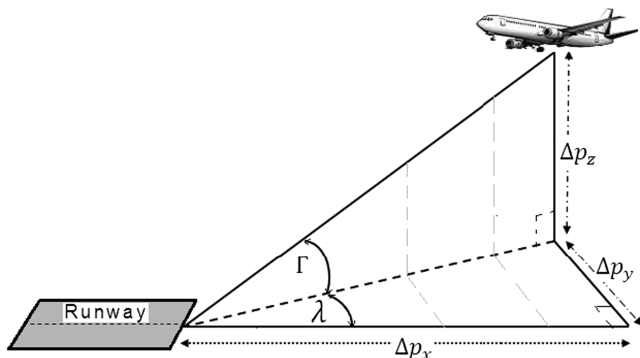


Fig. 17 Airplane position relative to the runway.

In this paper, the final touchdown position and heading define the desired target point (i.e., final state). The localizer, glide slope, landing velocity, and heading cost function terms have been defined to bias the flight planner to choose feasible trim conditions that satisfy standard approach and touchdown procedures to the extent possible. The algorithm presented in this paper could instead use a final approach fix (FAF) as its final state, enabling use of the instrument landing system (ILS) and potentially simplifying the cost function. However, planning only to the FAF would require the assumption that the damaged aircraft could in fact follow a standard final approach to landing. In severe failure or damage cases, the contracted flight envelope may not allow a standard final approach, requiring construction of an emergency landing trajectory that respects degraded performance constraints to the touchdown point. Although standard use of the ILS may not be possible with a nonstandard approach, GPS and other navigation sensors will remain available, and top priority must be placed on maintaining a feasible flight condition given the failure or damage scenario.

In this work, it is assumed that a terrain map is loaded onboard, and therefore all the obstacles and terrain in the region around the landing site are known. Per a standard potential field model [37], proximal terrain is penalized exponentially to reduce the risk of collision:

$$J_{\text{Terrain}} = w_{\text{Terrain}} \exp\left(\frac{1}{d_T}\right) \quad (35)$$

where d_T is the distance to the terrain and is calculated as

$$d_T = \sqrt{(p_x - x_T)^2 + (p_y - y_T)^2 + (p_z - z_T)^2}$$

where (p_x, p_y, p_z) and (x_T, y_T, z_T) are airplane and terrain inertial position, respectively. Terrain weighting factor w_{Terrain} increases linearly as altitude above ground level (AGL) decreases. Per flight envelope analysis, the airplane has a constraint on maximum negative flight path angle (γ_{\min}). Angle Γ is defined according to Fig. 17 as follows:

$$\Gamma = \tan^{-1} \left(\frac{\Delta p_z}{\sqrt{\Delta p_x^2 + \Delta p_y^2}} \right) \quad (36)$$

Γ represents the angle relative to horizontal of a vector direct to the landing site. Let γ_{\min} represent the minimum (maximum negative) achievable path angle in the damaged airplane maneuvering flight envelope. Note that we presume $\gamma_{\min} < 0$; otherwise, the plane could never descend to land. We then define an altitude cost function based on the angle Γ as follows:

$$J_{\text{Alt.}} = \begin{cases} w_{\text{Alt}}(\Gamma - |\gamma_{\min}|) & = w_{\text{Alt}}(\Gamma + \gamma_{\min}), & \Gamma > |\gamma_{\min}| \\ 0 & & \text{otherwise} \end{cases} \quad (37)$$

This term encourages the flight planner to reduce the altitude as quickly as possible either in turning flight or flight away from the landing site until the aircraft is able to descend direct to the airport.

E. Terrain Modeling

Different methods of terrain modeling exist based on a 3-D surface representation such as a trigonometric function, Bezier surface, and digital elevation map representation. In the current study, a trigonometric function terrain model is used to ensure continuous function descriptions of terrain, as given next [41]:

$$\begin{aligned} \text{elev}(x, y) &= \sin(y + a) + b \cdot \sin(x) + c \cdot \cos\left(d \cdot \sqrt{x^2 + y^2}\right) \\ &+ e \cdot \cos(y) + f \cdot \sin\left(f \cdot \sqrt{x^2 + y^2}\right) + g \cdot \cos(x) \\ a &= -1.5, \quad b = 3, \quad c = -5, \quad d = 0.6, \\ e &= 0.1, \quad f = -1, \quad g = -1 \\ -20 < x < 10, \quad -10 < y < 20 \end{aligned} \quad (38)$$

F. Simulation Results

Numerical simulations were performed to demonstrate the performance of the proposed algorithm. Several modifications were considered in the proposed algorithm to improve the performance of the trajectory planner with focus on evaluating execution time. Trajectory segments with double step size (Δt of 10 s) were used while the airplane is far from the runway and any terrain obstacle; these segments have direct correspondence to the standard trajectory segments found in the standard trim state library. In addition, high weights for the cost function term J_{P_t} reduce the algorithm computation time as well as total flight time because the number of segments to generate can be significantly reduced. Removal of segmented trajectories with positive path angle ($\gamma > 0$) from the feasible neighborhood list of trim conditions while the airplane is far from the terrain also assisted in minimizing solution computation time and path length.

