AIAA Guidance, Navigation and Control Conference and Exhibit
20 - 23 August 2007, Hilton Head, South Carolina

AIAA 2007-6873

EMERGENCY FLIGHT PLANNING FOR A GENERALIZED TRANSPORT AIRCRAFT WITH LEFT WING DAMAGE

Yunshen Tang*, Ella M. Atkins†, and Robert M. Sanner‡

Flight management systems are highly capable in nominal conditions but are unable to manage most emergency situations, particularly when the performance envelope is degraded due to damage or component failures. Emerging adaptive control and system identification technologies can maintain stable flight given reduced performance, but the pilot must assume the responsibility of guiding the disabled aircraft to a safe landing. With a highly restricted envelope, the family of feasible trajectories may be so unintuitive that a pilot may not be capable of identifying a safe landing flight plan given current high-level automation aids. This paper presents an adaptive flight planner to build landing trajectories for disabled aircraft. A case study is investigated in which a Generalized Transport Model aircraft must land following the loss of a significant fraction of its left wing. Trimmed (non-accelerating) flight conditions define the post-damage flight envelope. Nearby landing runways are prioritized and segmented trajectories to the top-priority sites are defined by trim state and transition sequences. An LQR-based PID controller enables the damaged GTM aircraft to correctly track trajectory commands over trimmed flight and transition segments. A suite of initial conditions are presented to evaluate flight planner performance.

I. Introduction

Modern aviation is a safe and reliable form of transportation. In 1971, fatal accidents on commercial jetliners occurred approximately once in every 140 million miles flown. Thirty years later, jets fly 1.4 billion miles for every fatal accident. Although a ten-fold safety improvement has been made, commercial aviation accidents still occur.[1] In-flight aircraft damage and failures cause a nontrivial fraction of remaining commercial aviation accidents. Significant damage/failures that impact actuators or the airframe reduce aircraft performance, presenting several challenges to the pilot. First, the pilot must understand the reduced performance envelope sufficiently to maintain long-term control, a problem addressed by adaptive control and system identification research. Additionally, with significant or progressive degradation, the pilot must rapidly select a landing site and plan a feasible landing trajectory to that site. This latter problem is less studied in the research community and is the focus of this paper.

To cope with in-flight emergencies, researchers have begun to design flight management architectures that can help the pilots with decision-making during emergencies.[2,3,4,5] Human factors experiments show that an Emergency Flight Planner[3] may be a useful tool to prevent pilots from performing unsafe tasks when emergencies occur, although post-failure flight planning algorithms were generally not emphasized in this work. There has been a growing interest in designing autonomous flight control systems for Unmanned Aerial Vehicles (UAVs). For example, Boskovic et al[6] have devised a multi-layer autonomous control architecture for UAVs and have described its operation during failures, yet analysis has focused on adaptive control technology rather than reduced-performance flight planning. An adaptive flight planner (AFP) with post-failure flight planning capability was previously proposed for a loss of thrust emergency,[7] with subsequent efforts concentrating on reduced-performance trajectory planning for control surface failures.[8,9] We extend this AFP and examine its operation in a scenario where structural damage severely impacts flight performance.

To handle an aircraft with degraded performance, Flight Management Systems (FMS) must be augmented with: (i) an adaptive/reconfigurable controller, (ii) a system identification module to characterize reduced performance for the flight management system and for a pilot, and (iii) an adaptive or emergency flight planner capable of functioning over a highly-constrained set of possible flight states. Significant emphasis has been placed on systems to stabilize aircraft with unknown or degraded performance through robust inner and outer loop feedback control.[10-17] Solutions are typically categorized as adaptive control,[10,12,14] fault-tolerant and reconfigurable control,[11,13,15] or intelligent flight control.[16,17] When emergencies occur, it is also crucial for the pilot-FMS system to guide the aircraft through a feasible landing flight plan. Nominally, any waypoints within range of the aircraft are reachable,

[*] Graduate Research Assistant, Aerospace Engineering Dept., University of Maryland.
[†] Associate Professor, Aerospace Engineering Dept., University of Michigan, Associate Fellow.
[‡] Associate Professor, Aerospace Engineering Dept., University of Maryland, Senior Member.

so such waypoints based on destination, air traffic constraints, or targets of opportunity (for military missions) are typically defined by a pilot/remote operator flying manually or through the flight management system. Given nominal performance and no need for evasion, a specified sequence of 4-D waypoints (*x,y,h,t*) is connected with direct routes augmented by heading adjustments (turns) when required. With a degraded flight envelope, however, nominal waypoints may not be reachable, and the default "turn-and-fly" strategy may not be possible. For real-time path planning, a variety of motion-planning techniques have been devised,[18] with some applications to aircraft.[6] Other techniques applied to Aerospace vehicles include probabilistic-map-based methods,[19] Voronoi diagrams[20] for cluttered environments, evolutionary path planning,[21,22] and derivatives of the A* algorithm for applications such as waypoint ordering with obstacle avoidance.[23] While the trajectories generated by these approaches are appropriate for autonomous UAVs given performance constraints are met, commercial and general aviation pilots prefer flight plans specified as a sequence of constant-trim flights and transitions between these trim states because *segmented* routes can be comprehended intuitively by human pilots and air traffic controllers.[24] An example of this concept applied to trajectory planning is the maneuver automaton.[25] This work applies a similar approach to segmented trajectory planning through the use of hybrid trim-trajectory motion primitives, extending previous work[9] that focused on control surface jam failures in an F-16 aircraft.

This work is performed in the context of the Emergency Flight Management Architecture illustrated in Figure 1. The Adaptive Flight Planner (AFP) in this architecture is designed to maintain aircraft safety in the presence of failures and/or damage. The Flight Plan Monitor propagates the existing flight plan through the post failure/damage performance model to verify feasibility of the flight plan. If the executing flight plan is infeasible, the pilot is notified via the Pilot Interface. Concurrently, the AFP is activated to generate a new flight plan. Within the AFP, a Landing Site Search (LSS) module identifies a safe landing site, currently defined as a runway deemed safe based on the degraded aircraft performance model. The Segmented Trajectory Planner then constructs a dynamically-feasible trajectory to the landing site.

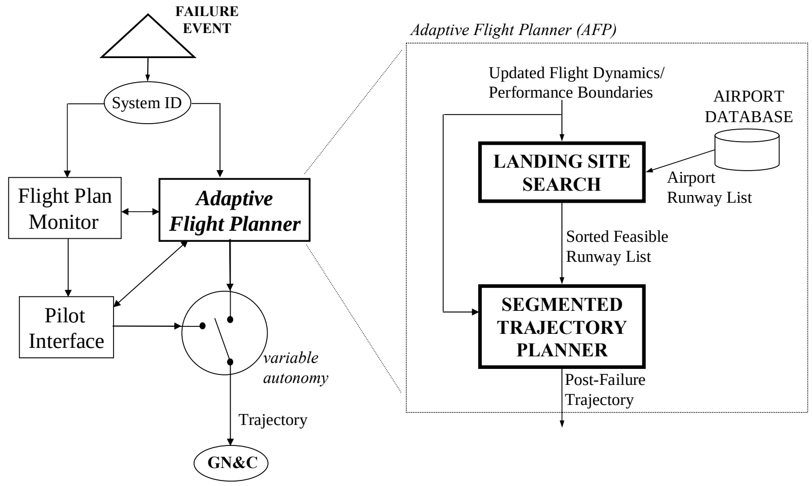


Figure 1: Emergency Flight Management Architecture[7]

This work integrates previously-developed landing site search[7] and path planning methods[9] into the end-to-end Adaptive Flight Planner (AFP), as illustrated in Figure 1. Stability derivative data for a Generalized Transport Model (GTM) aircraft with substantial left wing damage is used as the reference model in this work. Use of the GTM data requires offline characterization of trim states and simulation-based validation of planned flight trajectories. The severity of the damage also requires careful control law tuning to minimize transients and to maintain stability in the presence of disturbances. Efficient algorithms are used to identify and rank feasible landing runways based on the post-failure/damage aircraft performance limits. This work also augments an existing trim sequence trajectory planner to construct a complete trajectory leading the aircraft from the initial location at which the emergency occurs to the desired landing runway, by a sequence of trim state segments and the maneuver transitions between neighboring trim states.

II. GTM Aircraft Model and Trim Definition

The standard 6-DOF equations of motion for a rigid-body aircraft are used to describe the Generalized Transport Model (GTM) utilized in this work. The equations of motion assume a flat Earth, since accelerations associated with the Earth's rotation are negligible compared to those of a maneuvering aircraft. State variables include the components (*U, V, W*) of translational velocity vector v_B, angular rates (*p, q, r*), inertial position (*x, y, z*), and ZYX Euler angles (ϕ, θ, ψ).[26,27] For convenience, state variables (*U, V, W*) are replaced by angle of attack α, sideslip angle β, and true airspeed V_T, the primary factors determining aerodynamic forces and moments. The 6-DOF state vector used for this work is $z = [V_T \quad \alpha \quad \beta \quad p \quad q \quad r \quad x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T$. Alternatively, **z** can be split into two vectors, the 6-dimensional configuration of aircraft $\eta = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T$ and velocity vector $\nu = \begin{bmatrix} V_T & \alpha & \beta & p & q & r \end{bmatrix}^T$. Actuation vector $\mu = \begin{bmatrix} \mu_{lt} & \mu_{rt} & \mu_e & \mu_a & \mu_r \end{bmatrix}^T$ for the GTM includes elevator μ_e, aileron μ_a, and rudder deflections μ_r, as well as explicit left and right engine thrust terms (μ_{lt}, μ_{rt}) representing differential throttle control. The nonlinear aircraft dynamics are then represented by

$$\dot{z} = f(z, \mu) \tag{1}$$

where *f* represents a set of twelve nonlinear equations.

The GTM aircraft model was not available in analytical form. Instead, this 6-DOF nonlinear model is built on stability and control derivative data provided by our research sponsor in tabular form. This data is given over a discrete set of constant angles of attack α_0, sideslip angles β_0, and Mach numbers. A trilinear interpolation algorithm[28] is used to approximate the values of these derivatives for speeds ranging from Mach 0.1 to Mach 0.8, α values from -2º to 12º, and β values from -10º to 10º. The stability and control derivatives are then used to calculate dimensionless aerodynamic coefficients C_D, C_L, C_Y, C_l, C_M, and C_N, where subscripts *D*, *L*, *Y*, *l*, *M*, and *N* represent aerodynamic drag, lift, sideforce, rolling moment, pitch moment, and yawing moment, respectively. These coefficients are used to derive the aerodynamic forces and moments given control surface deflections, which, together with the forces and moments produced by engine thrust, dictate aircraft motion over time. Full control authority is assumed for this structurally damaged aircraft model. Therefore, deflection limits for control surfaces μ_e, μ_a, and μ_r are assumed ± 30º. Left and right engine thrust values μ_{lt} and μ_{rt} are constrained between zero for no thrust and 1.0 for maximum thrust, assumed to be 40,000 lbs for each GTM engine. All controls are modeled as ideal, neglecting the lag between issuing the command and the actual response.

This work adopts trimmed flight states as the building blocks for a landing trajectory. A *trim state* is an equilibrium (non-accelerating) flight condition, a state where linear and angular velocities in body coordinates are constant. Mathematically, the trimmed flight condition can be expressed as

$$\dot{\nu}^* = 0 \tag{2}$$

where the asterisk is used to denote equilibrium. Given our representation of aircraft state, steady trimmed flight is achieved under the following conditions:

$$\begin{aligned} &(\dot{V}_T, \quad \dot{\alpha}, \quad \dot{\beta}) \equiv 0 \\ &(\dot{p}, \quad \dot{q}, \quad \dot{r}) \equiv 0 \\ &(\dot{\phi} \quad \dot{\theta}) \equiv 0 \\ &\dot{\psi} \equiv \dot{\psi}^* \\ &\dot{h} \equiv \dot{h}^* \end{aligned} \tag{3}$$

where $\dot{\psi}^*$ and $\dot{h}^*$ are respectively the desired constant climb rate and constant turn rate for the trim condition. With these constraints, eight aircraft state variables fully define a steady climbing-turning trim state and can be grouped into a reduced state vector

$$\bar{z}^T = [V_T \quad \alpha \quad \beta \quad p \quad q \quad r \quad \phi \quad \theta] \tag{4}$$

By solving the aircraft equations of motion under the conditions expressed in Eq. (3), trim state $\bar{z}^*$ and trim control settings μ^* can be found for a given velocity V_T, climb rate $\dot{h}^*$, and turn rate $\dot{\psi}^*$. $\bar{z}^*$ and μ^* also depend on aircraft altitude *h* since atmospheric density variation as a function of altitude requires control setting changes to maintain the trim state. Each trimmed flight state can typically be achieved over a range of airspeeds. Therefore, the trim state $\bar{z}^*$ and the corresponding trim control settings μ^* are actually functions of four parameters that define a steady climbing-turning trimmed flight condition, expressed by the quadruplet $(h^*, V_T^*, \dot{h}^*, \dot{\psi}^*)$ where h^* and V_T^* are the altitude and true airspeed for the trim state, respectively. For this work, given a non-zero climb rate, h^* is presumed to be the initial altitude of that trim flight. This is a good assumption for relatively short flight segments or for descending flight segments typical for landing trajectories, particularly since envelope typically degrades at higher altitudes and since the commanded trim state can be actively maintained by varying control input settings.

