Journal of Aircraft
Vol. 52, No. 1, January–February 2015

Trim State Discovery with Physical Constraints

Guoxing Yi[*] and Jiapeng Zhong[†]
Harbin Institute of Technology, 150001 Harbin, People's Republic of China
Ella M. Atkins[‡]
University of Michigan, Ann Arbor, Michigan 48105
and
Changhong Wang[§]
Harbin Institute of Technology, 150001 Harbin, People's Republic of China

DOI: 10.2514/1.C032619

Failure or damage events that degrade performance pose significant risk to aircraft in flight. Adaptive control and system identification may stabilize a damaged aircraft, but identified models may be valid only near each local operating point. This paper presents a novel guidance strategy designed to discover a set of feasible flight states sufficient to enable a safe landing given an unknown degradation event. Rather than persistently exciting the aircraft with the potential to cause further damage, the aircraft may instead be progressively guided through a sequence of trim states that are stabilizable given local envelope estimates. The proposed guidance strategy progressively explores trim state space rather than three-dimensional physical space, identifying a set of low-speed gentle-descent trim states appropriate for landing approach. A potential field method is adapted to steer exploration through trim state space, modeling envelope constraint boundaries as obstacles and desirable approach trim states as attractors. Identified paths through trim state space may present physical-space conflicts such as flight into terrain. To cope, the family of trajectories emanating from the current trim state is propagated in physical space to identify and eliminate physical conflicts. F-16 aileron and rudder jam performance degradation scenarios are presented, showing that the trim state discovery algorithm can effectively explore trim state space without violating vehicle performance or environmental constraints.

Nomenclature

C	=	physical-space constraints
$\boldsymbol{D}_{\mathrm{esc}}$	=	escape direction (three-dimensional)
d	=	prediction search depth
d_x, d_y, d_h	=	distances from current position to the nearest physical-space constraint (longitude, latitude, altitude, respectively), ft
$\boldsymbol{F}_A$	=	attractive force (unitless)
$\boldsymbol{F}_P$	=	repulsive force (unitless)
$\boldsymbol{F}_{\mathrm{pc}}$	=	additional physical repulsive potential force
$\boldsymbol{F}_R$	=	resultant force (unitless)
k_f, k_s	=	step adjustable coefficients
k_p	=	repulsive force weighting coefficient
k_α	=	attractive force weighting coefficient
p, q, r	=	body angle rates (roll rate, pitch rate, yaw rate, respectively), deg/s
pc	=	physical constraint confliction parameter
r_{sc}	=	potential danger tolerance coefficient
r_x, r_y	=	trim state-space exploration circle radii (turn rate, flight-path angle dimensions, respectively)
s, s_t	=	$(V_T \quad \dot{\psi} \quad \gamma)$, trim state (at time t)
s_{app}	=	final approach trim state
s_0	=	initial trim state
T	=	flight trajectory
t_t	=	transition time, s
V_T	=	flight velocity, ft/s
x, y, z	=	aircraft position, ft
x_{sd}, y_{sd}, h_{sd}	=	safety distances (longitude, latitude, altitude, respectively), ft
$\boldsymbol{z}$	=	$[V_T \quad \alpha \quad \beta \quad p \quad q \quad r \quad x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T$, state vector
α	=	attack angle, deg
β	=	sideslip angle, deg
γ	=	flight-path angle, deg
$\boldsymbol{\eta}$	=	$[x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T$, configuration vector
$\boldsymbol{\mu}$	=	$[\mu_t \quad \mu_e \quad \mu_a \quad \mu_r]^T$, control vector
μ_e	=	elevator deflection, deg
μ_r	=	rudder deflection, deg
μ_t	=	thrust setting $\in[0, 1]$
μ_α	=	aileron deflection, deg
$\boldsymbol{\nu}$	=	$[V_T \quad \alpha \quad \beta \quad p \quad q \quad r]^T$, velocity vector
ϕ, θ, ψ	=	Euler angles (roll, pitch, yaw, respectively), deg
$\dot{\psi}$	=	turn rate, deg/s

I. Introduction

FLIGHT safety is of paramount concern to aircraft operators, flight crews, manufacturers, and passengers. Modern aircraft are safe, including standards requiring commercial aircraft to have triple redundancy in primary sensing and flight control systems. Although rare, off-nominal events can still occur that reduce aircraft performance to levels where safety can be compromised. Such events can be induced by factors including flight crew errors and degradation of the environment, e.g., bad weather or nearby traffic, degradation/failure of vehicle components or subsystems, or structural damage. The most catastrophic situations often involve multiple such factors. Technologies such as adaptive control are emerging to improve safety in the presence of these conditions. Although adoption has been slow due to issues with certification and perceived benefit to the industry, research and flight demonstrations continue to provide evidence such "intelligent" automation can reduce the risk of loss of control.

Figure 1 presents statistical accident data per year based on commercial transport operations [1]. Statistics indicate many accidents are related to error in pilot judgment and skill, although in many documented and undocumented cases, pilots also save disabled aircraft through appropriate skill and judgment. Still, pilot perception

Received 11 September 2013; revision received 22 March 2014; accepted for publication 22 March 2014; published online 19 June 2014.

[*] Associate Professor, Control Science and Engineering Department.

[†] Graduate Student, Control Science and Engineering Department.

[‡] Associate Professor, Aerospace Engineering. Associate Fellow AIAA.

[§] Professor, Control Science and Engineering Department.

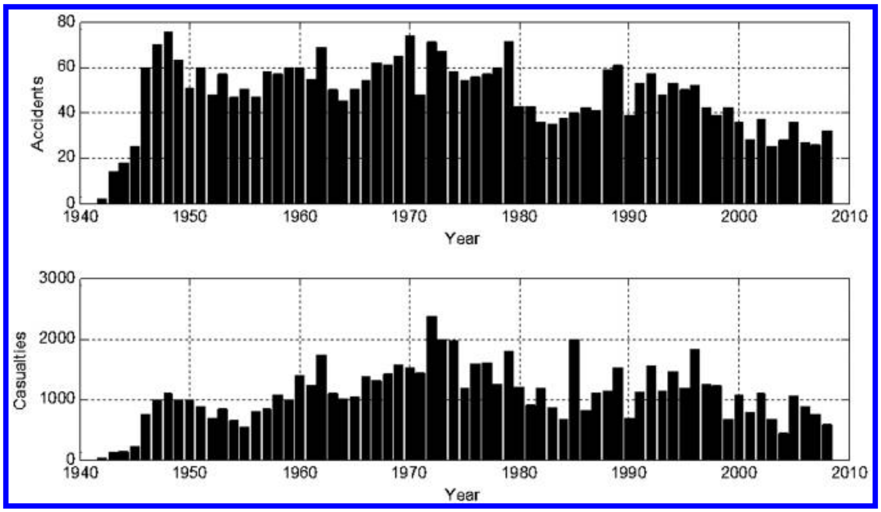

Fig. 1 Fatal airliner hull-loss accidents [1].

in the cockpit is limited, making it difficult to recognize and recover from adverse conditions, particularly under conditions of overload, stress, or fatigue. Failure/damage events, such as loss of thrust, icing, control surface failure, or structural damage, require fast and appropriate response. Flight crew ability to control and guide a disabled aircraft to a safe landing can appreciably benefit from flight simulator training in possible envisioned scenarios [2,3].