Two different landing scenarios are considered for the 33% left-wing-damaged airplane [14]. In scenario a, the damaged airplane is initially flying in the opposite direction ($\psi_a = 0$) of the runway ($\psi_{\text{runway}} = 180$ deg). The locations of the airplane and the airport are set at (11, 000, -5000, 2500 m) and (0, 0, 0), respectively. Figure 18a shows a three-dimensional sample trajectory generated by the potential field algorithm in which the airplane descends in a left-turning spiral trajectory using 10 s trajectory segments. Once the desired altitude is reached based on the maximum negative path

angle, the algorithm minimizes angle λ to align the airplane with respect to the runway. The final approach part of the trajectory satisfies the heading and position criteria using a smaller step size of 5 s. Figures 18b and 18c illustrate the top view and front view of the designed trajectory. The total flight trajectory length and time for the identified solution are 22,465 m and 630 s, respectively.

The schedule of cost function weights used for presented case studies is provided in Table 1. As previously discussed, two weight sets are used: one for trim segments far from the runway or obstacles, and one for trim segments considered near the landing runway or an obstacle. Trim segments with distance greater than 6 km to the landing runway and more than 3 km from an obstacle are classified as far; all closer segments are classified as near. Weights in the first three columns (w_{SVI} , w_{P_L} , and w_{Alt}) become larger when the airplane is far from the landing site or an obstacle. The five last columns ($w_{Terrain}$, w_λ , w_{ψ_L} , w_γ , and w_{V_L}) represent parameters relevant to runway approach (or obstacle avoidance) and thus are only active when the airplane is near the runway or an obstacle.

The performance and shape of the trajectory depend significantly on the selected set of cost function weights. The weights specify the relative importance of each cost at each decision step thus guide the algorithm to decide which segmented trajectory to choose among the 27 possible neighboring trim conditions. To illustrate the impact of poorly chosen cost function weights, we simulated a landing scenario in which the altitude weight w_{Alt} is not sufficiently large to encourage the aircraft to descend far from the runway. As shown in Fig. 19 (scenario a'), if w_{Alt} is too small, the helical part of the trajectory can occur in the final phase of the flight near the runway at a lower altitude just before final approach to landing. Although this solution is feasible, given flight envelope constraints, it is less desirable than a solution with an initial turning descent followed by a longer segmented approach. As discussed previously, a standard final approach path would be guaranteed only by construction of the flight plan only to a final approach fix; however, such a strategy will only work in cases where a standard final approach can be safely executed given the damage or failure scenario.

Scenario b again uses a 33% left-wing damage case, with an initial aircraft state at (9000, 9000, 2500 m) with the same opposite heading