III. Landing Site Search

Landing Site Search (LSS), one of the two primary components of the adaptive flight planning architecture (Figure 1), provides a position (runway approach end) and heading (runway heading) used as the "target" state by the trajectory planner. The Landing Site Search procedure and software used for this work are based on our previous research,[7] which provided a complete LSS capability but that required some extension for use with the damaged GTM aircraft due to the specific focus on loss-of-thrust emergency scenarios.

Landing Site Search is a four-step process. First, a *reachable footprint* is generated taking into account the range constraints imposed by the aircraft damage or failure scenario. Then, all *reachable* runways within this footprint are examined to identify the feasible subset that can safely accommodate the emergency landing, defining the set of *feasible* runways. In cases where more than one feasible runway is found, the LSS ranks the list of feasible runways based on a safety-oriented utility function.[7] The most desirable landing site (i.e., the highest-ranked feasible runway) is identified as the final state for the trajectory planner. In this work, the Landing Site Search process is extended to accommodate emergencies that do not limit range, such as the wing damage situation. This section first presents a brief introduction to the overall Landing Site Search procedure. The modified footprint generation algorithm applicable to the damaged GTM scenario is the contribution of this work, a description of which is followed by a brief review of the existing feasible runway identification and ranking processes.

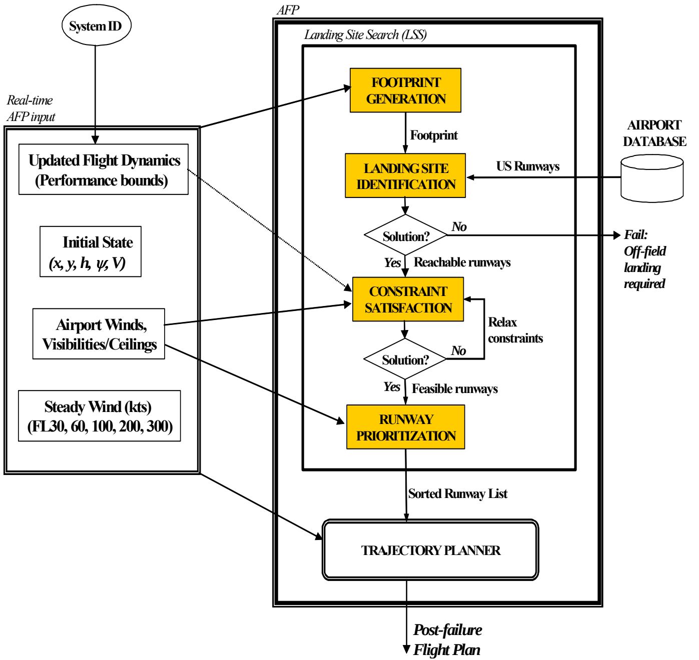


Figure 2: Landing Site Search Architecture[7]

Figure 2 shows the Landing Site Search (LSS) process. Inputs include a U.S. airport database, a degraded performance model, the initial LSS aircraft state, and the airport wind/weather conditions. The initial LSS aircraft state is specified by the instantaneous location (x, y), altitude (h), heading (ψ), and velocity (V_T) of the aircraft when the emergency occurs. Together with the runway data, real-time airport weather conditions are used to evaluate the feasibility and "quality" of a runway/airport to accommodate a safe emergency landing. The output of the Landing Site Search is a sorted runway list, which contains the candidate runways ranked according to their safety-oriented utility values. The top candidate is then selected as the landing site to which the disabled aircraft should be guided by the trajectory planner.

The LSS process includes footprint generation, landing site identification, constraint satisfaction, and utility-based runway prioritization. Footprint Generation calculates the approximate boundary of the region the post-failure/damage aircraft can or should reach before it is forced to land. Landing Site Identification builds a list of all runways within this footprint, then Constraint Satisfaction rules out the runways that cannot meet hard safety constraints such as minimum length or surface (e.g., asphalt vs. water). Particularly near urban areas, multiple feasible runways can be reached. The final step is to evaluate the feasible runways in terms of their desirability ranging from safety-oriented factors (e.g., runway length, width, instrument procedures) to company preferences (e.g., facility availability).[7] The final output is then the sorted runway list.

The Footprint Generation algorithm used in this work is built on previous work focused on the loss-of-thrust emergency.[7] In the original study, aircraft *footprint* had the definition "maximum region the aircraft can reach on the ground". When thrust is lost, there is a hard constraint on time aloft, so a maximum-range footprint is appropriate. The algorithm developed previously approximated footprint as a circular profile shifted to account for winds aloft.[7] In damage or failure situations where the aircraft can remain aloft until fuel is exhausted, the practical requirement to safely land the disabled aircraft near-term limits footprint size. Furthermore, it may be more difficult to describe footprint geometry for a disabled aircraft with highly constrained performance envelope. To manage these cases, a *virtual footprint* boundary is defined to artificially constraint the reachable region. For simplicity and compatibility with the loss-of-thrust footprint, a circular boundary is defined for this virtual footprint. The center of the circle is defined as the initial location at which the emergency occurs. The radius of this circle is initially set to a "reasonable" user-defined constant (for example, 50 nautical miles for this work) which is then incremented until a feasible runway is found, or an upper bound value is reached, whichever comes first. The aircraft footprint is then defined as the region within this circular boundary as shown in Figure 3.

A runway is considered *reachable* if its distance from the initial failure/damage site is less than the radius of the footprint. Each reachable runway must meet minimum runway length, minimum runway width, maximum crosswind component, runway surface type, and reported visibility vs. instrument approach minimum constraints. Reachable runways that cannot meet one or more of these constraints are eliminated, with remaining runways defined as the *feasible* runway set. There is a probability that all the reachable runways are disqualified by this procedure. For loss-of-thrust, the constraints were relaxed, reducing safety margins until at least one feasible runway was identified. With the virtual footprint, the radius is instead incrementally increased since safety is of paramount importance and footprint was limited by a preference to land nearby rather than a hard performance constraint. Once one or more feasible runways are identified, the following utility function is used to rank these runways:[7]

$$U = \sum_i C_i \cdot w_i = C_1 \cdot \frac{r_l}{r_{l,\max}} + C_2 \cdot \frac{r_w}{r_{w,\max}} + C_3 \cdot q_I + C_4 \cdot \left(\frac{d}{d_{\max}}\right) + C_5 \cdot \frac{w_h}{w_{h,\max}} + C_6 \cdot \frac{\left(w_{c,\max} - w_c\right)}{\left(w_{c,\max} - w_{c,\min}\right)} + C_7 \cdot q_S + C_8 \cdot q_f \tag{5}$$

Runway utility is defined as a weighted sum of runway length r_l and width r_w, instrument approach quality q_I, distance d from the footprint boundary, headwind velocity w_h, crosswind velocity w_c, surface quality q_S, and facility availability measure q_f.[7] Each term is normalized to guarantee cost values in the range [0.0 1.0], with weighting factors C_i (i = 1, 2, …, 8) adjustable to pilot preference or emergency type. The default weight set used for the loss-of-thrust case study[7] is

$$\{C_1,\ \ C_2,\ \ ...,\ \ C_8\} = \{0.15\ \ 0.15\ \ 0.15\ \ 0.15\ \ 0.1\ \ 0.1\ \ 0.1\ \ 0.1\} \tag{6}$$

giving equivalent preference to runway length, runway width, instrument approach quality, and runway distance to the footprint boundary. As will be seen for our damaged GTM, landings must be fast and with minimal control, suggesting a weight set that favors long and wide runways. The following weights were thus used for this work:

$$\{C_1,\ \ C_2,\ \ ...,\ \ C_8\} = \{0.4\ \ 0.4\ \ 0.05\ \ 0.01\ \ 0.05\ \ 0.03\ \ 0.03\ \ 0.03\} \tag{7}$$

As indicated by the values of the weighting factors, runway length and secondarily runway width are prioritized. However, non-zero weights still exist for the other parameters to distinguish runways of near-equivalent length and width. These two weight sets yielded different sorted runway output in our GTM case studies. In a scenario where the GTM is over San Francisco at (37.44^o North, 122.12^o West), a footprint with radius of 100 nautical miles is generated for the damaged GTM. Within the footprint, 28R/SFO (runway 28R at SFO) is the top candidate runway with Eq. (6) weights while 22L/MHR is favored by the Eq. (7) weights. 28R/SFO was nominally preferred due to its facilities. However, SFO has runway length 11,870 ft and width 200 ft. 22L/MHR is 11,301 ft in length and 300 ft in width, providing increased margin for lateral misalignment. This example shows that the selection of runway utility weight set can significantly affect the ranking of feasible runways.

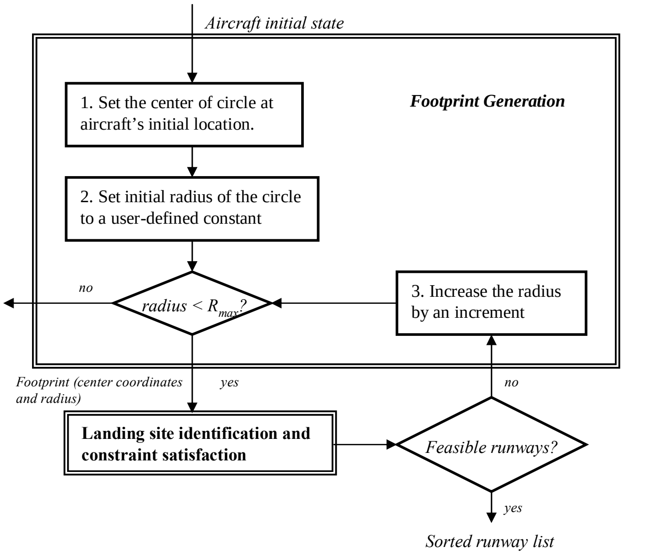

Aircraft initial state
Footprint Generation
1. Set the center of circle at aircraft's initial location.
2. Set initial radius of the circle to a user-defined constant
no
radius < R_max?
3. Increase the radius by an increment
Footprint (center coordinates and radius)
yes
no
Landing site identification and constraint satisfaction
Feasible runways?
yes
Sorted runway list

Figure 3: Damaged Aircraft Footprint to Limit Time Aloft.

IV. Trajectory Specification as Trim State Sequences

Once a landing runway is identified, the adaptive flight planner must compute a trajectory to this runway. This section overviews the procedure previously developed for an F-16 aircraft experiencing control surface jam failures.[9,8] In this work, segmented trajectories are defined as sequences of trimmed flight segments connected by transitions between these segments, a design believed to maximize intuitive comprehension by pilots and air traffic controllers. To determine the feasible trim states the post-failure/damage aircraft can achieve, a discrete flight envelope is computed for the disabled aircraft. Recall that a trim state is defined as a non-accelerating flight condition that can be maintained indefinitely. Once trim and transition information is compiled, a simplified aircraft kinematic model is used to compute position and heading change incurred during trimmed flight over a finite time period. The emergency path planner sequences trim states and adjusts their duration to accurately guide the aircraft to the designated landing runway. To manage trim state transitions and reject disturbances during trimmed flight, a closed-loop PID controller was developed to reduce transition settling time and provide close tracking of a desired flight trajectory. Based on the definition of aircraft trim state presented above, a nonlinear constrained optimization procedure is used to compute the trim state $\bar{z}^*$ and control settings under given flight conditions by minimizing the cost function

$$J_{trim}(z,\mu)=\frac{1}{2}\dot{\bar{z}}^T Q\dot{\bar{z}} \tag{8}$$

constrained by trimmed flight conditions

$$h=h^*, V_T=V_T^*, \tan\theta=\frac{ab+\sin\gamma^*\sqrt{a^2-\sin^2\gamma^*+b^2}}{a^2-\sin^2\gamma^*}, p=-\dot{\psi}^*\sin\theta, q=\dot{\psi}^*\cos\theta\sin\phi, r=\dot{\psi}^*\cos\theta\cos\phi \tag{9}$$

where $a=\cos\alpha\cos\beta$, $b=\sin\phi\sin\beta+\cos\phi\sin\alpha\cos\beta$, and the trimmed flight path angle γ^* is subject to constraint $\sin\gamma^*={h^*}/{V_T^*}$. The first two constraints in Eq. (9) directly constrain aircraft altitude and airspeed. The third constraint indirectly specifies desired climb rate, while the last three constraints indirectly specify desired turn rate, as well as constraining roll and pitch rates to be zero. Under the above constraints, the minimization of (8) over aircraft state variables and the control inputs yields

$$J_{trim}^*=J_{trim}(z^*,\mu^*)=\min_{z,\mu}\{J_{trim}(z,\mu)\} \tag{10}$$

where z^* and μ^* are the minimized solution vectors. z^* is then a trim state if $J_{trim}^*=0$. Practically, a multi-dimensional numerical optimization algorithm is used to compute an approximate solution since this problem cannot be solved analytically. The algorithm iteratively varies the values of a set of independent variables until

$$J_{trim}<\varepsilon<<1 \tag{11}$$

where ε is small positive scalar. In this work, z^* is considered a trim state if the associated cost satisfies Eq. (11).