Adaptive control methods have been devised to avoid loss of control by maintaining stability and controllability despite sudden or incremental changes in performance parameters [4–6]. Minor degradation can be managed seamlessly though increased thrust or larger actuator deflections to compensate. As saturation limits are reached, however, reference commands may no longer be possible, necessitating alteration of the flight plan in addition to the control law. Aircraft dynamics are nonlinear. It is therefore difficult to extend stability and controllability results from a local identification process in a manner that characterizes the full operating envelope. In other words, although an adaptive controller can stabilize the aircraft, it typically cannot inform the crew or autopilot whether the sequence of states they prefer for landing is feasible or not without further exploration of the flight states comprising that landing flight plan. Exploration of the flight envelope must be done carefully rather than with generalized and potentially large-magnitude accelerations to minimize the potential for loss of control, structural damage to the airframe, and trauma to passengers.

Faced with in-flight damage or failure(s), pilots must be aware of flight envelope constraints, particularly when these bounds have reduced the set of possible safe landing trajectory options. To "discover" these new constraints, time aloft constraints permitting, flight crews may attempt "trial runs" of approach or maneuver sequences at an altitude believed sufficient to support recovery. The goal of such crew-initiated behaviors is not to "fly to the next waypoint" or initiate an immediate emergency approach to a landing site but instead to maneuver in a manner that allows the crew to better understand the new performance limits and necessary changes to their control strategy. Such a "discovery" process was an important part of the recovery and emergency landing sequence successfully executed in 2003 by a DHL Airbus A300 flight crew after the left wing of their aircraft was struck by a missile just after takeoff from Baghdad International Airport.

This paper presents a novel trim state discovery (TSD) guidance algorithm inspired by efforts such as those of the DHL A300 crew previously noted. The objective of TSD is to guide the aircraft through a sequence of feasible trim states sufficient to support a safe landing without losing control or further damaging the aircraft. For trim state discovery, the notion of "optimality" is redefined in a context-appropriate manner. The primary concern is no longer a minimum-time or minimum-fuel trajectory to some destination waypoint. Instead, the priority is on understanding new performance constraints. Optimality is therefore defined in terms of exploration of the state space to enable a safe landing. This work focuses on optimizing exploration of a state space consistent with aircraft trim or equilibrium state specification, including airspeed, flight-path angle, and turn rate for a conventional fixed-wing aircraft. Maneuvers through this state space must be planned in a manner that avoids excursion outside the reduced flight envelope and that changes the reference state sufficiently slowly to avoid exposing the occupants and airframe to large accelerations. While the goal of trim state discovery is to efficiently explore the space of achievable and useful flight states in real time, this path must also respect physical constraints, typically translating to avoiding terrain and low-altitude overflight of populous areas, given that a damaged aircraft might also pose risk to people on the ground (e.g., dropping structural components, releasing fuel, and increased risk of loss of control, should the damage or failure conditions progress to a more critical state).

In the context of an emergency flight management capability, the goal of trim state discovery is to sufficiently inform the flight planning, guidance, and control of reduced performance characteristics. Researchers have provided insight into crew response to off-nominal situations through studies of cockpit management strategies and workload [7]. To help a flight crew make an appropriate sequence of decisions, researchers such as Chen and Pritchett have proposed an emergency flight planner that depends on predetermined plans and degraded models; they then examined its utility through pilot studies [8]. Emergency flight management architectures more focused on autonomous recovery have also been proposed, as illustrated in Fig. 2 [9,10]. In this design, a flight plan monitor continually validates the existing flight plan against the most current system model to verify feasibility of the flight plan. If the executing flight plan becomes infeasible, the pilot is notified. Concurrently, an adaptive flight planner (AFP) is activated to generate a new flight plan, requiring identification of safe landing site(s) followed by construction of a dynamically feasible trajectory to the chosen landing site. Other researchers have proposed similar adaptive planning and control concepts: for example, a NASA architecture focused on avoiding loss of control [11]. Researchers have also proposed emergency landing planner algorithms [12,13] that generate optimal landing sequences presuming accurate knowledge of the updated set of feasible trim states as well as other decision-making information, including weather, wind, terrain, and airport characteristics [12].

Previous work has proposed assembling trim state data in a database indexed by specific damage or failure characteristics [13]. This database of stabilizable trimmed flight states describes the feasible postfailure flight envelope, capturing in a reduced kinematic form the performance characteristics of the aircraft after a specific failure. The database not only indicates the feasibility of trim state but also captures the dynamic characteristics of transition from one trim state to another [13]. For failure cases that can be modeled a priori (e.g., control surface jams, loss of thrust), such databases can be generated a priori, and then used online to evaluate the stability and

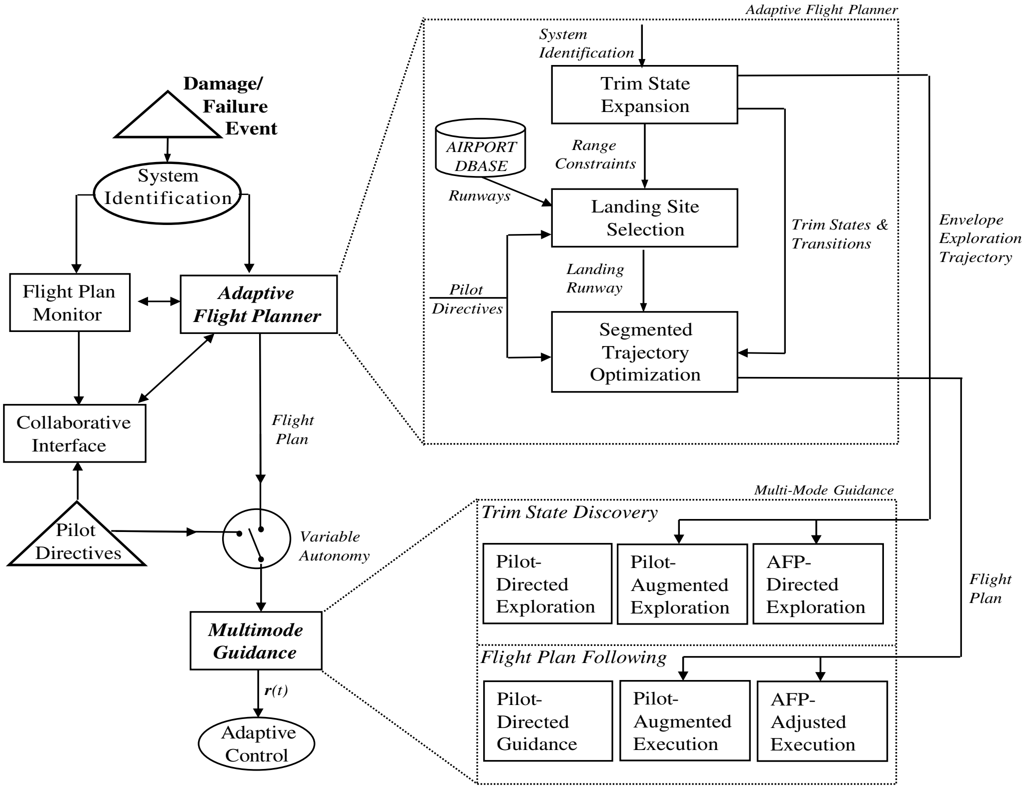

Fig. 2 Emergency flight management architecture.

controllability of each trimmed or equilibrium flight state. However, many damage and failure cases that might occur in flight are unknown or not able to be modeled, e.g., structural damage. In such cases, performance must be characterized online rather than via an existing performance database. The aerodynamic characteristics of a damaged aircraft can be considered as an uncertain nonlinear system, with the stabilizable trim state as the equilibrium point.