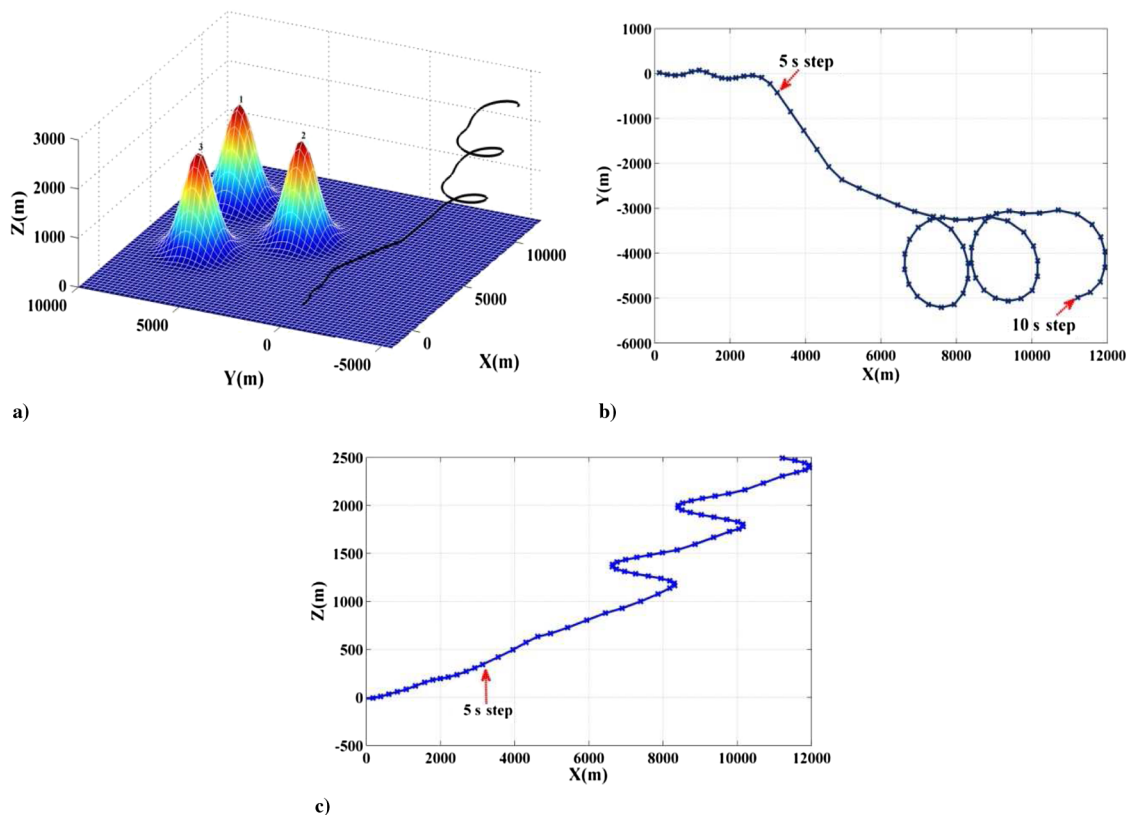


Fig. 18 Landing trajectory in scenario a: a) 3-D, b) top view, and c) side view.

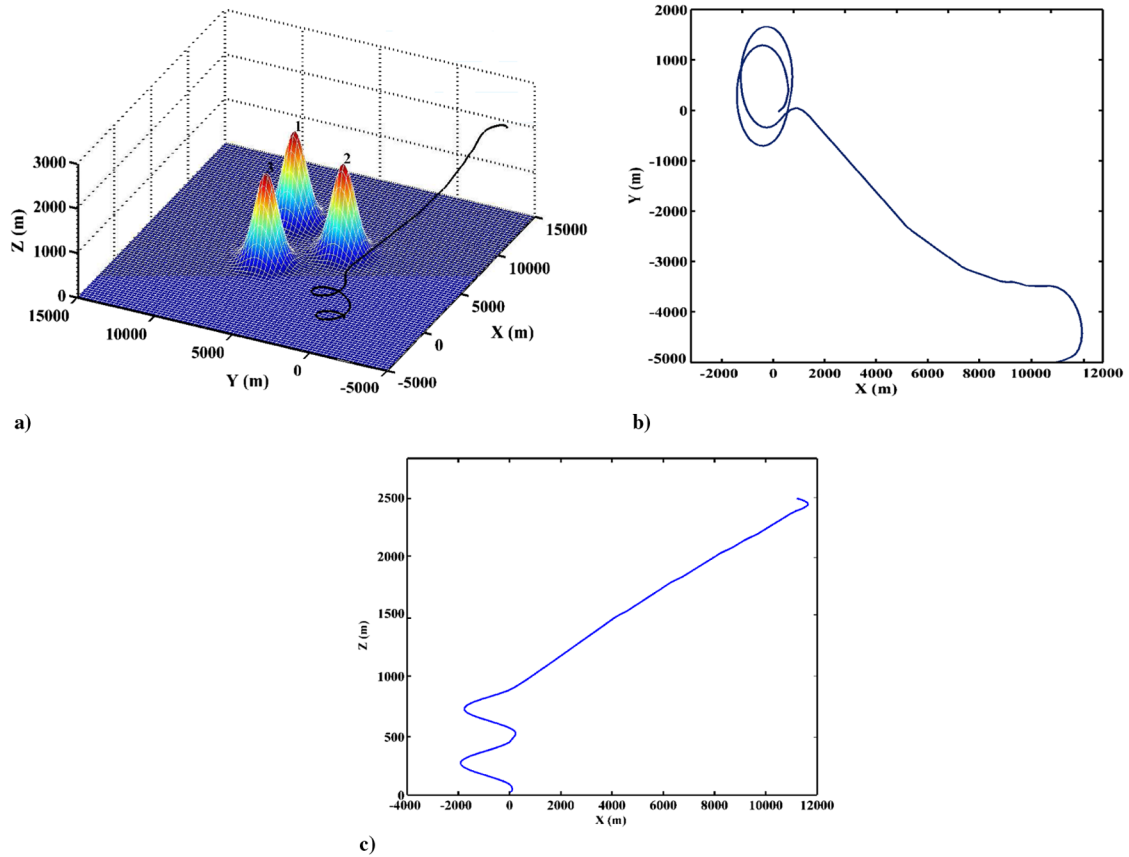


Fig. 19 Landing trajectory in scenario *a'*: a) 3-D, b) top view, and c) side view.