A trim state z^*, solved by above procedure, is only considered feasible if the corresponding control settings μ^* satisfy the following constraints on the control input:

$$|\mu_{lt}-0.5|\le 0.5, |\mu_{rt}-0.5|\le 0.5, |\mu_e|\le 30^\circ, |\mu_a|\le 30^\circ, |\mu_r|\le 30^\circ \tag{12}$$

The first two constraints limit left and right engine thrust (%) to a range [0 1], while the other constraints limit the control surfaces deflections between ±30º. A steady trimmed flight condition $(h^*,V_T^*,\dot{h}^*,\dot{\psi}^*)$ is considered feasible if there exists a valid trim state z^* and control vector μ^* for this flight condition. The flight envelope is then defined as the set of feasible trimmed flight conditions $(h^*,V_T^*,\dot{h}^*,\dot{\psi}^*)$ in four-dimensional space.

To further categorize these feasible trimmed flight conditions, aircraft stability and controllability within a small neighborhood of each trim state is assessed. Since aircraft dynamics are nonlinear, a full nonlinear analysis would be ideal for characterizing aircraft dynamics at these trim states. However, the damaged GTM aircraft model is based on discrete aerodynamic data in tabular form, with no analytical model available. Nonlinear systems can be approximated by a linearization of their dynamics about equilibrium points, in a small neighborhood surrounding that equilibrium point.[30] To perform a linear stability and controllability analysis for the aircraft system requires first the linearization of partial nonlinear aircraft dynamics represented as

$$\dot{\bar{z}}=\bar{f}(\bar{z},\mu) \tag{13}$$

about a trim state $\bar{z}_k^*$. The linear perturbation model about that trim state is described by

$$\dot{x}_k=A_k x_k+B_k u_k \tag{14}$$

where

$$x_k=\bar{z}-\bar{z}_k^*, \quad u_k=\mu-\mu_k^*, \; A_k=\left.\frac{\partial\bar{f}}{\partial\bar{z}}\right|_{\bar{z}=\bar{z}_k^*,\mu=\mu_k^*}, \; B_k=\left.\frac{\partial\bar{f}}{\partial\mu}\right|_{\bar{z}=\bar{z}_k^*,\mu=\mu_k^*}$$

In these equations, $\bar{f}$ is the system of nonlinear equations defining the dynamics of state $\bar{z}$ at trim altitude h_k^*, and A_k and B_k are the Jacobian matrices at the trim state, which are practically approximated by deriving the first-order differences. The nonlinear aircraft can be considered *stable* in a small neighborhood of a trim state if all eigenvalues of the corresponding linear perturbation system fall within the left-half complex plane, i.e., the real parts of the system eigenvalues are strictly negative numbers:

$$\boldsymbol{R}\{\lambda_i(A_k)\}<0 \qquad i=1,\ldots,8 \tag{15}$$

where $\lambda_i(A_k)$ is the i^{th} eigenvalue of A_k. When Eq. (14) is not naturally stable, it is necessary to check its stabilizability and controllability. A linear system is *stabilizable* if there exists a controller

$$u_k = -K_k x_k \tag{16}$$

that can make the closed-loop dynamics of (14) stable. This condition is represented by

$$\boldsymbol{R}\{\lambda_i(A_k - B_k K_k)\} < 0 \qquad i = 1, ..., 8 \tag{17}$$

While an unstable trim state is acceptable if it is stabilizable, preference is given to states that are *controllable*. The aircraft is considered controllable in a small neighborhood of a trim state z_k^* if the corresponding linear perturbation system about this trim state has a controllable matrix pair $[\mathbf{A}_k, \quad \mathbf{B}_k]$, or equivalently, has a controllability matrix

$$\mathbf{U}_C = [\mathbf{B}_k \quad \mathbf{A}_k\mathbf{B}_k \quad \mathbf{A}_k^2\mathbf{B}_k \quad \ldots \quad \mathbf{A}_k^{n-1}\mathbf{B}_k] \tag{18}$$

with full row rank *n*. The closed-loop eigenvalues of Eq. (14) can be assigned arbitrarily with the linear controller in Eq. (16) if the system is controllable. Therefore, controllability is a more dominant concern than stability and stabilizability, since a controllable trim state can be maintained despite disturbances given a capable control law.

An aircraft trim database *D* can now be generated by characterizing each trim state in four-dimensional space $(h, V_T, \dot{h}, \dot{\psi})$ in terms of feasibility (based on the nonlinear optimization), stability, and controllability. By fixing one component of the trim state quadruplet, in this case altitude h^*, as well as plotting each type of trim state with different colors in three-dimensional space, slices of the trim database can be presented. Figure 4 shows a trim database slice at a fixed altitude of 10 ft for the left-wing-damaged GTM. In this figure, a green asterisk in $(V_T, .\dot{h}, \dot{\psi})$ space indicates a naturally stable trim state for that trimmed flight condition, while a blue dot represents an unstable but controllable trim state. The unmarked area indicates infeasible or uncontrollable $(V_T, .\dot{h}, \dot{\psi})$.

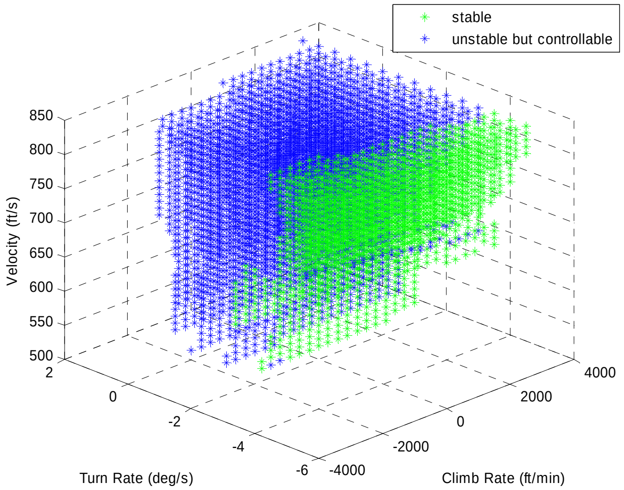


Figure 4: Envelope for the Damaged GTM at h^*=10 ft.

In addition to trimmed flight segments, an emergency landing trajectory also requires accurate characterization of the transition maneuvers that connect neighboring trim segments. A trim transition is defined as a finite time evolution between two trim states. A transition from trim state *i* into trim state *j* is defined intuitively as the change in flight conditions $(V_{T,i}^*, \dot{h}_i^*, \dot{\psi}_i^*) \rightarrow (V_{T,j}^*, \dot{h}_j^*, \dot{\psi}_j^*)$ over a finite transition time Δt, where $(V_{T,i}^*, \dot{h}_i^*, \dot{\psi}_i^*)$ denotes the initial trim flight condition and $(V_{T,j}^*, \dot{h}_j^*, \dot{\psi}_j^*)$ represents the terminal trim state. Correspondingly, the transition leads to a change in aircraft state and a change in the control setting. In this work, an interpolation algorithm is used to define a smooth profile over intermediate flight conditions as well as control settings over the transition. With this strategy, the desired flight condition during a transition $(V_{T,i}^*, \dot{h}_i^*, \dot{\psi}_i^*) \rightarrow (V_{T,j}^*, \dot{h}_j^*, \dot{\psi}_j^*)$ is given by

$$V_T^*(t) = V_{T,i}^* + \frac{V_{T,j}^* - V_{T,i}^*}{\Delta t}(t - t_0), \dot{h}^*(t) = \dot{h}_i^* + \frac{\dot{h}_j^* - \dot{h}_i^*}{\Delta t}(t - t_0), \dot{\psi}^*(t) = \dot{\psi}_i^* + \frac{\dot{\psi}_j^* - \dot{\psi}_i^*}{\Delta t}(t - t_0), \forall t \in [t_0, \quad t_0 + \Delta t] \tag{19}$$

where t_0 denotes transition start time. Using the same linear interpolation strategy, the desired control settings and the desired aircraft state at time *t* during the transition are given by

$$\mu^*(t) = \mu_i^* + \frac{\mu_j^* - \mu_i^*}{\Delta t}(t - t_0), \bar{z}^*(t) = \bar{z}_i^* + \frac{\bar{z}_j^* - \bar{z}_i^*}{\Delta t}(t - t_0) \qquad \forall t \in [t_0, \quad t_0 + \Delta t] \tag{20}$$

where the terminal trim state and trimmed control settings can be found by estimating the terminal altitude $h_j = h_i + 0.5 \times (\dot{h}_j^* + \dot{h}_i^*)\Delta t$. Eqs. (19) and (20) define a desired reference trajectory for the transition. However, open-loop control strategies do not typically provide good tracking performance. In fact, an open-loop controller cannot maintain stability during transitions involving naturally unstable but controllable trim states, used for the damaged GTM due to sparseness of naturally stable trim states. As a result, a closed-loop control strategy is required to stabilize such transitions and to characterize the closed-loop time response profile for our trajectory planner.

Adapted from previous work,[9] a nonlinear PID controller is designed to guarantee system stability, as well as to provide good configuration tracking over the transitions. Given the linear perturbation system from Eq. (14), this approach takes into account tracking errors to improve closed-loop tracking performance. The tracking error of a trimmed flight condition $(V_{T,k}^*, \dot{h}_k^*, \dot{\psi}_k^*)$ is defined as

$$y_k = \begin{bmatrix} V_T - V_{T,k}^* & \dot{h} - \dot{h}_k^* & \dot{\psi} - \dot{\psi}_k^* \end{bmatrix}^T = C_k x_k \tag{21}$$

where C_k is a constant matrix defined by

$$C_k = \left. \frac{\partial y_k}{\partial x_k} \right|_{\bar{z}_k^*, \mu_k^*} \tag{22}$$

Unlike Jacobian matrices A_k and B_k that must be computed numerically, C_k can be determined analytically. The definition of a new state vector ξ with dynamics $\dot{\xi}_k = -y_k$ leads to an augmented system

$$\begin{bmatrix} \dot{x}_k \\ \dot{\xi}_k \end{bmatrix} = \begin{bmatrix} A_k & 0 \\ -C_k & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \xi_k \end{bmatrix} + \begin{bmatrix} B_k \\ 0 \end{bmatrix} u_k \tag{23}$$

The substitution $\zeta_k = \begin{bmatrix} x_k^T & \xi_k^T \end{bmatrix}^T$ enables Eq. (23) to be written compactly as

$$\dot{\zeta}_k = \hat{A}_k \zeta_k + \hat{B}_k u_k \tag{24}$$

where

$$\hat{A}_k = \begin{bmatrix} A_k & 0 \\ -C_k & 0 \end{bmatrix}, \quad \hat{B}_k = \begin{bmatrix} B_k \\ 0 \end{bmatrix}$$