Figure 3 illustrates the process of adaptive flight planning, including the trim state discovery system module defined in this paper. As shown in Fig. 3, when damage or failure events happen, the adaptive flight planner determines whether the failure is known. If so, the aircraft dynamic model links to the appropriate model (e.g., from a trim database) and continues, selecting a landing site and generating a feasible trajectory to that site. If specifics of the damage or failure are not known, the aircraft dynamic model must be determined online, potentially through the trim state discovery process proposed in this work. To ensure the aircraft remains stabilizable, at each operating point, the set of nearby trim states is examined to determine which will be safe. So long as these states are sufficiently close to the current state for the model estimate to be a good approximation of behavior in that state, any nearby state predicted to be feasible will be safe to command. As a baseline for TSD, the system must identify feasible left turn, straight, and right turn trim states, in a gentle descent and for level flight as a minimum; climbing flight is also useful to ensure go-around or terrain avoidance will be possible on approach as needed.

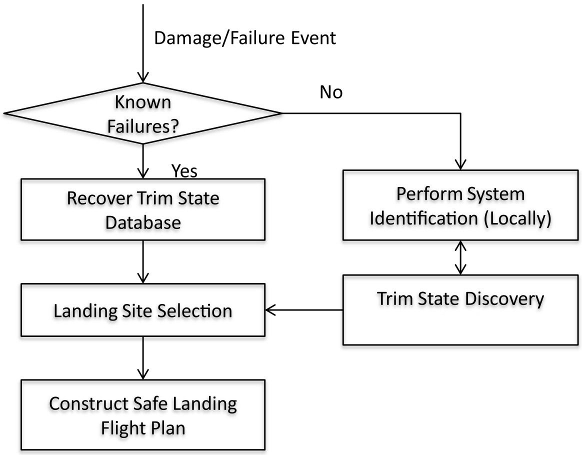

Fig. 3 Adaptive flight planner architecture.

As shown in Fig. 3, TSD must iteratively interact with a system identification module to ensure parameter updates enable envelope constraints to be accurately predicted as the aircraft approaches each local envelope boundary during the TSD exploration process. Once TSD concludes, trim states that have been tested in flight can be sequenced into the emergency landing flight plan. Methods in system identification have been investigated to estimate unknown performance characteristics [14–18]. TSD explores and exploits the feasible trim state space of the damaged aircraft. The goal of this iterative process shown in Fig. 3 is to identify a sufficient but not necessarily comprehensive set of trim states and transitions to enable construction of a feasible emergency landing flight plan.

Because TSD must execute online in response to parameters specified by system identification, adapt reactive motion planning is required to guide the exploration process. In traditional robot motion planning with an unknown physical environment, onboard sensors characterize local obstacle fields and targets, enabling a motion planner to incrementally build paths through this potentially complex environment. To adapt motion planning to TSD, we map the “physical space” traditionally assumed for motion planners to a novel “trim state space” with coordinate system dimensions (climb rate, turn rate, and airspeed) replacing physical-space coordinates (x, y, z), and with flight envelope boundaries represented as impenetrable “obstacles” analogous to the physical obstacles sensed and avoided by traditional robot motion planners. With this translation, we introduce an automated guidance protocol capable of conservatively targeting trim states to sequence for a safe landing.

Next, we describe aircraft model preliminaries, followed by definitions of progressively sophisticated algorithms for trim state discovery. An artificial potential field path-planning [19–24] method was adopted to guide the TSD process, augmented with an edge-following algorithm to avoid being trapped in local minima. An initial two-dimensional (2-D) formulation over turn rate and climb rate is extended to a three-dimensional (3-D) algorithm that also includes airspeed. Extensions to the 3-D algorithm are proposed to avoid hard conflicts in physical space. A well-understood F-16 aileron jam failure case study demonstrates effectiveness of the TSD approach and differences between 2-D and 3-D TSD for postfailure flight planning.

II. Trim States

The six-degree-of-freedom nonlinear aircraft equations of motion are given by

$$\boldsymbol{f}(\dot{z}, z, \boldsymbol{\mu}) = 0 \tag{1}$$

where $\boldsymbol{f}$ is a vector of n scalar nonlinear functions, and $\boldsymbol{z}$ is the 12-dimensional state vector including aircraft position $[x \quad y \quad z]$, attitude $[\phi \quad \theta \quad \psi]$, angular velocities $[p \quad q \quad r]$, and body linear velocity relative to the wind frame $[V_T \quad \alpha \quad \beta]$:

$$\boldsymbol{z} = [V_T \quad \alpha \quad \beta \quad p \quad q \quad r \quad x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T \tag{2}$$

Control vector $\boldsymbol{\mu}$ can be expressed as

$$\boldsymbol{\mu} = [\mu_t \quad \mu_e \quad \mu_a \quad \mu_r]^T \tag{3}$$

where subscripts $t =$ throttle, $e =$ elevator, $a =$ aileron, and $r =$ rudder.

The vector $\boldsymbol{z}$ can be divided into two parts: aircraft configuration $\boldsymbol{\eta} = [x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T$ and velocity $\boldsymbol{v} = [V_T \quad \alpha \quad \beta \quad p \quad q \quad r]^T$.