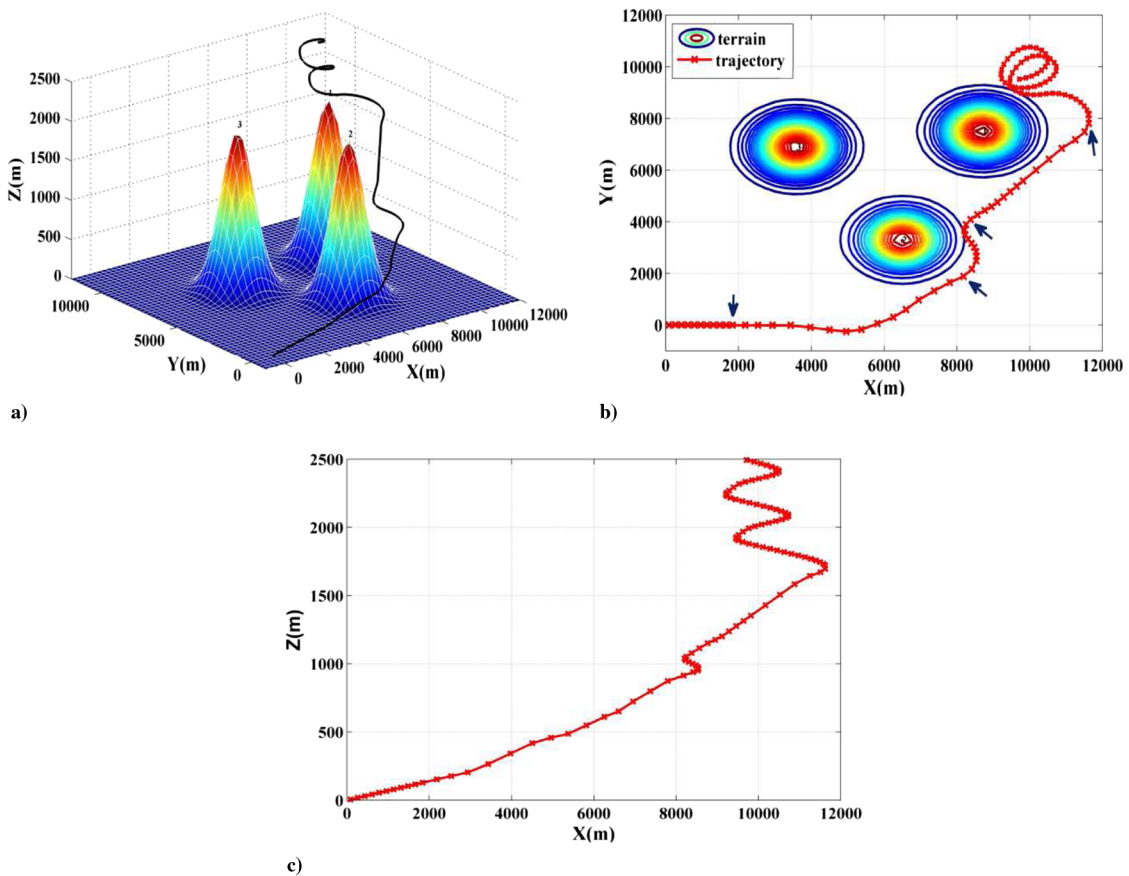


Fig. 20 Landing trajectory in scenario *b*: a) 3-D, b) top view, and c) side view.

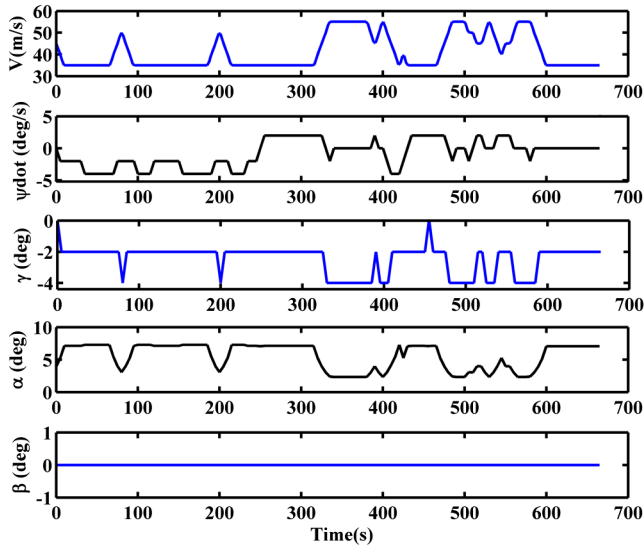


Fig. 21 Variation of flight states in flight vs time.

condition as in scenario a. Multiple obstacles were present in the vicinity of the initial state. As shown in Fig. 20, the airplane reduces altitude above the terrain height in the left-turning spiral and then changes direction to avoid the obstacle while trying to minimize λ to drive the airplane toward the runway. In the final approach phase of the flight plan, the airplane continues its descent with the airplane heading aligned with the runway. Figures 20b and 20c show the top and front views of the trajectory. Obstacles are depicted by contours in Fig. 20b while the step size variations are indicated with arrows. The total flight trajectory length and time for the identified solution are 24,675 m and 670 s, respectively.