If this augmented system is controllable, there exists a feedback controller

$$u_k = -\hat{K}_k \zeta_k \tag{25}$$

that can make the closed-loop system

$$\dot{\zeta}_k = (\hat{A}_k - \hat{B}_k \hat{K}_k)\zeta_k \tag{26}$$

stable and the state vector ζ_k converge to zero asymptotically. As a result, the integral of the tracking error decays to zero asymptotically, thereby improving the configuration tracking performance. Extending the above control design for transition control leads to nonlinear PID controller

$$\begin{aligned} \mu(h(t), V_T^*(t), \dot{h}^*(t), \dot{\psi}^*(t)) &= \mu^*(t) - \hat{K}(t)\zeta(t) \\ &= \mu^*(t) - K_x(t)x(t) + K_\xi(t)\int_{t_0}^{t} y(\tau)d\tau \\ &= \mu^*(t) - K_x(t)(\bar{z}(t) - \bar{z}^*(t)) + K_\xi(t)\int_{t_0}^{t} \begin{bmatrix} V_T(\tau) - V_T^*(\tau) \\ \dot{h}(\tau) - \dot{h}^*(\tau) \\ \dot{\psi}(\tau) - \dot{\psi}^*(\tau) \end{bmatrix} d\tau \end{aligned} \tag{27}$$

where $\hat{K}(t) = \begin{bmatrix} K_x(t) & K_\xi(t) \end{bmatrix}$ and t_0 is the transition start time. The desired control settings $\mu^*(t)$ and the desired aircraft states $\bar{z}^*(t)$ are given by Eq. (20), while the desired flight condition ($V_T^*(t)$, $\dot{h}^*(t)$, $\dot{\psi}^*(t)$) is determined by Eq. (19). Similarly, $\hat{K}(t)$ is computed with linear interpolation and given by

$$\hat{K}(t) = \hat{K}_i + \frac{\hat{K}_i + \hat{K}_j}{\Delta t}(t - t_0), \qquad \forall t \in [t_0, t_0 + \Delta t] \tag{28}$$

where $\hat{K}_i$ is the controller gain matrix designed for the initial trim state, and $\hat{K}_j$ is the gain matrix designed for the terminal trim state. A gain scheduling strategy is utilized for this nonlinear controller. Different techniques can be used to design the controller gain matrix $\hat{K}_k$ for a trim state. For example, by placing the closed-loop poles of the Eq. (24) system at desired locations on complex plane, $\hat{K}_k$ can be determined for the trim state. These desired closed-loop poles represent the nominal closed-loop dynamics. However, this technique cannot be used for controller gain design with the GTM because no nominal controller is available. In this work, controller gains are designed for the GTM by using an LQR technique, which yields an optimal controller that minimizes a cost function over system error and control effort.

Given a controllable aircraft operating about a trim state, the LQR design finds an optimal control vector

$$u_k^* = -R_k^{-1}\hat{B}_k^T P\zeta_k \tag{29}$$

that minimizes the cost function

$$J_{LQR} = \int_{t_0}^{\infty}(\zeta_k^T Q_k \zeta_k + u_k^T R_k u_k)dt \tag{30}$$

where Q_k and R_k are $n \times n$ and $m \times m$ positive definite matrices, respectively, n is the length of ζ_k, and m is the length of u_k. The matrix P in Eq. (29) is defined by solving the Ricatti equation $\hat{A}_k^T P + P\hat{A}_k + Q_k = P\hat{B}_K R_k^{-1}\hat{B}_k^T P$.[32] The optimal control vector from Eq. (29) automatically guarantees the stability of the closed-loop system

$$\dot{\zeta}_k = (\hat{A}_k - \hat{B}_k R_k^{-1}\hat{B}_k^T P)\zeta_k \tag{31}$$

if weighting matrices Q_k and R_k are positive definite. In practice, Q_k and R_k are typically chosen to be diagonal matrices. Varying Q_k and R_k yields the optimal solution over a suite of closed-loop system responses. Generally, a dominant Q_k enables a closed-loop system with high control precision and high control effort, while a dominant R_k yields a closed-loop system with low control effort and low control precision. By following the LQR design as above, the control gain matrix $\hat{K}_k$ in Eq. (25) is therefore determined as $\hat{K}_k = R_k^{-1}\hat{B}_k^T P$. Although different weighting matrices can be chosen for each trim state, constant Q_k and R_k are used in this work since they yield uniform closed-loop system performance characteristics well within the flight envelope. Furthermore, constant weighting matrices can significantly simplify the control design by avoiding the tedious work of manually tuning the controller at each trim state. For the left-wing-damaged GTM aircraft, the diagonal constant matrix Q_k is chosen to minimize lateral motion state errors since the missing left wingtip results in difficulty controlling lateral motion. Meanwhile, roughly even weights are enforced on the five components of control effort by the matrix R_k defined as $R_k = r \times I_{5\times 5}$ where r is a constant, set to r=100 for the case study presented below.

Simulation results from the damaged GTM demonstrate the success of this control strategy. Performance of the PID controller during a trim transition is shown in Figure 5. In the simulation, the aircraft at initial altitude 5,000 ft is commanded to perform a desired transition from an initial trimmed flight condition with $V_T^* = 750 ft/s$, $\dot{h}^* = 0$, and $\dot{\psi}^* = 0$, to a terminal trimmed flight condition with $V_T^* = 750 ft/s$, $\dot{h}^* = -300 ft/\min$, and $\dot{\psi}^* = -0.5\deg/s$, over a time interval of 15 seconds, the nominal transition time interval presumed in our GTM case study.

As shown by Figures 5(a) and 5(b), the controller provides good tracking of the desired trim states as well as the transition between them. The relatively large terminal tracking errors shown in Figure 5(c) will not influence subsequent trajectory planning since the trajectory planner uses the actual transition path changes rather than the desired path changes for analyses. Subplot (d) shows that all the control inputs are well within their saturation limits.

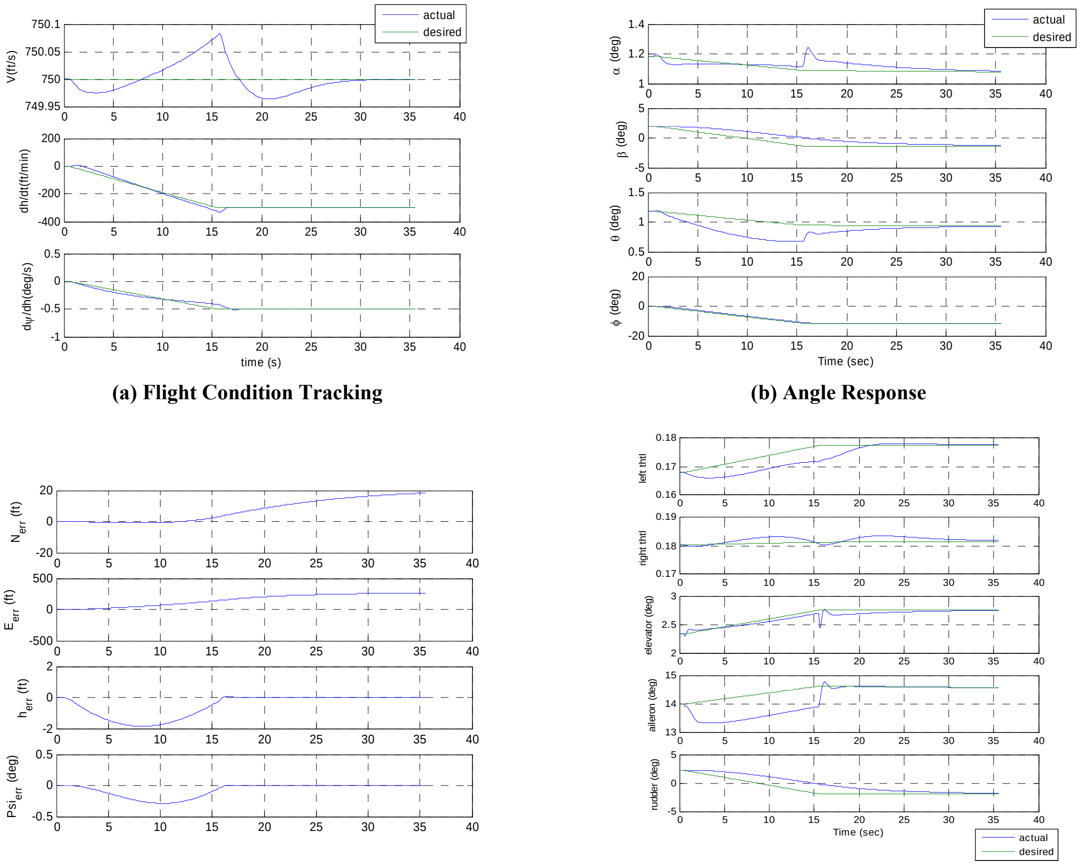
This work uses a kinematic model to specify the aircraft's motion, both position and heading, along a segmented flight path.[9] The initial aircraft flight path configuration is represented by a 4×4 matrix

$$F_0 = \begin{bmatrix} R_{\psi_0}^T & p_0 \\ 0 & 1 \end{bmatrix} \tag{32}$$

where p_0 is the 3×1 vector that specifies initial aircraft position in the inertial space, and

$$R_{\psi_0} = \begin{bmatrix} \cos\psi_0 & \sin\psi_0 & 0 \\ -\sin\psi_0 & \cos\psi_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{33}$$

where ψ_0 is the aircraft's initial flight path heading.



(a) Flight Condition Tracking

(b) Angle Response

(c) Ground Tracking Error

(d) Control Response

Figure 5: Closed-loop GTM Transition Simulation Results using an LQR Controller Design

The terminal aircraft flight path configuration after a sequence of N consecutive trimmed flight path segments is then represented by

$$F_N = F_0 \prod_{i=1}^{N} G_{i-1,i} G((V_T^*)_i, \dot{h}_i^*, \psi_i^*, \Delta t_i) \tag{34}$$

where $G_{i-1,i}$ represents the flight path change over the transition connecting the (i-1)th and ith trimmed flight segments, and $G((V_T^*)_i, \dot{h}_i^*, \dot{\psi}_i^*, \Delta t_i)$ represents the flight path change over the ith trimmed flight segment. For a trimmed flight segment, flight path change is a function of flight condition $((V_T^*)_i, \dot{h}_i^*, \dot{\psi}_i^*)$ and duration Δt_i:

$$G((V_T^*)_i, \dot{h}_i^*, \dot{\psi}_i^*, \Delta t_i) = \begin{bmatrix} R^T_{\dot{\psi}_i^* \Delta t_i} & R_{\psi_{0,i}} \Delta p_i \\ 0 & 1 \end{bmatrix} \tag{35}$$

where ψ_{0i} denotes the initial heading of this trimmed flight segment, and Δp_i, the flight path position change over Δt_i, is computed rapidly since it can be expressed analytically. For the transition segment:

$$G_{i-1,i} = \begin{bmatrix} R^T_{\Delta\psi_{i-1,i}} & R_{(\psi_0)_{i-1,i}} \Delta p_{i-1,i} \\ 0 & 1 \end{bmatrix} \tag{36}$$

The flight path position change $\Delta p_{i-1,i} = p(t_{i-1}) - p(t_i)$ and heading change $\Delta\psi_{i-1,i} = \psi(t_{i-1}) - \psi(t_i)$ over the transition must be obtained via numerical simulation, where t_{i-1} denotes transition starting time and t_i denotes the time by which all the transients have died away. Performing the numerical simulations over the spectrum of potential trim transitions yields a transition database M from which the trajectory planner retrieves the path changes that occur during each transition. This kinematic path representation with the support of trim database D and transition database M for a particular failure/damage scenario provide the tools required to generate segmented post-failure/damage landing trajectories.