A trim state is a steady-state flight condition in which linear and angular aircraft velocities remain constant over time. We denote trim flight states with superscript $*$. In a trim flight condition,

$$(\dot{V}_T, \dot{\alpha}, \dot{\beta}) \equiv 0, \quad (\dot{p}, \dot{q}, \dot{r}) \equiv 0, \quad (\dot{\phi}, \dot{\theta}) \equiv 0, \quad \dot{\psi} \equiv \dot{\psi}^*, \quad \dot{h} \equiv \dot{h}^* \tag{4}$$

represents the ability of an aircraft to turn at constant rate $\dot{\psi}^*$ and climb at constant rate $\dot{h}^*$ in steady state. A trim state therefore represents a constant airspeed V_T^*, turn rate $\dot{\psi}^*$, and climb rate $\dot{h}^*$ condition valid for a certain altitude (interval). For trajectory planning,

$$\bar{z} = [V_T \quad \alpha \quad \beta \quad p \quad q \quad r \quad \phi \quad \theta]^T \tag{5}$$

To determine a trim state, a cost function is defined as

$$J_{\text{trim}}(\boldsymbol{z}, \boldsymbol{\mu}) = \frac{1}{2} \dot{\bar{z}}^T Q \dot{\bar{z}} \tag{6}$$

where z^*, corresponding to $J_{\text{trim}}^* = 0$, is a trimmed or equilibrium flight state. Since this problem cannot be solved analytically, numerical optimization is employed, taking into account the dynamic model, e.g., based on stability derivatives. Application of each trim state must satisfy basic kinematics [Eq. (7)] and actuator [Eq. (8)] saturation constraints:

$$\begin{aligned} h &= h^*, \\ v_T &= v_T^*, \\ p &= \dot{\psi}^* \sin\theta, \\ q &= \dot{\psi}^* \cos\theta \sin\phi, \\ r &= \dot{\psi}^* \cos\theta \cos\phi, \\ \tan\theta &= \frac{ab + \sin\psi^* \sqrt{a^2 - \sin\gamma^{*2} + b^2}}{a^2 - \sin\gamma^{*2}} \end{aligned} \tag{7}$$

Where $a = \cos\alpha \cos\beta$, $b = \sin\phi \sin\beta + \cos\phi \sin\alpha \cos\beta$, γ^* is the trimmed flight-path angle, and throttle is commanded in percent [0, 1]:

$$|\mu_t| \leq 1, \quad |\mu_e| \leq 25, \quad |\mu_a| \leq 21.5, \quad |\mu_r| \leq 30 \tag{8}$$

Control surface deflections are in degrees [Eq. (8)]. Since the nonlinear system can be approximated by a linearization of its dynamics about a trim state in a small neighborhood surrounding that trim state [9,12], a linear perturbation method has been used to calculate the Jacobian matrices for each trim state. Therefore, the stability and controllability can be evaluated by system eigenvalues and the controllability matrix, respectively. Figure 4 shows the stable and controllable trim states of an F-16 aircraft at an altitude of 20,000 ft for a case in which the rudder is jammed at a 20 deg deflection. Figure 5 represents trim states of F-16 aircraft at an altitude of 20,000 ft for a case in which the aileron was jammed at 5 deg. In the figures, a green asterisk indicates a naturally stable trim state with trimmed flight condition (V_T, $\dot{h}$, $\dot{\psi}$), whereas a blue asterisk represents an unstable but controllable trim state. The unmarked trim states are infeasible or uncontrollable. Given a full envelope approximation, the trajectory planner has the maximum possible set of steady flight states from which to construct a safe landing plan. However, real-time discovery of this full envelope, due to our ability to only locally approximate it from a particular flight state, may require prohibitively extensive excursion through flight envelope space. The proposed TSD method directly addresses this problem.

III. Trim State Discovery

As previously discussed, system identification provides local estimates of aircraft model parameters enabling prediction of feasible trim states only in the local region around the current operating state. The goals of TSD are then to incrementally explore trim state space and build a trim state database sufficient for emergency flight

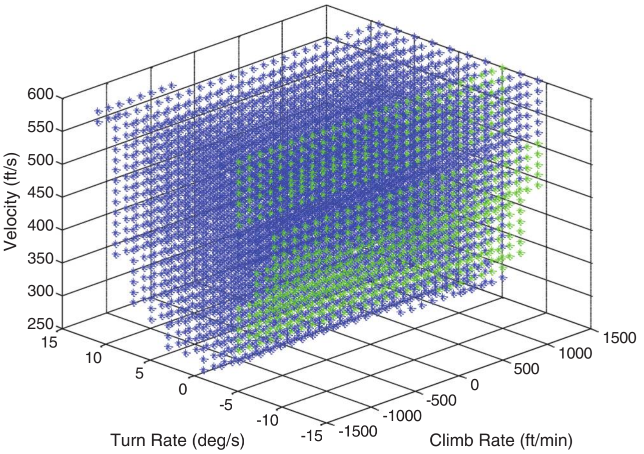

Fig. 4 Feasible trim states of F-16 with jammed rudder [13].

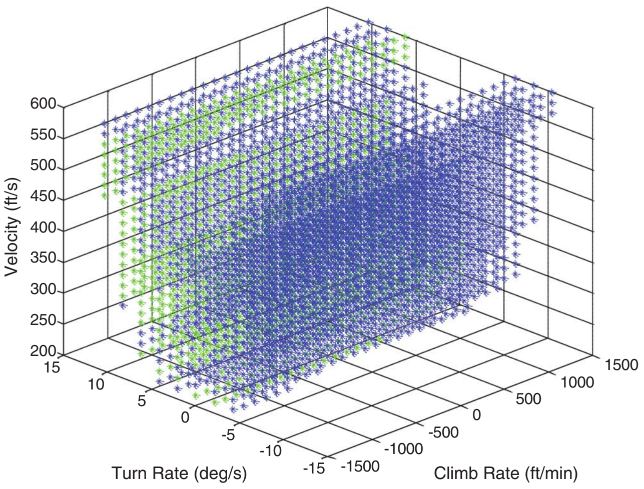

Fig. 5 Feasible trim states of F-16 with jammed aileron [13].

planning, as shown in Fig. 2. Without loss of generality, we presume the aircraft initially establishes and holds a stable initial trim state immediately after the failure/damage occurs; such stabilization could be achieved by a traditional autopilot or adaptive control law. Once the need for TSD is then recognized, either by the crew or by automation (beyond the scope of this manuscript), TSD then initiates its incremental exploration process, terminating once a sufficient trim state database has been identified to support a safe landing, per criteria specified next.

A. Trim State Discovery Definition

The purpose of trim state discovery is to enable the pilot/autopilot to confirm the stability of a nominal final approach trim state sequence (including transitions) with the help of local envelope approximation or to show that the nominal approach is no longer possible and then replan a feasible approach to landing, per Fig. 3. The problem of trim state discovery is then defined as follows:

Given an initial stable position in trim state space $s_0 = (V^*_{T,0}, \dot{\psi}^*_0, \gamma^*_0)^T$ and an ideal final approach trim state $s_{\mathrm{app}} = (\,V^*_{T,\mathrm{app}} \quad \dot{\psi}^*_{\mathrm{app}} \quad \gamma^*_{\mathrm{app}}\,)^T$, 1) generate a sequence of trim states and transitions connecting the initial trim state and final approach trim state in trim state space from s_0 to S_{app}, $P = \{z^*_i, \mu^*_i, \Delta t_i\}^n_{i=1} = \{(V^*_{T,i}, \dot{\psi}^*_i, \gamma^*_i, \mu^*_i, \Delta t_i)^T\}^n_{i=1}$ and 2) for a given sequence P, generate an approximate corresponding physical-space trajectory

$$T = \{w_j\}^m_{j=1} = \{(\mathrm{lon}_i, \mathrm{lat}_i, \mathrm{alt}_i, \psi_i, t)^T\}^m_{j=1}$$

and verify this trajectory is safe, or else modify as needed.