The potential field planner was implemented in MATLAB on a Pentium 4 system with 3.4 GHz CPU. We ran MATLAB in the standard interpreted code mode; embedded implementations will be significantly faster. The number of expanded nodes for scenario b is only 128 while using 10 s segments as described previously and is 149 while using only the 5 s segments. These cases require 0.1 and 0.12 s to run in MATLAB, respectively. These results also represent the number of trajectory segments sequenced into the solution path. Note that the number of expanded nodes depends on the distance to the airport and terrain complexity.

Figures 21–23 illustrate the flight variables of scenario b. Figure 21 illustrates solutions obtained over variations of flight states $[V, \psi, \gamma, \alpha, \beta]$, which demonstrate the algorithm's ability to identify solutions. Solutions were identified in all examined cases. In each

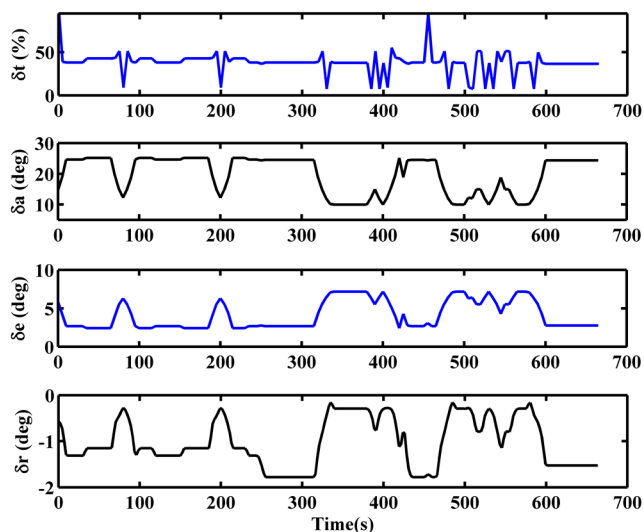


Fig. 22 Control inputs vs time.

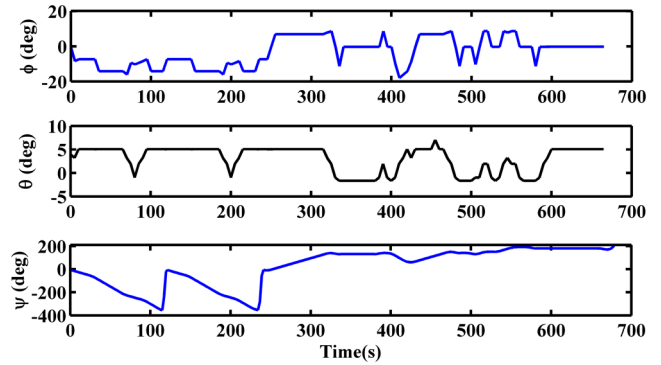


Fig. 23 Kinematic angles vs time.

case, the damaged airplane velocity is reduced to near the ideal landing velocity while maintaining a fixed heading with no sideslip angle ($\beta = 0$). The landing path angle is adjusted to the airport glide slope ($\gamma = -2$ deg). Figure 22 shows control inputs $[\delta_r, \delta_a, \delta_e, \delta_t]$ required for maintaining trim conditions and maneuvers during the flight in scenario b [42%, 25 deg, 3.2 deg, -1.6 deg], respectively. Aircraft attitude in scenario b is illustrated in Fig. 23. It is noted that the roll angle is zero in touchdown, and the airplane heading evolves as expected from the initial value to the runway heading ($\psi_{\text{runway}} = 180$ deg).

A wide range of scenarios with different initial/final positions, headings, and number of obstacles have been investigated. More than 3000 scenarios with different initial states (initial velocity, flight path angle, heading rate, and altitude) and different relative distance range from 50 to 2 km in obstacle fields of 3, 6, and 9 obstacles were randomly generated. A suite of different initial and final heading angles in the range $[0 \ 2\pi]$ have been investigated. The algorithm successfully reaches the goal state with a sufficiently small number of expanded nodes to be considered a viable real-time algorithm. An example of random initial heading scenarios with initial distance to the runway of about 14 km in an obstacle field of size 3 is illustrated in Fig. 24.

The performance of the potential field planning algorithm was also compared with A^* and WA^* algorithms with different weighting coefficients in more than 300 scenarios with initial states in the range of 46 km to the landing runway, different relative heading angles, and a known obstacle field of size 6.