V. Trajectory Planning

The task of the trajectory planner is to identify a sequence of trimmed flight conditions that allows the aircraft to reach the landing runway with the correct heading. Although trim transitions are necessary to connect neighboring trim segments, it is sufficient to represent a trajectory (flight plan) by only specifying each trim flight segment, since a constant-time trim transition is completely specified by the initial and terminal trim states it connects. Let each trimmed flight segment be specified as

$$s_i = ((V_T^*)_i, \dot{h}_i^*, \dot{\psi}_i^*, \Delta t_i) \tag{37}$$

where the triplet $((V_T^*)_i, \dot{h}_i^*, \dot{\psi}_i^*)$ specifies the segment's trimmed flight condition and Δt_i is the duration of the segment. A trajectory can then be expressed as a sequence of *N* trim segments:

$$P = \{s_i\}_{i=1}^N = \{((V_T^*)_i, \dot{h}_i^*, \dot{\psi}_i^*, \Delta t_i)\}_{i=1}^N \tag{38}$$

A candidate trajectory plan $\tilde{P}$ is a partially-instantiated plan consisting of valid trim state triplets for each segment, but with no durations specified. Thus, a candidate plan $\tilde{P}$ can be expressed as

$$\tilde{P} = \{((V_T^*)_i, \dot{h}_i^*, \dot{\psi}_i^*)\}_{i=1}^N \tag{39}$$

The Landing Site Search procedure executes quickly (under one second on a 1GHz PC). Therefore, several aircraft states can be considered approximately equivalent if the LSS is executed immediately after the failure or damage occurs. These instantaneous states are: state at the time when failure/damage occurs (or is detected, assumed concurrent in this work), state at the time the LSS is executed, and state at the time the target runway is selected. The value for these near-equivalent states is straightforwardly set to the state just after the failure/damage occurs. This value will be used by the trajectory planner as the initial state, of which the position and heading components are represented as p_{init} and ψ_{init}. Correspondingly, the desired landing site can be represented as p_{des} and ψ_{des}. The planner builds a feasible trajectory between the initial (p_{init}, ψ_{init}) and the desired (p_{des}, ψ_{des}) states in two stages. The first stage checks the relative position of p_{init} with respect to p_{des} and builds an initial positioning trajectory (*trajectory part I*) if p_{init} doesn't satisfy a three-dimensional geometric constraint, expressed as

$$\begin{gathered} \underline{h} \le h_{init} - h_{des} \le \overline{h} \\ (x_{init} - x_{des})^2 + (y_{init} - y_{des})^2 \le r^2 \end{gathered} \tag{40}$$

where (*x, y*) and *h* denote respectively the location and the altitude of a 3-D point. By following *trajectory part I*, the aircraft flies to an intermediate position p_{int}, which falls in the cylindrical neighborhood of p_{des}, with heading ψ_{int}. Next, a search-based path planner constructs *trajectory part II* that connects the intermediate $(p_{\text{int}}, \psi_{\text{int}})$ to the final (p_{des}, ψ_{des}). For consistency with the subsequent discussion, the initial state where *trajectory part I* is generated (i.e., (p_{init}, ψ_{init})) is denoted by (p_0^I, ψ_0^I) and the initial state for planning *trajectory part II* (i.e., $(p_{\text{int}}, \psi_{\text{int}})$) is denoted by (p_0^{II}, ψ_0^{II}). The subscript "0" denotes initial state for generating the trajectory components, while the superscript *I* and *II* denote *trajectory part I* and *trajectory part II* respectively.

Figure 6 shows this two-step trajectory planning procedure in the context of the Adaptive Flight Planner (AFP). At step 5 and step 9, *PLANNER_I* and *PLANNER_II* generate respectively the partial flight plan P^I for *trajectory part I* and the other partial plan P^{II} for *trajectory part II*. Time t_{plan} is the time interval over which the aircraft flies from (p_0^I, ψ_0^I) to (p_0^{II}, ψ_0^{II}) by executing P^I, or the duration of *trajectory part I*. t_{plan} is also used as a planning time constraint for *PLANNER_II* since P^{II} must be planned before it can be executed. Step 7 evaluates whether t_{plan} is sufficient for *PLANNER_II* to complete at least one feasible plan, selecting an alternate landing site if t_{plan} is too brief. Time t_{min} is the minimum value set to test t_{plan} and is empirically selected to be 60 seconds for the case study since it is sufficient for all the scenarios studied in this work. The efficient execution (under one second) of Steps 1 through 7 guarantees the aircraft is still approximately in state (p_0^I, ψ_0^I) when P^I is initiated.

Trajectory planners *PLANNER_I* and *PLANNER_II* build solutions as sequences of trim states found in the post-failure/damage database. To enable real-time plan development, the space of possible trim states to be sequenced must have a tractable size. The full set of controllable states in the trim database can be represented as the set

$$D = \left\{\left(h_k^*,\ (V_T^*)_k,\ \dot{h}_k^*,\ \dot{\psi}_k^*\right)\right\}_{k=1,\dots,N_D} \tag{41}$$

where N_D is the total number of the trim states in database *D*. Because a full trim database is developed to provide understanding of flight characteristics as well as provide candidate trim states for the trajectory planners, N_D is very large even for the left-wing-damaged GTM aircraft. Tractable planning thus makes it necessary to reduce this database to a small subset of its original size.

```
1. Set AFP flight envelope/algorithmic model based on type of failure Fail
2. Compute initial state $(p_0^I, \psi_0^I)$
3. Find sorted feasible landing site list ls_list via LSS: $ls\_list \leftarrow LSS(p_0^I, \psi_0^I, Fail)$
4. if ls_list is empty
       alert pilot AFP cannot find a feasible runway; exit
   else
       determine the desired ( $p_{des}$, $\psi_{des}$ ) by copying the top node in the ls_list, then remove the top node
   end if
5. Execute trajectory planner I: $(P^I, p_0^{II}, \psi_0^{II},) \leftarrow PLANNER\_I(p_0^I, \psi_0^I, p_{des}, \psi_{des})$
6. Compute the allowed planning time $t_{plan}$ for trajectory planner II : $t_{plan} = time(p_0^{II}, \psi_0^{II}) - time(p_0^I, \psi_0^I)$
7. if $t_{plan} < t_{min}$
       Choose next-highest ranked landing runway (goto step 4)
   end if
8. Initiate $P^I$ (continue to step 9 concurrently)
9. Execute trajectory planner II until completed or until $t_{plan}$ expires: $L \leftarrow PLANNER\_II(p_0^{II}, \psi_0^{II}, p_{des}, \psi_{des}, t_{plan})$
10. if L is not empty
       $P^{II} \leftarrow L(1)$
    else
       alert pilot AFP was unable to find a landing trajectory; exit
    end if
11. Execute $P^{II}$ to landing runway
```

Figure 6: Two-Step Trajectory Planning in the Context of AFP

Since altitude cannot be independently specified, *D* is first contracted over altitude to produce the flight condition database *D'*, the intersection of all three-dimensional altitude slices, from h_0 to h_n, in the trim database:

$$D' = \bigcap_{i=0}^{n} D(h_i^*, (V_T^*)_k, \dot{h}_k^*, \dot{\psi}_k^*) = \{((V_T^*)_k, \dot{h}_k^*, \dot{\psi}_k^*)\}_{k=1,\ldots,N_{D'}} \tag{42}$$

where $N_{D'} < N_D$. This procedure is successful so long as this intersection yields a sufficiently large set of trim states, which has been the case for all GTM and F-16 failures analyzed to-date.[9] Further contraction of the database can be accomplished by removing additional climb rate, turn rate, and airspeed points, retaining a sufficient subset approximately spanning the flight envelope. The new, contracted database $\tilde{D}$ can be represented as

$$\tilde{D} = \{((V_T^*)_k, \dot{h}_k^*, \dot{\psi}_k^*)\}_{k=1,\ldots,N_{\tilde{D}}} \tag{43}$$

where $\tilde{D} \subset D'$ by definition and $N_{\tilde{D}}$ is the size of the contracted database with $N_{\tilde{D}} << N_D$. A heuristic method was utilized to perform the database contraction $D' \rightarrow \tilde{D}$.[9] Transition database *M* is also straightforwardly reduced in accordance with $\tilde{D}$, resulting in databases $\tilde{M}$ to be used with $\tilde{D}$ by the trajectory planner.

An efficient algorithm *PLANNER_I* algorithm (Figure 7) was developed to rapidly plan landing *trajectory part I*, which has two effects. First, by following this trajectory the aircraft can fly into a neighborhood region of (p_{des}, ψ_{des}). Next, the time required to fly this trajectory will allow search-based *PLANNER_II* sufficient time to find a feasible landing *trajectory part II*. The *PLANNER_I* algorithm is shown in Figure 7. The inputs to *PLANNER_I* are initial aircraft 3-D location and heading, and the desired terminal location and heading. In Figure 7, step 4-13 plan the first trimmed flight segment s_1^I (spiral flight or circle flight), which is used to adjust aircraft altitude so that the altitude offset of this segment terminal point over the desired runway, $h_1^I - h_{des}$, is upper and lower bounded by $\bar{h}$ and $\underline{h}$ respectively. Step 15 pre-defines the second trimmed flight segment s_2^I, a circular segment that bridges segment s_1^I and the last segment s_3^I if determined necessary subsequently in the algorithm. Steps 18-25 build a third trimmed flight segment s_3^I if the terminal position of segment s_2^I is too far away from the runway (x_{des}, y_{des}). Segment s_3^I is then a straight and level flight toward the runway (x_{des}, y_{des}). *PLANNER_I* assumes that the post-failure/damage aircraft is capable of trimmed spiral ascending and descending flight, as well as circling and straight level flights. In the damaged GTM case study, the required trimmed turn rates and climb rates are selected manually from the trim database $\tilde{D}$, and in this case transitions between all trim state pairs are feasible. In future work, the planner may select these trim states automatically from $\tilde{D}$. The Figure 7 algorithm was successfully used for the results presented in this work. However, extensive simulations conducted as part of this research and for NASA Ames have revealed that the last flight segment s_3^I, a trimmed straight-line level flight toward the desired landing site, is unnecessary and potentially detrimental provided straight flight is possible since the range reduction over this segment can be handled in *trajectory part II.* Furthermore, in some situations, an initial state of *trajectory part II*, (p_0^{II}, ψ_0^{II}), that is too close to the desired landing site (p_{des}, ψ_{des}), may decrease the number of solutions from *PLANNER_II* if the aircraft is limited to shallow turn rates. An appropriate selection for *r* at Figure 7, step 18 may prevent this problem. The computation of *r* is dependent on the turning ability of the aircraft. Simulations show that a large *r* will be required if the aircraft has a large minimum turn radius in $\tilde{D}$. In this work, *r* is set manually to a safe static value for the case study. While this approach works well for the scenarios that will be presented below, it also suggests the need of an automatic algorithm that computes *r* based on the turning ability of the aircraft.

```
Algorithm: $PLANNER\_I(p_0^I, \psi_0^I, p_{des}, \psi_{des})$
1.  Initialize $P^I$ as an empty plan
2.  Initialize $(p_{end}^I, \psi_{end}^I) \leftarrow (p_0^I, \psi_0^I)$
3.  Read data from databases $\tilde{D}$ and $\tilde{M}$
4.  Compute altitude offset: $\Delta h = h_0^I - h_{des}$
5.  if $\Delta h > \bar{h}$
6.      $(s_1^I, p_1^I, \psi_1^I) \leftarrow GetSpiralD\,own(p_0^I, \psi_0^I, p_{des}, \tilde{D}, \tilde{M}, ((V_T^*)_0^I, \dot{h}_0^I, \dot{\psi}_0^I))$
7.  else if $\Delta h < \underline{h}$
8.      $(s_1^I, p_1^I, \psi_1^I) \leftarrow GetSpiralU\,p(p_0^I, \psi_0^I, p_{des}, \tilde{D}, \tilde{M}, ((V_T^*)_0^I, \dot{h}_0^I, \dot{\psi}_0^I))$
9.  else
10.     Set segment duration #1: $\Delta t_1^I \leftarrow 60$
11.     $(s_1^I, p_1^I, \psi_1^I) \leftarrow GetCircle\ (p_0^I, \psi_0^I, p_{des}, \Delta t_1^I \equiv 60,, \tilde{T}, \tilde{M}, ((V_T^*)_0^I, \dot{h}_0^I, \dot{\psi}_0^I))$
12. end if
13. Append flight segment $s_1$ to plan $P^I$: $P^I \leftarrow Append(P^I, s_1^I)$
14. Set segment duration #2: $\Delta t_2^I \leftarrow 0$
15. $(s_2^I, p_2^I, \psi_2^I) \leftarrow GetCircle(s_1^I, p_1^I, \psi_1^I, p_{des}, \tilde{D}, \tilde{M})$
16. Append flight segment $s_2^I$ to plan $P^I$: $P^I \leftarrow Append(P^I, s_2^I)$
17. Compute the position offset: $d = \left|(x_2^I, y_2^I) - (x_{des}, y_{des})\right|$
18. if $d \leq r$
19.     $(p_{end}^I, \psi_{end}^I) \leftarrow (p_2^I, \psi_2^I)$
20. else
21.     $(s_2^I, s_3^I, p_3^I, \psi_3^I) \leftarrow FlyToLS(s_2^I, p_2^I, \psi_2^I, p_{des}, \tilde{D}, \tilde{M})$
22.     Modify segment $s_2^I$ in $P^I$: $P^I \leftarrow Modify(P^I, s_2^I)$
23.     Append segment $s_3^I$ to $P^I$: $P^I \leftarrow Append(P^I, s_3^I)$
24.     $(p_{end}^I, \psi_{end}^I) \leftarrow (p_3^I, \psi_3^I)$
25. end if
26. $(p_0^{II}, \psi_0^{II}) \leftarrow (p_{end}^I, \psi_{end}^I)$
return $(P^I, p_0^{II}, \psi_0^{II})$
```