The triplet $(V^*_{T,i}, \dot{\psi}^*_i, \gamma^*_i)^T$ specifies a trimmed flight condition, and all trim states designated as "feasible" must be stabilizable in the presence of disturbances. Figure 6 depicts a path through trim state space from initial state $s_0 = (V^*_{T,0}, \dot{\psi}^*_0, \gamma^*_0)^T$ to final approach trim state $s_{\mathrm{app}} = (\,V^*_{T,\mathrm{app}} \quad \dot{\psi}^*_{\mathrm{app}} \quad \gamma^*_{\mathrm{app}}\,)^T$.

In this illustration, there are multiple flight envelope boundaries that must be circumvented. With the aforementioned formulation, trim state discovery is mapped to motion planning with obstacle avoidance.

B. Two-Dimensional Trim State Discovery

In the TSD process, stability of the aircraft must be consistently maintained. For a case in which failure/damage occurs but the specific cause is undetectable, local flight envelope estimation can approximate a local flight envelope. Before transitioning from the current equilibrium state to a new state, the feasibility of the new state should be predicted by the current local envelope estimation; otherwise, the aircraft may transition outside its operating envelope. In terms of dynamic system analysis, feasibility prediction translates to determining whether the new trim state belongs to the local flight envelope or not, and whether the new stability margin is sufficient for the damaged aircraft with a disturbance such as wind. The problem of determining the local flight envelope for a nonlinear system has received substantial attention in the literature. Some researchers use a Lyapunov function method to estimate the attraction region of a stable equilibrium point in a nonlinear system. Chiang et al. presented a comprehensive theory to derive a complete dynamical characterization of the stability boundary of a large class of nonlinear autonomous dynamical systems and proposed a method for finding the stability region based on its topological properties [16]. Recently, linear matrix inequality (LMI) theory has been introduced to deal with nonconvex distance problems in attraction region estimation. Researchers have obtained a lower bound of the largest estimate of the domain of attraction for a fixed quadratic Lyapunov function via LMI and have provided a condition for checking tightness of a lower bound [17]. Tibken used real algebraic geometry theory to compute subsets of the region of asymptotically stable stationary points of polynomial systems by reformulating the problem as a LMI [18]. Amato et al. proposed a method to investigate the attraction region of equilibrium points of quadratic systems, ascertaining whether a certain box belongs to the attraction region [19].

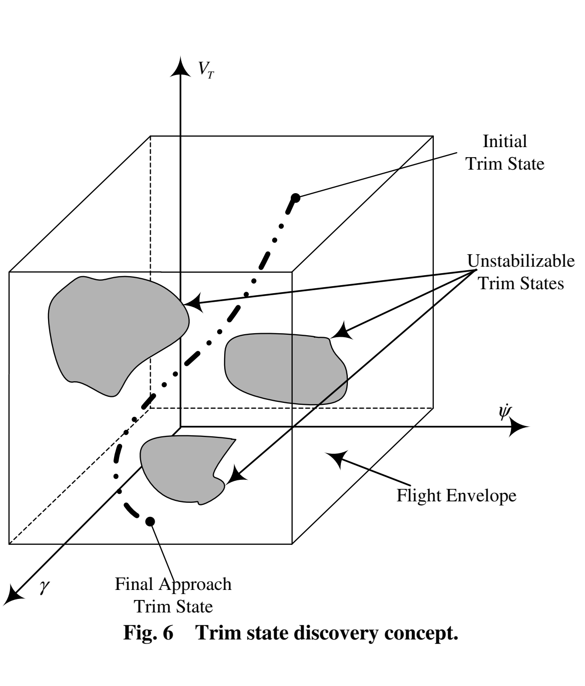

Fig. 6 Trim state discovery concept.

We focus on the process of TSD with the goal of migrating toward and exploring the region around an ideal final approach trim state. We presume the identified dynamics model is at least valid within the neighborhood of the present trim state. The specifics of local flight envelope estimation are outside the scope of this paper but are specifically studied in complementary work, such as that presented by McDonough et al. [25].

Two-dimensional TSD searches transitions through trim state space described by coordinates ($\dot{h}$, $\dot{\psi}$) and bases altitude extrapolation given changing density ρ on the hypothesis that trim states achievable at high altitudes will also be achievable at lower altitudes. This hypothesis is consistent with all failure/damage cases we have studied to date and is intuitive, since control surface effectiveness (force) increases with ρ. This assumption allows us to assume each identified trim state will remain feasible at or below the maximum altitude at which stable operation at this state was demonstrated.

An artificial potential field (APF) path-planning algorithm [19–24] is adopted for incremental TSD exploration due to its ability to incrementally (locally) build motion plans that can adapt to real-time changes in obstacle (envelope boundary) constraints. Let $\boldsymbol{F}_A$ represent the attractive potential force at current trim state s_t due to goal s_{app}, and let the obstacle apply repulsive force $\boldsymbol{F}_P$ on s_t. These two forces will generate resultant force $\boldsymbol{F}_R$. As a physical particle analog, the vehicle is then guided to follow this artificial potential field via a transition in direction $\boldsymbol{F}_R$. For each iteration, step length depends on the magnitude of $\boldsymbol{F}_R$ and the local attraction region, as described next. Figure 7 depicts how the three forces interact. The calculation of APF forces is as follows:

1) For attractive goal force,

$$\boldsymbol{F}_A = k_a\left(1 - \exp\left(\frac{|\boldsymbol{s}_{\mathrm{app}} - \boldsymbol{s}_t|}{|\boldsymbol{s}_{\mathrm{app}} - \boldsymbol{s}_0|}\right)\right)\frac{\boldsymbol{s}_{\mathrm{app}} - \boldsymbol{s}_t}{|\boldsymbol{s}_{\mathrm{app}} - \boldsymbol{s}_t|} \tag{9}$$

where k_a is the adjustable weighting coefficient of the attractive force terms that can be selected by the user. The exponential function used to describe the goal attractor force is consistent with the artificial potential field literature, except that the path is planned through the trim state rather than physical space. The weighting coefficient k_a is chosen based on the level of conservativeness desired in trim state discovery, as described next.

2) First, the centroids of all known obstacles representing in the approximate or nearby flight envelope boundaries must be calculated.