The performance of each algorithm was characterized by the solution cost, the number of expanded nodes, the solution depth, the mean SVI in the total trajectory, and the number of heading changes. Three different weights of 1.5, 2, and 3 were used for WA^* algorithm. The mean values of the performance criteria in different scenarios introduced previously are presented in Table 2. As shown in Table 2, the potential field planning strategy expands far fewer nodes than A^*

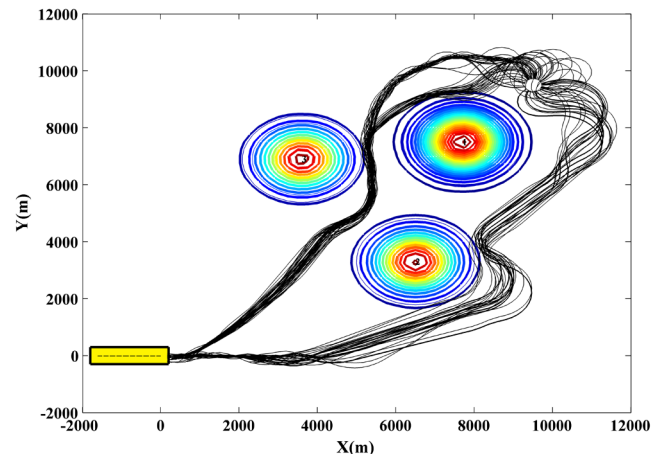
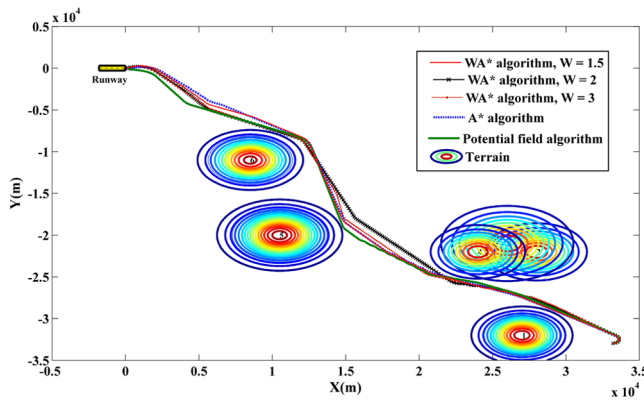


Fig. 24 Landing trajectories with different random initial heading angles (top view).

Table 2 Algorithm performance criteria comparison

Six obstacle, more than 300 scenario	Nodes	Execution time, s	Cost ($g(n)$)	Depth	SVI	Heading change
A*	14,358	31.7	49,214	208	0.73	24
WA*, $W = 15$	4,869	7.9	49,582	211	0.72	24
WA, $W = 2$	2,987	2.8	49,963	212	0.71	26
WA*, $W = 3$	2,874	2.6	50,096	212	0.72	26
Potential field planning algorithm	218	0.12	50,460	221	0.68	31

**Fig. 25** Comparison of landing trajectories generated by potential field, A*, and WA* algorithms in presence of six obstacles (top view).

and WA*. Comparison of execution time required by the different algorithms using a Matlab implementation on a standard PC demonstrates that the potential field algorithm is a viable as a real-time algorithm, generating solutions in well under a second for presented cases. Although one could expect an optimized, compiled implementation of all the search algorithms to yield substantially smaller execution times than those shown in Table 2, it is important to also recognize that the case studies presented in this work search over a substantially smaller space of possible landing trajectories than would be typical given a worst-case scenario (e.g., with a much larger distance from initial state to the landing runway). Ability to rapidly generate a solution in real time is critical for emergency situations requiring real-time safe landing solutions.

Other performance criteria (cost, SVI, and heading change) demonstrate the lower cost (optimality) of solutions identified with the A* algorithm. Results also demonstrate that increasing the weighting factor which drives the WA* algorithm toward a greedy search strategy that expands fewer nodes at the expense of higher solution costs. An example set of landing trajectories generated by the three algorithms is illustrated in Fig. 25.

IV. Conclusions

This paper has presented a discrete search optimization strategy for trajectory generation in the presence of terrain to enable a damaged aircraft to plan a safe landing trajectory. A database of trajectory segments is generated by the designed controller. Each segment has a safety value index (SVI_T) based on the stability and handling quality of that path with respect to the damaged airplane flight envelope. A potential field planning strategy is used to obtain trajectories from the trajectory library based on criteria defined for safe landing. In addition, safety is considered for the landing trajectory in terms of touchdown velocity, heading, bank angle, and glide slope. At each search step, the segmented trajectory with minimum cost is chosen. The trajectory shape highly depends on the cost function weighting factors. Weights are balanced based on several factors including the airplane location with respect to the runway. Numerical simulations demonstrate that the proposed algorithm has the capability to rapidly generate safe landing trajectories in the presence of obstacles with a significant reduction in computational effort and minimal increase in solution cost when compared to A* and weighted A* algorithms.

The trajectory planner presented in this paper presumes no wind is present. Although mild, variable winds might be handled with a

trajectory-tracking control law, plans that presume no wind will be impossible to follow in moderate to high actual wind conditions. Future work is required to consider winds within the trajectory planner. This might be accomplished through real-time computation of wind drift into each progressive search-space waypoint, a computationally expensive but memory-efficient option, or through comprehensive modeling of steady wind directions and magnitudes within the trim state database, an option that would substantially increase database size.