Figure 7: *PLANNER_I* Algorithm

Once *PLANNER_I* has returned plan P^I for *trajectory part I* and intermediate state (p_0^{II}, ψ_0^{II}) defined as the initial state for *trajectory part II*, *PLANNER_II* can then complete the entire flight plan to desired landing site P_{des} with desired heading ψ_{des}. At the top level, *PLANNER_II* combinatorially searches the reduced trim database for a partially instantiated solution $\tilde{P}^{II}$ composed of N^{II} ordered trim segments, then it performs a continuous optimization over flight segment durations Δt_i to determine the existence of a landing solution P^{II} given $\tilde{P}^{II}$.[9] Figure 8 shows this algorithm; N^{II} is set to 4 in this work, a choice of which was motivated in previous work.[9] Given reduced trim database size $N_{\tilde{D}}$ and number of segments N^{II}, there are a total $(N_{\tilde{D}})^{N^{II}}$ trim flight sequences possible. The i^{th} *while* loop examines the existence of solution given a unique trim state ordering P_i^{II} optimized over segment durations.

```
Algorithm: $PLANNER\_II(p_0^{II}, \psi_0^{II}, p_{des}, \psi_{des}, t_{plan})$
1.  Adopt $\tilde{D}$ and $\tilde{M}$ from $PLANNER\_I$
2.  Determine the number of trim states in $\tilde{D}$: $N_{\tilde{D}} \leftarrow SizeOf(\tilde{D})$
3.  Compute the total number of possible trim sequences: $i_{\max} \leftarrow Power(N_{\tilde{D}}, N^{II})$, clear index: $i \leftarrow 0$
4.  Initiate an empty plan list: $L \leftarrow NIL$
5.  while $(i < i_{max})$ and $(\delta t < t_{plan})$
6.      $\tilde{P}_i^{II} \leftarrow GetCandidatePlan(i, \tilde{D}, N^{II})$
7.      if false $\leftarrow ValidateSequence(\tilde{P}_i^{II}, \tilde{M})$
8.          $i \leftarrow i+1$
9.          continue
10.     end if
11.     if false $\leftarrow PlanConstraint(\tilde{P}_i^{II})$
12.         $i \leftarrow i+1$
13.         continue
14.     end if
15.     $(J_{plan}^*, P_i^{II}) \leftarrow GetDurations(\tilde{P}_i^{II}, \tilde{M}, p_0^{II}, \psi_0^{II}, p_{des}, \psi_{des})$
16.     if $J_{plan}^* \le \varepsilon_{plan}$   (cost of plan is acceptably low indicating accurate solution)
17.         if false $\leftarrow CheckAltitude(P_i^{II})$
18.             $i \leftarrow i+1$
19.             continue
20.         end if
21.         $l_i^{II} \leftarrow ComputeLength(P_i^{II})$
22.         $L \leftarrow SortList(L, P_i^{II})$
23.         $i \leftarrow i+1$
24.         continue
25.     else
26.         $i \leftarrow i+1$
27.         continue
28.     end if
29.     Update running time $\delta t$
30. end while
return $(L)$
```

Figure 8: *PLANNER_II* Algorithm

In Figure 8, procedure *GetCandidatePlan* is responsible for generating partially instantiated plans P_i^{II} from the search space. *ValidateSequence* checks if the transitions between the trimmed flight segments are feasible, while *PlanConstraint* eliminates the candidate plans in which the final segment is not descending flight. At step 15, *GetDurations* determines duration $\Delta t_{i,j}^{II}$ of each trimmed flight segment by performing a numerical optimization to minimize cost function

$$J_{plan} = \left\| F_0^{II} \prod_{j=1}^{N^{II}} G_{j-1,j}^{II} G((V_T^*)_j^{II}, (\dot{h}^*)_j^{II}, (\dot{\psi}^*)_j^{II}, \Delta t_{i,j}^{II}) - F_{des} \right\|_2 \tag{44}$$

over variables $\Delta t_{i,j}^{II_i}$. Eq. (34) computes the terminal flight path configuration by propagating initial flight path configuration F_0^{II} along the trajectory defined by trim sequence $\prod_{j=1}^{N^{II}} G_{j-1,j}^{II} G((V_T^*)_j^{II}, (\dot{h}^*)_j^{II}, (\dot{\psi}^*)_j^{II}, \Delta t_{i,j}^{II})$, and the matrix F_{des} defines the desired terminal flight path configuration. Therefore, cost J_{plan} includes the magnitude of terminal position and heading error. A Nelder-Mead simplex optimization algorithm minimizes J_{plan} in Eq. (44). When (44) is minimized, optimal solutions $\{\Delta t_{i,j}^{II}\}_{j=1}^{N^{II}}$ then defines a complete plan P_i^{II}. The complete plan P_i^{II} is considered an acceptable plan if the associated cost J_{plan}^* is less than threshold ε_{plan}, a small positive scalar. Furthermore, plan P_i^{II} will only be considered a feasible plan when the procedure *CheckAltitude* verifies the planned terminal aircraft position has altitude at or above the landing runway's altitude. Length l_i^{II} of feasible plan P_i^{II} is defined as the duration sum

$$l_i^{II} = \sum_{j=1}^{N^{II}} \Delta t_{i,j}^{II} \tag{45}$$

and is used by procedure *SortList* to insert plan P_i^{II} into the sorted plan list *L*, the top-ranked of which is selected as the best solution. Step 29 updates execution time δt, then a new candidate plan P_{i+1}^{II} will be examined with iteration proceeding until the solution space is exhaustively explored or available planning time expires.

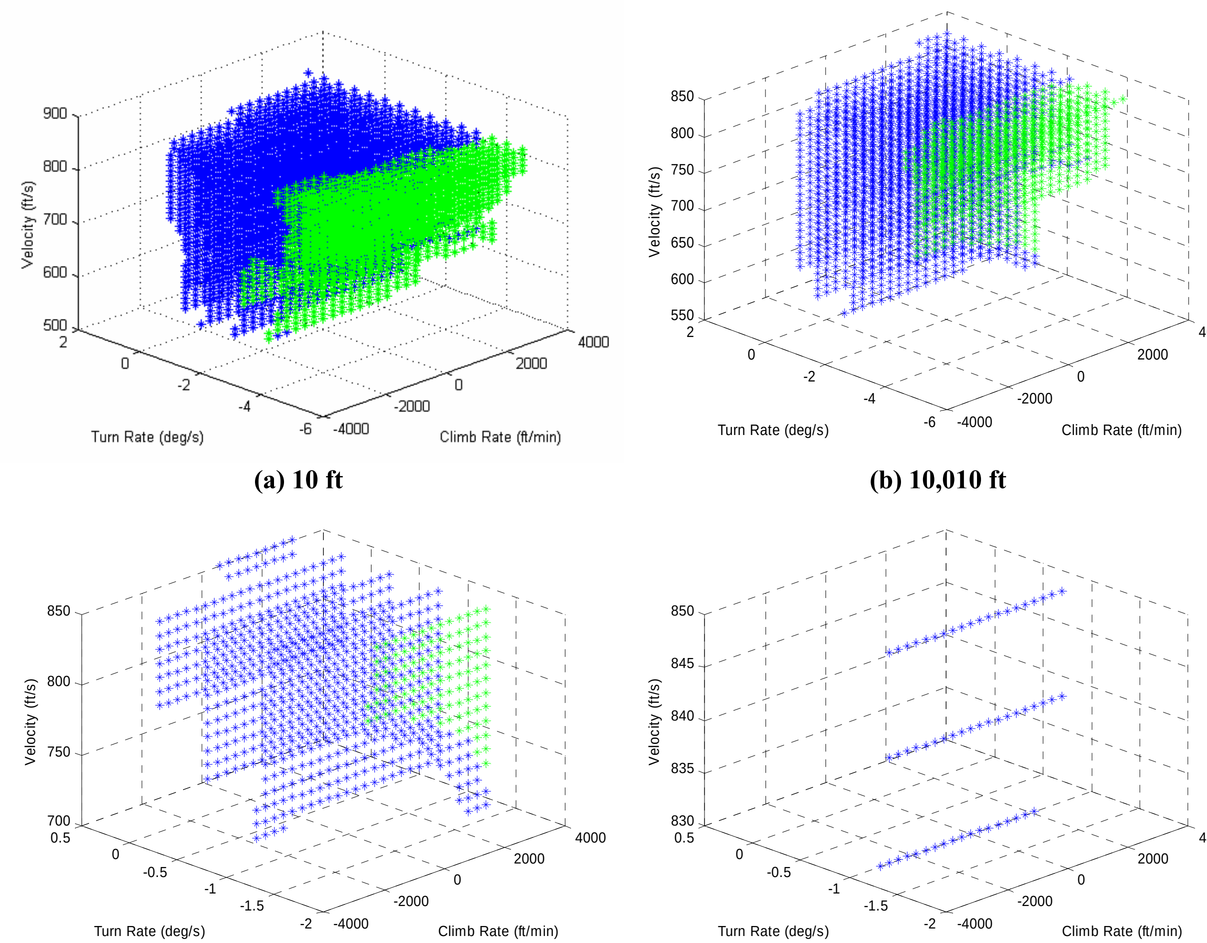
In this application, solutions are deemed acceptable if position and heading errors at the desired landing site are smaller than ε_{plan}. While this method can reduce the planned touchdown errors by eliminating unacceptable Nelder-Mead solutions, these errors can also be reduced during optimization by increasing the error weighting factors since position and heading errors are weighted in Eq. (44). Of course, reducing the planned touchdown errors by weighting these errors more heavily requires more computation time for each optimization. Although the Nelder-Mead simplex algorithm is computationally efficient, the accumulated computation time over all candidate plans is considerable given the size of the search space. Thus, there is a tradeoff between reducing the planned touchdown errors and reducing the computation time, as well as the number of plans with acceptable touchdown errors, when choosing design parameters, the touchdown error weighting factors and acceptable plan criterion ε_{plan}. This tradeoff is left for future work, with the remaining discussion focused on the existence of feasible solutions given a set of design parameters. For the following case study, threshold ε_{plan} is set to 1, each component of landing position error is evenly weighted by 1, and the landing heading error is weighted by 1000. Thus, the maximum acceptable 1-D landing position error is 1 ft, while the maximum landing heading error is 0.001 radians (approximately 0.06 deg).

VI. Case Study

The adaptive flight planner was applied to a damaged General Transport Model (GTM) aircraft with missing left wingtip for this case study. Initially, the full trim database was calculated to specify the reduced flight envelope of the damaged aircraft. This database was then contracted to facilitate real-time trajectory planning. A series of emergency scenarios show the adaptability of the adaptive flight planning approach to the different initial aircraft states when the wingtip damage occurs. To provide context for execution time statistics, this case study was performed on a 2.20GHz AMD® Athlon® 64 processor. The full trim database was created by characterizing trim states over a discrete set of trimmed steady climbing-turning flight conditions, of which each flight condition is defined by a combination of values from Table 1. The full trim database of the damaged GTM aircraft is shown in Figure 10.

Table 1: Trim Flight Parameter Ranges for Constructing the Trim Database

Variable	*Minimum Value*	*Maximum Value*	*Step Size*	*# of Pts*
h (ft)	10	30,010	10,000	4
V_T (ft/s)	450	850	20	21
$\dot{h}(ft/s)$	-50	50	10	11
$\dot{\psi}(\deg/s)$	-6	2	0.5	17
Total trimmed flight conditions				15,708



(a) 10 ft

(b) 10,010 ft

(c) 20,010 ft

(d) 30,010 ft

Figure 9: Trim States of the Left Wing Damaged GTM Aircraft

In Figure 9, the color at each point again indicates trim state characteristics, with green points representing stable and controllable trim states, while blue points indicate unstable but controllable trim states. The missing left wingtip has a significant impact on the flight envelope of the GTM. The aircraft cannot achieve trimmed flight with airspeed lower than 520 ft/sec since relatively high airspeeds are necessary to compensate for the lift loss due to the decrease in left wing area. While the aircraft can achieve a variety of trimmed left turn rates, it can only slightly turn right with the help of a controller to maintain closed-loop stability. This behavior is expected since the smaller lift on the left wing due to the decrease of left wing area causes a negative rolling moment. To counter this negative rolling moment requires a negative aileron deflection which subsequently incurs the negative yawing moment. As a result, left turning flight is easier to trim than right turns. Compared to the limited turning capability, the aircraft has a wide range of climb rates, enabling the damaged aircraft to straightforwardly change altitude as required for landing. As the altitude increases, the flight envelope contracts in all three dimensions: higher airspeeds are required to trim the aircraft, while the trimmed turning and climbing capabilities are more limited. This contraction is more obvious as the altitude increases to 20,010 ft and 30,010 ft. At an altitude of 30,010 ft, the few feasible trim points as shown are not sufficient to define a meaningful flight envelope for the damaged aircraft. For this work, we limit our case study examples to initial altitudes at 15,200 ft and below. However, in practice it would be possible to utilize these few trim states to descend to a lower altitude where the trim database becomes more substantial, since descending flight is possible at 30,010 ft. Based on the full trim database, a reduced trim database $\tilde{D}$ is then defined. Table 2 shows the flight condition values manually chosen as $\tilde{D}$ for the damaged GTM, representing values that can be trimmed up to altitudes of 15,200 ft.