The centroid of the ith detected obstacle in the local flight envelope estimate region (Fig. 7) is defined as

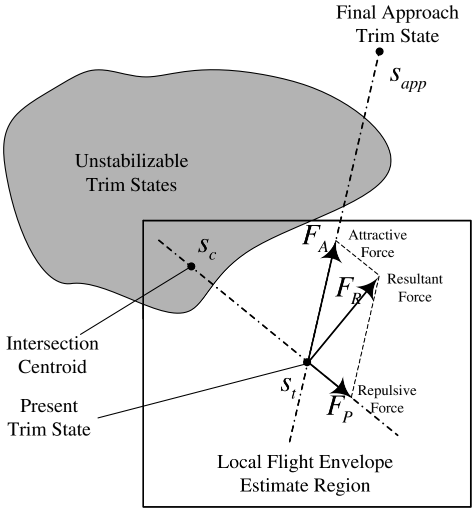

Fig. 7 Potential force model for 2-D TSD.

$$\boldsymbol{s}_c^i = (x_c^i, y_c^i) = \left(\frac{1}{A_o^i}\int_{A_a} x\,\mathrm{d}A^i, \frac{1}{A_o^i}\int_{A_a} y\,\mathrm{d}A^i\right) \tag{10}$$

where $\boldsymbol{s}_c^i$ is position vector of the ith dynamically discovered (explored) obstacle centroid. A_o^i and A_a are the area of the ith explored obstacle and local attraction region, respectively. In practice, we first decompose a local flight envelope area to explore with an $N \times N$ grid. For our case studies, we set $N = 20$, providing 400 possible exploration elements. The ith obstacle centroid $\boldsymbol{s}_c^i$ is then specified as

$$\boldsymbol{s}_c^i = (x_c^i, y_c^i) = \left(\frac{1}{m^i}\sum_{j=1}^{m^i} x_j^i, \frac{1}{m^i}\sum_{j=1}^{m^i} y_j^i\right) \tag{11}$$

where m^i is the number of cells in the local envelope region contained in the ith dynamically discovered (explored) obstacle. The geometrical center of the jth cell in the ith dynamically discovered (explored) obstacle is (x_j^i, y_j^i).
Then, the total repulsive force over all detected obstacles is defined by

$$\boldsymbol{F}_P = k_p \sum_{i=1}^{n_o}\left(\frac{A_o^i}{A_a}\right)\cdot\frac{\boldsymbol{s}_t - \boldsymbol{s}_c^i}{|\boldsymbol{s}_t - \boldsymbol{s}_c^i|} \asymp k_p \sum_{i=1}^{n_o}\left(\frac{m^i}{N * N}\right)\cdot\frac{\boldsymbol{s}_t - \boldsymbol{s}_c^i}{|\boldsymbol{s}_t - \boldsymbol{s}_c^i|} \tag{12}$$

where k_p is the adjustable weighting coefficients of the force terms that can be selected by the user, and n_o is the number of detected obstacles in this local region. $\boldsymbol{F}_p$ includes all repulsive forces generated by the n_o obstacles in the local flight envelope estimate region. For each obstacle, the first term scales the repulsive force by the fraction (percentage) of overall space occupied by the obstacle. The second term provides the direction vector to "repel" TSD away from this obstacle.

3) The resultant force over all obstacles and the goal attractor is then defined by

$$\boldsymbol{F}_R = \boldsymbol{F}_P + \boldsymbol{F}_A \tag{13}$$

4) Next, define the next trim state, s_{i+1}:

$$\mathrm{step}_i = k_f \cdot k_s \cdot \frac{\boldsymbol{F}_R}{|\boldsymbol{F}_R|} + (1 - k_f)\mathrm{step}_{i-1} \tag{14}$$

$$s_{i+1} = s_i + \mathrm{step}_i \tag{15}$$

where k_f and k_s are adjustable coefficients of force set by the user.

In the TSD algorithm, attractor force weighting coefficient k_a and repulsive force weighting coefficient k_p are chosen to make sure the attractive goal force and obstacle repulsive force are numerical comparable. If k_a is relatively high, the path will follow a more direct route from initial to goal states, whereas if k_a is relatively low, the trajectory will steer well clear of envelope boundaries, thus following a more circuitous route to the goal.[¶] Note that k_s is chosen to maintain an appropriate TSD step length, and k_f is chosen to provide a smoothing (moving average filter) over the past two steps.

Although computationally tractable and capable of straightforwardly integrating obstacle constraints and goals, the artificial potential field method is sensitive to local minima that typically arise for a concave obstacle or with multiple proximal obstacles that collectively generate a concave field. To cope with the problem of local minima, we adopt an edge-following algorithm [26,27] to explore the obstacle along its boundary until the goal can again be pursued. In trim state space, this corresponds to traversing within but near the local edge of the flight envelope with respect to primary state-space parameter flight-path angle and turn rate. In our APF algorithm, a local minimum is detected by monitoring the speed of state transition. The average speed of the latest 10 transitions will be approximately zero when the system becomes trapped. After the trap is detected, the TSD will transition along the obstacle in a direction along the edge of the obstacle. In edge-following mode, the distance between the edge and the present trim state guides the transitions through trim state space. Once the angle between repulsive force and attractive force is less than η_e, with $\eta_e < 90$ deg, the obstacle is no longer introducing a local minimum and TSD resumes its nominal potential field navigation mode to follow a path to the target trim state region. If TSD finds an infeasible final approach state with the given velocity, TSD will change the airspeed within nominal performance interval $[v_{\min}, v_{\max}]$ to seek further envelope exploration. When the final approach trim state is feasible, the 2-D TSD process is able to guide the aircraft to this state in trim state space. When the neighborhood of the targeted approach state is reached, this neighborhood is explored with grid or step size defined based on system identification constraints to more closely examine the ability of the aircraft to robustly operate in this region of trim state space. If the stability margin is not sufficient, TSD will change some condition, such as path angle, and explore again. The neighborhood is explored following a trim state sequence computed as

$$s_{ep} = \{s_{ep,i}\}_{i=1}^{4} \tag{16}$$

where

$$s_{ep,i} = \left\{s_{\mathrm{app}} + \left(i \cdot r_x \cdot \sin\left(\frac{2\pi}{16}j\right), i \cdot r_y \cdot \cos\left(\frac{2\pi}{16}j\right)\right)\right\}_{j=0}^{16} \tag{17}$$

and r_x and r_y are the user-defined explored circle radii for turn rate (in degrees per second) and path angle (in degrees), respectively. Figure 8 illustrates the explored neighborhood of the approach landing state, with a turn rate step of $r_x = 1.25$ deg /s and a flight-path angle step of $r_y = 0.5$ deg.

C. Three-Dimensional Trim State Discovery

Two-dimensional TSD identified a final trim state strictly by turn rate and path angle with a fixed airspeed to simplify the search space. As shown next, 2-D TSD simulation results illustrate that trim states can be explored more than once. The general TSD problem requires three-dimensional trim space exploration over airspeed, path angle, and turn rate. Note that, although atmospheric density (thus altitude) can also impact the stabilizable trim state space, all cases we have studied to date have a flight envelope that only expands as atmo-

[¶] It is possible that local minima will be induced by choosing low k_a, particularly when multiple obstacles exist, as will be discussed subsequently. An inappropriately high k_a value may instead drive TSD outside the flight envelope (i.e., inside an envelope obstacle). These two exceptions represent the most fundamental problems of using an APF algorithm; the tradeoffs favoring APF are a fast and intuitive algorithm with similarity to how a flight crew might manually explore capabilities and limits of a damaged aircraft.