References

- [1] Jordan, T. L., and Bailey, R. M., "NASA Langley's AirSTAR Testbed—A Subscale Flight Test Capability for Flight Dynamics and Control System Experiments," *AIAA Guidance, Navigation and Control Conference and Exhibit*, AIAA Paper 2008-6660, Aug. 2008. doi:10.2514/6.2008-6660
- [2] Shah, G., "Aerodynamic Effects and Modeling of Damage to Transport Aircraft," *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, AIAA Paper 2008-6203, Aug. 2008. doi:10.2514/6.2008-6203
- [3] Nguyen, N. T., and Krishnakumar, K., "Hybrid Intelligent Flight Control with Adaptive Learning Parameter Estimation," *Journal of Aerospace Computing, Information and Communication*, Vol. 6, No. 3, 2009, pp. 171–186. doi:10.2514/1.35929
- [4] Laban, M., "Online Aircraft Aerodynamic Model Identification," Ph.D. Thesis, Delft Univ. of Technology, Delft, The Netherlands, 1994.
- [5] Lombaerts, T., "Fault Tolerant Flight Control, A Physical Model Approach," Ph.D. Thesis, Delft Univ. of Technology, Faculty of Aerospace Engineering, Delft, The Netherlands, 2010.
- [6] Meuleau, N., Plaunt, C., Smith, D. E., and Smith, T., "A Comparison of Risk Sensitive Path Planning Methods for Aircraft Emergency Landing," *ICAPS-09: Proceedings of the Workshop on Bridging the Gap Between Task and Motion Planning*, 2009, pp. 71–80.
- [7] Betts, J., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207. doi:10.2514/2.4231
- [8] LaValle, S., *Planning Algorithms*, Cambridge Univ. Press, Cambridge, England, U.K., 2006, pp. 228–236, Chap. 5.
- [9] Yang, H., and Zhao, Y., "Trajectory Planning for Autonomous Aerospace Vehicles Amid Known Obstacles and Conflicts," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 6, 2004, pp. 997–1008. doi:10.2514/1.12514
- [10] Strube, M., "Post-Failure Trajectory Planning from Feasible Trim State Sequences," M.S. Thesis, Dept. of Aerospace Engineering, Univ. of Maryland, College Park, MD, 2005.
- [11] Atkins, E., "Dynamic Waypoint Generation Given Reduced Flight Performance," *42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2004-0779, Jan. 2004. doi:10.2514/6.2004-779
- [12] 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. doi:10.2514/1.18816
- [13] Tang, Y., Atkins, E. M., and Sanner, R. M., "Emergency Flight Planning for a Generalized Transport Model Aircraft with Wing Damage," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2007-6873, Aug. 2007.
- [14] Yi, G., and Atkins, E., "Trim State Discovery for an Adaptive Flight Planner," *48th AIAA Aerospace Science Meeting*, AIAA Paper 2010-0416, Jan. 2010. doi:10.2514/6.2010-416
- [15] Tong, P., "Application of Genetic Algorithm to a Forced Landing Maneuver on Transfer of Training Analysis," Ph.D. Dissertation, Royal