Table 2: Definition of $\tilde{D}$ for the Damaged GTM Aircraft

Airspeed (ft/s)	Climb Rate (ft/s)	Turn Rate (deg/s)	No. Pts.
750	0, ±5	-2.5, -1, -0.5, 0, 0.5	15

Given reduced trim database $\tilde{D}$, the corresponding transition database $\tilde{M}$ is computed for every thousand feet of altitude between 200 and 15,200 using the LQR controller with transition time $\Delta t = 15$ sec and coast time $t_c = 20$ sec. Transitions involving turn rate $\dot{\psi}$ transitions $-2.5 \leftrightarrow 0.5$ are infeasible for altitudes over 6,200 ft since the aileron deflection transients are out of the deflection limits over these transitions. All other transitions are feasible and their kinematic effects over interval $(t_c + \Delta t)$ are stored in $\tilde{M}$. A series of two emergency scenarios were then used to examine the adaptive flight planning approach in this work. In the first scenario, the GTM is over the San Francisco Bay area when the left wing is damaged. Specifically, the initial aircraft's latitude and longitude are 37.44º N and 122.12º W respectively; the initial altitude is 200 ft MSL (mean sea level) and initial heading is 90º, due East. In the LSS, the minimum runway length and width for a feasible landing runway are set to 10,000ft and 200ft respectively since the damaged aircraft must touch down at the high airspeed of 750 ft/s. The runway utility weighting factors defined in Eq. (7) are thus used. The initial footprint radius is set to 20 nautical miles, within which the LSS finds eight feasible runways and ranks them as shown in Table 3. The entire LSS procedure executes in 0.177 seconds, which includes the time for data logging not required in a deployed LSS system.

Table 3: Feasible Landing Sites for Scenario 1

Rank	Airport	Runway	Utility
1	SFO	28R	0.9917
2	SFO	28L	0.9489
3	SFO	10L	0.9017
4	SFO	10R	0.8589
5	OAK	29	0.8131
6	SJC	30L	0.8107
7	SJC	12R	0.7667
8	OAK	11	0.7654

SFO/28R is selected by the AFP as the desired landing site for the damaged GTM aircraft for this case. The runway SFO/28R is located at 37.619002º N and 122.374843º W; its elevation is 11ft and the runway heading is 0.48692 radians. The trajectory planner generates the segmented landing flight plan shown in Table 4.

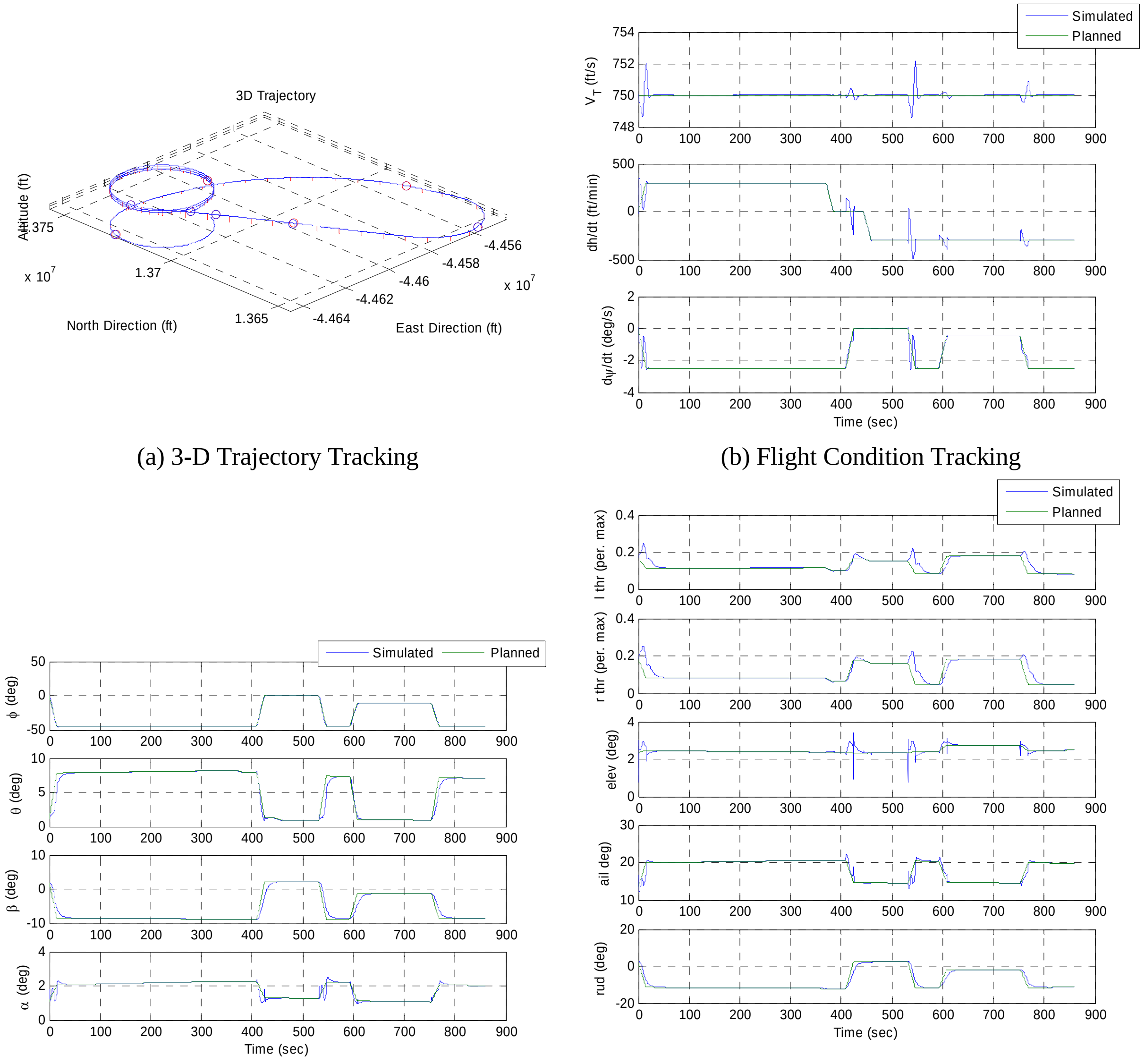
Table 4: Optimal Flight Plan for Scenario 2

Trajectory	*i*	$(V_T^*)_i$ (ft/sec)	$\dot{h}_i^*$ (ft/min)	$\dot{\psi}_i^*$ (deg/sec)	Δt_i (sec)
Part I	0	750	0	0	0.00
	1	750	300	-2.5	334.700
	2	750	0	-2.5	4.400
	3	750	0	0	0.000
Part II	4	750	-300	0	55.066
	5	750	-300	-2.5	54.386
	6	750	-300	-0.5	69.754
	7	750	-300	-2.5	95.783

Since the aircraft is initially at a very low altitude, the planner *PLANNER_I* generates the landing *trajectory part I* as a spiral-up trajectory requiring 0.028 seconds of computation time. Here, the minimum altitude offset $\underline{h}$ described in the Figure 7 *PLANNER_I* algorithm is 1,000 ft. Note that the duration of the third flight segment, a straight/level trim state, is zero since the initial point of this segment is sufficiently close to the desired landing site. It will take the aircraft approximately 400 seconds to complete *trajectory part I* by following the first three trimmed flight segments. Meanwhile, *PLANNER_II* generates the remaining flight plan, *trajectory part II,* in less than 12 seconds. Therefore, the generation of the entire flight plan meets real-time constraints.

The results from the full path simulation are shown in Figure 10. Figure 10(a) shows the simulated 3-D trajectory which yields final touchdown errors of 60.96 ft South, 8.36 ft West, 0.50 ft in altitude, and -0.0004 deg in headings. In Figure 10(b) and 10(c), tracking errors occurring over the trim transitions converge to zero during the subsequent trimmed flight segments. Figure 10(d) shows the control effort required for the aircraft to follow the planned trajectory. All control values satisfy saturation constraints, although the significant transients during trim transitions may require unrealistically high-speed response from the actuators.

In the second scenario, the GTM aircraft is over a remote area when the damage occurs. The initial aircraft location is at 40.89º N and 94.01º W, which is in Iowa. The aircraft's initial altitude is 10,000 ft and initial heading is 210º (South-South-West). Since the damaged GTM aircraft is initially located in a remote area, identification of a feasible runway requires adjustment to the LSS parameters. The minimum runway width requirement is relaxed to be 150 ft. Other requirements are the same as for the previous scenario. For the same reason, the initial footprint radius is manually set to be 50 nautical miles instead of the 20 nautical miles used in the previous scenario, and the radius increment is also 100 nautical miles. Although 82 reachable runways are found within the initial footprint, none of them meet the minimum feasible runway requirements. Therefore, the LSS increases the footprint radius to 150 nautical miles, and then finds 1028 reachable airport runways, out of which 6 feasible runways are identified Based on the same runway utilities weighting factors used previously, these feasible runways are sorted and ranked as shown in Table 5. The LSS procedure is completed in 0.383 seconds. The top ranking runway, OFF/30, is selected by the AFP as the desired landing site for the damaged GTM aircraft. The runway OFF/30 is located at 41.118332º N and 95.912511º W with elevation 1,048 ft and runway heading 0.52 radians. In the trajectory planner, the maximum altitude offset $\bar{h}$ from the *PLANNER_I* algorithm is 2,000 ft**.** The trajectory planner generates an optimal flight plan to runway OFF/30, as shown in Table 6.



(a) 3-D Trajectory Tracking

(b) Flight Condition Tracking

(c) Orientation Information

(d) Controller Information

Figure 10: Full-path Simulation Results of Emergency Landing Trajectory Tracking for Scenario 1

Table 5: Feasible Landing Sites for Scenario 2

Rank	Airport	Runway	Utility
1	OFF	30	0.9199
2	OFF	12	0.8699
3	SZL	19	0.7553
4	SZL	1	0.7452
5	MCI	19R	0.6759
6	MCI	1L	0.6659

Table 6: Optimal Flight Plan for Scenario 2

Trajectory	i	$(V_T^*)_i$ (ft/sec)	$\dot{h}_i^*$ (ft/min)	$\dot{\psi}_i^*$ (deg/sec)	Δt_i (sec)
Part I	0	750	0	0	0.00
	1	750	-300	-1	1362.901
	2	750	0	-1	304.326
	3	750	0	0	880.538
Part II	4	750	-300	-0.5	64.789
	5	750	0	-2.5	15.460
	6	750	-300	-1	168.334
	7	750	-300	-2.5	61.867

The first four flight segments form *trajectory part I*, generated by *PLANNER_I* in 0.000071 seconds. The other four segments comprise *trajectory part II*. As the aircraft executes the flight plan part for trajectory part I, *PLANNER_II* builds the remaining flight segments to complete the plan. While flying *trajectory part I* requires over 2,400 seconds, *PLANNER_II* completes the entire plan in 6.492 seconds. Thus, the real-time requirement is met. Although the 2,400 seconds duration for *trajectory part I* may be too long to be practical for an emergency landing, it results by the artificially-imposed 300 ft/sec descent rate limit consistently selected throughout the case study. Full path simulation results are shown in Figure 11. The simulated 3-D trajectory shown in Figure 11(a) results in final touchdown errors of 41.51 ft South, 3.484 ft East, 0.125 ft in altitude, and -0.0144 deg in heading. Figures 11(b) and 11(c) show the actual trajectory closely tracks the reference trajectory with transients over the trim transitions. Controller responses over the entire trajectory are shown in Figure 11(d). As with the previous example, control inputs fall well within saturation limits while tracking the commands closely.