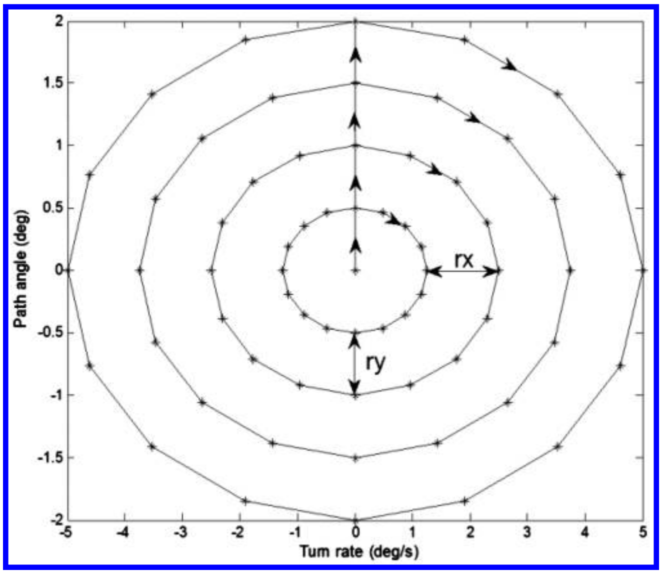

Fig. 8 Trim state neighborhood exploration for 2-D TSD.

spheric density increases. In this work, it is therefore presumed that TSD in a higher-than-approach altitude space with ample clearance from the ground will provide more tolerance for exploration, thus offering a conservative (safe) envelope estimate also valid for the eventual approach.

Figure 9 illustrates the three-dimensional trim state space. The symbols have the same definitions as Fig. 7, with s_c representing centroid of a known obstacle from any previous exploration. Three-dimensional TSD guides the exploration process through changing V_T, $\dot{\psi}$, and γ to s_{app} with obstacle avoidance. With 3-D search, the cyclic exploration problem shown next for two-dimensional search can be resolved. Therefore, the time of TSD can be significantly reduced, which will improve the real-time performance of AFP.

The potential force terms inspired from the robot path-planning literature are also introduced to steer 3-D trim state discovery (3-D TSD) through the extended trim state space. Detected obstacles generate a "repulsive force" $\boldsymbol{F}_P$ to push the path away from the obstacle, and the final approach state will generate an "attractive force" $\boldsymbol{F}_A$. These two forces will generate resultant force $\boldsymbol{F}_R$, which will guide the path finding process. Similar to the 2-D algorithm, the calculation of the forces and the algorithm steps are as follows:

1) Attractive force:

$$\boldsymbol{F}_A = k_a\left(1-\exp\left(\frac{|\boldsymbol{s}_{\mathrm{app}}-\boldsymbol{s}_t|}{|\boldsymbol{s}_{\mathrm{app}}-\boldsymbol{s}_0|}\right)\right)\frac{\boldsymbol{s}_{\mathrm{app}}-\boldsymbol{s}_t}{|\boldsymbol{s}_{\mathrm{app}}-\boldsymbol{s}_t|} \tag{18}$$

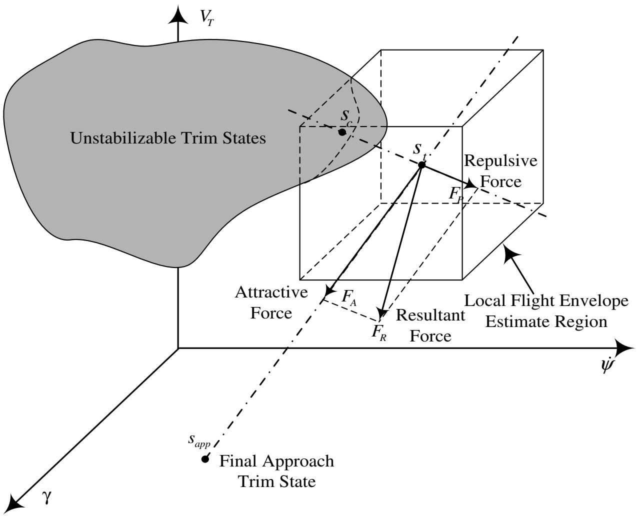

Fig. 9 Potential force for 3-D TSD.

2) Repulsive force: the centroid of ith detected obstacle in the local flight envelope estimate region (Fig. 9) is defined as

$$\boldsymbol{s}_c^i=(x_c^i,y_c^i,z_c^i)=\left(\frac{1}{\mathrm{Vol}_o^i}\int_{\mathrm{Vol}} x\,\mathrm{dVol}^i,\frac{1}{\mathrm{Vol}_o^i}\int_{\mathrm{Vol}} y\,\mathrm{dVol}^i,\frac{1}{\mathrm{Vol}_o^i}\int_{\mathrm{Vol}} z\,\mathrm{dVol}^i\right) \tag{19}$$

In practice, we again decompose this local exploration volume (Fig. 8) into a grid; for three dimensions, this grid has $N*N*N$ elements. We again choose $N=20$, yielding 8000 elements to explore. Then, $\boldsymbol{s}_c^i$ is calculated as

$$\boldsymbol{s}_c^i=(x_c^i,y_c^i,z_c^i)=\left(\frac{1}{m^i}\sum_{j=1}^{m^i}x_j^i,\frac{1}{m^i}\sum_{j=1}^{m^i}y_j^i,\frac{1}{m^i}\sum_{j=1}^{m^i}z_j^i\right) \tag{20}$$

$$\boldsymbol{F}_P=k_p\sum_{i=1}^{n}\left(\frac{\mathrm{Vol}_o^i}{\mathrm{Vol}_a}\right)\cdot\frac{\boldsymbol{s}_t-\boldsymbol{s}_c^i}{|\boldsymbol{s}_t-\boldsymbol{s}_c^i|}\approx k_p\sum_{i=1}^{n}\left(\frac{m^i}{N*N*N}\right)\cdot\frac{\boldsymbol{s}_t-\boldsymbol{s}_c^i}{|\boldsymbol{s}_t-\boldsymbol{s}_c^i|} \tag{21}$$

3) Resultant force:

$$\boldsymbol{F}_R=\boldsymbol{F}_P+\boldsymbol{F}_A \tag{22}$$

4) Next step s_{i+1} through trim state space:

$$\mathrm{step}_i=k_f\cdot k_s\cdot\frac{\boldsymbol{F}_R}{|\boldsymbol{F}_R|}+(1-k_f)\mathrm{step}_{i-1} \tag{23}$$

$$s_{i+1}=s_i+\mathrm{step}_i \tag{24}$$

Where $\boldsymbol{s}_0$, $\boldsymbol{s}_t$, and $\boldsymbol{s}_c^i$ are trim state-space "position" vectors of target state, present state, and the ith dynamically discovered (explored) obstacle centroid, respectively. The amount of detected obstacles in this local region is n, and m^i is the number of elements contained in the ith dynamically discovered (explored) obstacle. Vol_o^i and Vol_a are the volumes of the ith explored obstacle and exploration region, respectively. The adjustable weighting coefficients of force that can be adjusted by the user are k_a, k_p, k_f, and k_s; they assume the same roles for 3-D TSD as they had for 2-D. A moving average filter was used to smooth the steps. TSD will follow a sequence of transitions to the target state with obstacle avoidance, provided such a path is feasible given the obstacle map.