- Melbourne Inst. of Technology, School of Aerospace, Mechanical and Manufacturing Engineering, Melbourne, Australia, April 2006.
- [16] Sarigul-Klijn, N., Rapetti, R., Jordan, A., Lopez, I., Sarigul-Klijn, M., and Nespeca, P., "Intelligent Flight Trajectory Generation to Maximize Safe Outcome Probability After a Distress Event," *Journal of Aircraft*, Vol. 47, No. 1, 2010, pp. 255–267.
doi:10.2514/1.45264
- [17] Lopez, I., and Sarigul-Klijn, N., "Aggregating Imprecise Information in Distressed Aircraft Path Planning," *AIAA Infotech@Aerospace*, AIAA Paper 2010-3379, April 2010.
doi:10.2514/6.2010-3379
- [18] Zhao, Y., "Efficient and Robust Aircraft Landing Trajectory Optimization," Ph.D. Dissertation, Georgia Inst. of Technology, Atlanta, 2012.
- [19] Frazzoli, E., Dahleh, M. A., and Feron, E., "Real-Time Motion Planning for Agile Autonomous Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 116–129.
doi:10.2514/2.4856
- [20] Bottaso, C. L., Leonelle, D., and Savani, B., "Path Planning for Autonomous Vehicle by Trajectory Smoothing Using Motion Primitives," *IEEE Transactions on Control Systems Technology*, Vol. 16, No. 6, 2008, pp. 1152–1168.
doi:10.1109/TCST.2008.917870
- [21] Hwangbo, M., Kuffner, J., and Kanade, T., "Efficient Two-Phase 3D Motion Planning for Small Fixed-Wing UAVs," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, IEEE Publ., Piscataway, NJ, April 2007, pp. 1035–1041.
- [22] Neas, B. C., and Farhood, M., "A Hybrid Architecture for Maneuver-Based Motion Planning and Control of Agile Vehicles," *Proceedings of the 18th IFAC World Congress*, Milano, Italy, Aug.–Sept. 2011, pp. 3521–3526.
doi:10.3182/20110828-6-IT-1002.03805
- [23] Asadi, D., Sabzehparvar, M., and Talebi, H. A., "Damaged Airplane Flight Envelope and Stability Evaluation," *Aircraft Engineering and Aerospace Technology*, Vol. 85, No. 3, 2013, pp. 186–198.
doi:10.1108/00022661311313623
- [24] Bacon, B., and Gregory, I., "General Equations of Motion for a Damaged Asymmetric Aircraft," *AIAA Atmospheric Flight Mechanics Conference and Exhibition*, AIAA Paper 2007-6306, Aug. 2007.
doi:10.2514/6.2007-6306
- [25] Ouellette, J., Raghavan, B., Patil, M. J., and Kapania, R. K., "Flight Dynamics and Structural Load Distribution for a Damaged Aircraft," *AIAA Atmospheric Flight Mechanics Conference*, AIAA Paper 2009-6153, Aug. 2009.
doi:10.2514/6.2009-6153
- [26] Lombaerts, T., Huisman, H., Chu, P., and Mulder, J. A., "Nonlinear Reconfiguring Flight Control Based on Online Physical Model Identification," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, May–June 2009, pp. 727–748.
doi:10.2514/1.40788
- [27] Chakraborty, A., Seiler, P., and Balas, G. J., "Nonlinear Region of Attraction Analysis for Flight Control Verification and Validation," *Journal of Control Engineering Practice*, Vol. 19, No. 4, April 2011, pp. 335–345.
doi:10.1016/j.conengprac.2010.12.001
- [28] Stevens, B. L., and Lewis, F. L., *Aircraft Control and Simulation*, 3rd ed., Wiley, New York, 2003, pp. 118–128, Chap. 2.
- [29] Khalil, H., *Nonlinear Systems*, 3rd ed., Prentice-Hall, Upper Saddle River, NJ, 2002, pp. 41–44, Chap. 2.
- [30] Goman, M. G., Kharmatovsky, A. V., and Kolesnikov, E. N., "Evaluation of Aircraft Performance and Maneuverability by Computation of Attainable Equilibrium Sets," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 2, March–April 2008, pp. 329–339.
doi:10.2514/1.29336
- [31] Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls, Part 1*, Design, Analysis and Research Corporation (DARcorporation), Lawrence, KS, 1998, pp. 417–430, Chap. 6.
- [32] Tandale, M. D., Bowers, R., and Valasek, J., "Trajectory Tracking Controller for Vision-Based Probe and Drogue Autonomous Aerial Refueling," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 4, 2006, pp. 846–857.
doi:10.2514/1.19694
- [33] Kaminer, I., Pascoal, A., Hallberg, E., and Silvestre, C., "Trajectory Tracking for Autonomous Vehicles: An Integrated Approach to Guidance and Control," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, 1998, pp. 29–38.
doi:10.2514/2.4229
- [34] Kluever, C., "Unpowered Approach and Landing Guidance Using Trajectory Planning," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 6, Nov.–Dec. 2004, pp. 967–974.
doi:10.2514/1.7877
- [35] Bares, P., Lazarus, S., Tsourdos, A., and Savvaris, A., "Adaptive Guidance for UAV Based on Dubins Path," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2013-5181, Aug. 2013.
- [36] Techy, L., and Woolsey, C., "Minimum-Time Path Planning for Unmanned Aerial Vehicles in Steady Uniform Winds," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 6, Nov.–Dec. 2009, pp. 1736–1746.
doi:10.2514/1.44580
- [37] Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *International Journal of Robotics Research*, Vol. 5, No. 1, 1986, pp. 90–98.
doi:10.1177/027836498600500106
- [38] Russel, S., and Norvig, P., *Artificial Intelligence: A Modern Approach*, 3rd ed., Prentice-Hall, Upper Saddle River, NJ, 2009, pp. 92–99, Chap. 3.
- [39] Pohl, I., "Heuristic Search Viewed as Path Finding in a Graph," *Artificial Intelligence*, Vol. 1, Nos. 3–4, 1970, pp. 193–204.
doi:10.1016/0004-3702(70)90007-X
- [40] Ebdndt, R., and Drechsler, R., "Weighted A* Search—Unifying View and Application," *Artificial Intelligence*, Vol. 173, No. 14, 2009, pp. 1310–1342.
doi:10.1016/j.artint.2009.06.004
- [41] Pehlivanoglu, Y. V., Baysal, O., and Hacioglu, A., "Path Planning for Autonomous UAV via Vibrational Genetic Algorithm," *Aircraft Engineering and Aerospace Technology*, Vol. 79, No. 4, 2007, pp. 352–359.
doi:10.1108/00022660710758222