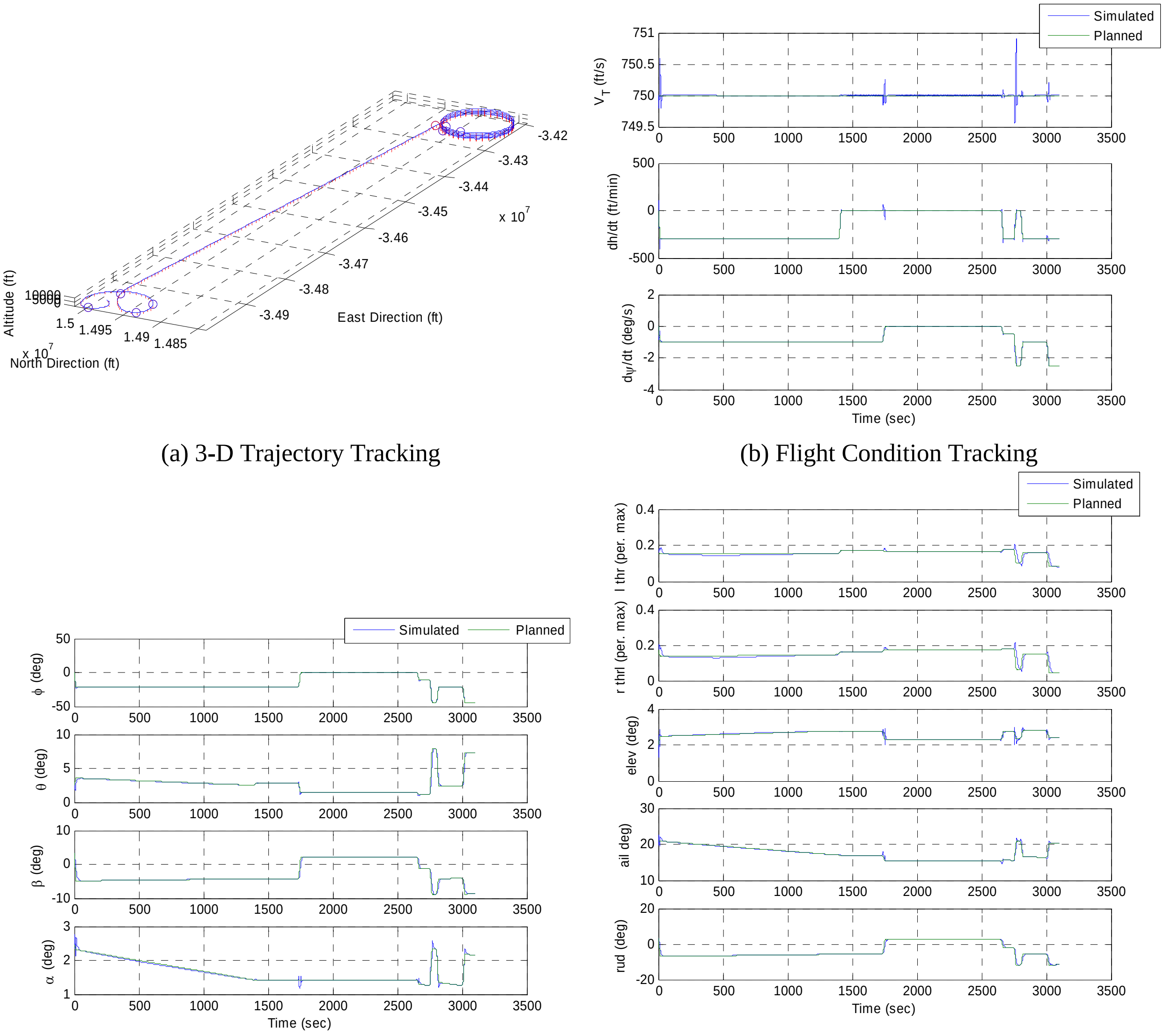
VII. Conclusion and Future Work

This paper has described an Adaptive Flight Planner (AFP) and applied it to a Generalized Transport Model (GTM) aircraft with significant structural damage to its left wing. Each landing trajectory is a sequence of constant-trim segments connected by transitions between trim states. A trim database was developed to define the post-damage flight envelope, and transitions between trim states were characterized in simulation using a controller tuned to handle the post-damage dynamics. A Landing Site Search module identified and ranked nearby runways. A two-step trajectory planner generated a feasible landing flight plan to the top-ranked runway in real-time. An LQR-based PID controller was developed to correctly track trajectory commands over both trimmed flight and transition segments. The success of the adaptive flight planner requires that the aircraft's initial flight condition fall well within the post-damage flight envelope. In the damaged GTM case study, different scenarios were presented to examine the ability of the damaged aircraft to build and successfully execute flight plans in simulation. Typically, the planner was successful and the plan executed accurately. However, the AFP occasionally failed to find a solution when a 30 degree bank constraint was imposed, suggesting future work to augment the current *PLANNER_I* algorithm to a station sufficiently distant from the landing runway to enable shallow turns to final approach.

In this work, the Landing Site Search module must return at least one feasible landing runway before emergency trajectory planning can be performed. It is possible that there are no feasible runways within the reachable region of the post-failure/damage aircraft. Future work is required to enable the LSS to identify a feasible off-runway landing site through use of terrain and population database information not currently available to the AFP. Another issue to be investigated is the initial guidance of the aircraft to a controllable trim state in cases where the aircraft state lies outside the post-failure/damage flight envelope when the failure/damage occurs.

Although the two-step strategy presented enables trajectory planning to be performed in real-time, search heuristics (i.e., for trim state sequence orderings) may be included to more efficiently locate the optimal candidate plan for *trajectory part II* without exhaustive search. Future research is also required to deploy the AFP in practice.

Although emergency flight plans are specified as sequences of trim states, the trim transitions are a necessary part of the plans and thus the trim transition kinematic information is important to emergency trajectory planning. However, the difficulty of tuning a capable linear-system-based controller that allows the post-failure/damage aircraft to accurately follow the trajectory commands over the trim transitions suggests the need to develop a uniform fault-tolerant trajectory tracking control strategy. The complexity of computing the post-failure/damage trim and transition databases prevents their real-time generation. Perhaps the most significant future work required to make the AFP a practical solution is the development of more efficient online techniques for performing the analysis required for the AFP to identify feasible post-failure trim states it can use to plan feasible landing trajectories. This approach would require tight coupling of the AFP with a system identification process, as depicted in the Figure 1 architecture but not yet developed for this work.



(a) 3-D Trajectory Tracking

(b) Flight Condition Tracking

(c) Orientation Information

(d) Controller Information

Figure 11: Full-path Simulation Results of Emergency Landing Trajectory Tracking for Scenario 1

VIII. Acknowledgment

The authors would like to thank Matt Strube and Igor Alonso Portillo for their valuable contributions on the adaptive flight planning technologies that formed the basis of this work. We also thank NASA Ames collaborators Nhan Nguyen, John Kaneshige, and Krishnakumar for providing GTM models and for their valuable feedback. This work was supported in part under NASA Ames Cooperative Agreement NCC21427.

IX. References

[1] T. Grayling, Bishop, S., "Sustainable Aviation 2030: Discussion Document," Institute for Public Policy Research, London, UK 2001.

[2] T. L. Chen, Pritchett, A. R., "On-The-Fly Procedure Development for Flight Re-Planning Following system Failures," *AIAA 38th Aerospace Sciences Meeting and Exhibit,* Reno, NV, 2000.

[3] T. L. Chen, Pritchett, A. R., "Development and Evaluation of a Cockpit Decision Aid for Emergency Trajectory Generation," *Journal of Aircraft*, vol. 38, pp. 935-943, 2001.

[4] T. Fan, Hyams, D. S., Kuchar, J.K., "Study of In-Flight Replanning Decision Aids," *AIAA Guidance Navigation and Control Conference,* Boston, MA, 1998.

[5] B. Schroeder, Sarter, N., "Supporting Decision-making and Action Selection Under Time Pressure and Uncertainty: The Case of In-Flight Icing," *Intl. Symposium on Aviation Psychology,* Columbus, OH, 2001.

[6] J. D. Boskovic, Prasanth, R., Mehra, R. K., "A Multi-layer Autonomous Intelligent Control Architecture for Unmanned Aerial Vehicles," *J. Aerospace Computing, Information, and Communication*, vol. 1, pp. 605-628, 2004.

[7] E. Atkins, Alonso Portillo, I., Strube, M. J., "Emergency Flight Planning Applied to Total Loss of Thrust," *Journal of Aircraft*, AIAA, 43(4):1205-1216, Jul-Aug 2006.

[8] M. Strube, R. Sanner, and E. Atkins, “Dynamic Flight Guidance Recalibration after Actuator Failure,” *Proc. 1st AIAA Intelligent Systems Conference,* Chicago, IL, September 2004.

[9] M. J. Strube, *Post-failure Trajectory Planning From Feasible Trim State Sequences*, Master of Science Thesis, Aerospace Engineering Dept., University of Maryland, College Park, 2005.

[10] F. Ahmed-Zaid, Ioannou, P., Gousman, K., Rooney, R., "Accommodation of Failures in the F-16 Aircraft Using Adaptive Control," in *IEEE Control Systems Magazine*, vol. 11, 1991, pp. 73-78.

[11] M. Bodson, Groszkiewicz, J., "Mutlivariable Adpative Algorithms for Reconfigurable Flight Control," *IEEE Transactions on Control Systems Technology*, vol. 5, pp. 217-229, 1997.

[12] J. D. Boskovic, Mehra, R. K., "A Multiple Model Adaptive Flight Control Scheme for Accommodation of Actuator Failures," *Journal of Guidance, Control, and Dynamics*, vol. 25, pp. 712-724, 2002.

[13] J. D. Boskovic, Mehra, R. K., "Failure Detection, Identification and Reconfiguration in Flight Control," in *Fault Diagnosis and Fault Tolerance for Mechatronic Systems, Recent Advances Series: Springer Tracts in Advanced Robotics*, vol. 1, F. Caccavale, Villani, L., Ed. NY: Springer Verlag, 2002.

[14] J. D. Boskovic, Mehra, R. K., "Intelligent Adaptive Control of a Tailess Advanced Fighter Aircraft Under Wing Damage," *Journal of Guidance, Control, and Dynamics*, vol. 23, pp. 876-884, 2000.

[15] P. Chandler, Pachter, M., Mears, M., "System Identification for Adaptive and Reconfigurable Control," *Journal of Guidance, Control, and Dynamics*, vol. 18, pp. 516-524, 1995.

[16] K. Gundy-Burlet, "Augmentation of an Intelligent Flight Control System for a Simulated C-17 Aircraft," *Journal of Aerospace Computing, Information, and Communication*, vol. 1, pp. 526-542, 2004.

[17] P. Williams, "Selected Flight Test Results for Online Learning Neural Network-Based Flight Control System," *AIAA 1st Intelligent Systems Technical Conference,* Chicago, IL, 2004.

[18] J. C. Latombe, *Robot Motion Planning*. Boston, MA, 1991.

[19] S. M. LaValle, Kuffner, J. J., "Randomized Kinodynamic Planning," *Intl. J. Robotics Res.*, v. 20, 378-400, 2001.

[20] T. H. Cormen, Leiserson, C. E., Rivest, R. L., *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1990.

[21] A. Pongpunwattana, Rysdyk, R., "Real-Time Planning for Multiple Autonomous Vehicles in Dynamic Uncertain Environments," *Journal of Aerospace Computing, Information, and Communication*, vol. 1, 2004.

[22] J. Rubio, Vagners, J., Rysdyk, R., "Adaptive Path Planning for Autonomous UAV Oceanic Search Missions," *AIAA 1st Intelligent Systems Technical Conference,* Chicago, IL, 2004.

[23] N. D. Richards, Sharma, M., Ward, D. G., "A Hybrid A*/Automaton Approach to On-Line Path Planning with Obstacle Avoidance," *AIAA 1st Intelligent Systems Technical Conference,* Chicago, IL, 2004.

[24] F. Vormer, Mulder, M., Paassen, M. Mulder, J., "Design and Preliminary Evaluation of a Segment-based Routing Methodology," *AIAA Guidance, Navigation, and Control Conference,* Monterey, CA, 2002.

[25] E. Frazzoli, "Maneuver-based Motion Planning and Coordination for Single and Multiple UAV's," *AIAA's 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles,* Portsmouth, VA, 2002.

[26] D. McRuer, Ashkenas, I., Graham, D., *Aircraft Dynamics and Automatic Control*: Princeton Univ. Press, 1973.

[27] B. L. Stevens, Lewis, F. L., *Aircraft Control and Simulation*: John Wiley & Sons, Inc., 1992.

[28] S. R. Buss, *3D Computer Graphics: A Mathematical Introduction with OpenGL*. Cambridge Univ. Press, 2003.

[29] J. Roskam, *Airplane Design IV: Layout Design of Landing Gear & Systems*. Lawrence, KS: DARcorporation, 2000.

[30] Khalil, *Nonlinear Systems*, 3rd ed: Prentice Hall, 2002.

[31] C. T. Chen, *Linear System Theory and Design*, 3rd ed. Oxford: Oxford, 1999.

[32] P. Dorato, Abdallah, C. T., Cerone, V., *Linear Quadratic Control: An Introduction*: Prentice Hall, 1998.