To evaluate the performance of TSD and examine its sensitivity to model parameters, a case is studied in which an F-16 aileron is jammed. For the simulation, we use a trim database, initially unknown to TSD, specified by

$$D=\{(V_{T,k}^*\dot{\psi}_k^*\gamma_k^*,h)\}_{k=1,\ldots,N_D} \tag{25}$$

to represent the set of N_D possible (stabilizable) states. To enable real-time plan development, the space of feasible trim states must have a tractable size to support an efficient search. However, accuracy improves with increased trim state database resolution. In previous work [13,28] where D was presumed known with certainty from offline analysis, D is first contracted over altitude to produce a new and much smaller database D'. Further contraction was accomplished from D' to $\tilde{D}$ by a heuristic method, defining a series of nested concentric cubes inside the three-dimensional volume defined by D', a "coarse grid" over climb rate, turn rate, and airspeed trim states. This contraction method has been shown effective for emergency flight planning [9,13,28]. However, to guarantee feasibility of transitions between two random states in $\tilde{D}$, which may not be proximal in the trim state space, the system must transition through states outside $\tilde{D}$, which may not be possible. To address this in previous work [13,28], a specific transition database was developed through simulations, giving confidence that the transitions were indeed possible but substantially increasing total database size. With damage that must be characterized in real time, a trim state database generated through trim state discovery should cover the entire postfailure flight envelope. Such a database could be traversed locally rather than relying on a distinct transition database needed to ensure any of the large-scale transitions could in fact be achieved. The neighborhood of the final trim state in 3-D TSD is explored following a trim state sequence computed as

$$s_{ep} = \{s_{ep,i}\}_{i=-4}^{4} \tag{26}$$

where

$$s_{ep,i} = \{s_{ep,i,j}\}_{j=1}^{4} \tag{27}$$

where

$$s_{ep,i,j} = \left\{ s_{\text{app}} + \left(i \cdot r_x, j \cdot r_y \cdot \sin\left(\frac{2\pi}{16}k\right), j \cdot r_z \cdot \cos\left(\frac{2\pi}{16}k\right), \right) \right\}_{k=0}^{16} \tag{28}$$

where r_x, r_y, and r_z are the user-defined basic explored circle radii for airspeed (in feet per second), turn rate (in degrees per second), and path angle (in degrees), respectively.

D. Trim State Discovery Algorithm

Figure 10 describes the TSD algorithm. First, an initial stabilized trim condition must be established, defined as state vector $s_0 = (\,V_{T,0} \quad \dot\psi_0 \quad \gamma_0\,)^T$, and control vector $\boldsymbol{\mu}_0 = [\,\mu_{t_0} \quad \mu_{e_0} \quad \mu_{a_0} \quad \mu_{r_0}\,]^T$. In step 2, aircraft position is acquired from the aircraft's navigation system, including longitude, latitude, altitude, and yaw angle. In step 3, a vector of trim state transition times is defined based on a tradeoff between speed of trim space exploration and expected stability/stabilizability of the aircraft as it moves through trim state space. During the process of discovery, the aircraft will maintain its commanded reference state for time t_{mi}, and then it will generate a linearly interpolated transition from s_i to s_{i+1} over $[t_i + t_{mi}, t_i + t_{mi} + t_{si}]$. It then maintains s_{i+1} for duration t_{mi+1}. t_m is the time for AFP to calculate the local flight envelope. Minimizing t_s in turn minimizes transits though physical space. However, if t_s is too small, the controller will not have sufficient time to change states.

Changing t_s will produce different trajectories during the process of discovery. Therefore, the transition time should be set with consideration of parameter adaptation and the physical trajectory. In our simulations for TSD, transition time parameters are fixed due to the fixed step in trim state search, and we assume that $t_m = 0$ s and $t_s = 30$ s. From step 5 to step 12, a trim state discovery potential field algorithm directs the trim state transition to the final approach trim state, avoiding flight envelope constraint obstacles (Fig. 9) in trim state space. With the discovered path to the final trim state, a waypoint trajectory will be computed using the aircraft's kinematics model per step 15. Simultaneously, the discovered trim states and transitions are stored in a database for future use in flight planning. The emergency landing plan is then composed of sequences of trim flight segments corresponding with trim states determined feasible by TSD.

In this work, a pole-placement controller was used to enable smooth transitions between trim states, defined by $(V_{T,i}^*, \dot\psi_i^*, \gamma_i^*)^T \rightarrow (V_{T,j}^*, \dot\psi_j^*, \gamma_j^*)^T$ over predetermined transition time t_t. Linear interpolation was used to define reference inputs for intermediate flight conditions, as given by

$$\begin{cases} V_T^*(t) = V_{T,i}^*,\ \dot h^*(t) = \dot h_i^*,\ \dot\psi^*(t) = \dot\psi_i^* \ \ \forall\, t \in [t_i, t_i + t_{mi}] \\ V_T^*(t) = V_{T,i}^* + \frac{V_{T,j}^* - V_{T,i}^*}{t_{si}}(t - t_i - t_{mi}),\ \dot h^*(t) = \dot h_i^* + \frac{\dot h_j^* - \dot h_i^*}{t_{si}}(t - t_i - t_{mi}), \\ \dot\psi^*(t) = \dot\psi_i^* + \frac{\dot\psi_j^* - \dot\psi_i^*}{t_{si}}(t - t_i - t_{mi}) \ \ \forall\, t \in [t_i + t_{mi}, t_i + t_{mi} + t_{si}] \end{cases} \tag{29}$$

Following the procedure presented by Strube [13], the F-16 aircraft model is formulated from discrete aerodynamic data in tabular form rather than an analytical model. An approximate linearization method is used to describe dynamics in the neighborhood of each equilibrium or trim state z_k^* to compute a linear time-invariant model:

$$\dot x_k = A_k x_k + B_k u_k \tag{30}$$

```
Algorithm TSD( $s_0, T_0, t_0$ )
1.  Set initial state $s_0 = (V_{T,0}^*, \dot\psi_0^*, \gamma_0^*)^T$ with $\mu_0 = (\mu_{t_0}, \mu_{e_0}, \mu_{a_0}, \mu_{r_0})^T$ ;
2.  Read initial position $T_0 = (lon_0, lat_0, alt_0, \psi_0)^T$ ;
3.  Determine transition time sequence $t_{ti} = (t_{mi}, t_{si})^T$ ;
4.  Set final approach trim state $s_{app} = (V_{T,app}^* \ \dot\psi_{app}^* \ \gamma_{app}^*)^T$ ;
5.  while $|s_i - s_{app}| > \varepsilon$
6.    Compute attractive force $F_A$ = TSD_attractive($s_i, s_{app}$) ;
7.    Compute repulsive force $F_P$ = TSD_repulsive($s_i$, explorerange) ;
8.    Compute resultant force $F_R = F_P + F_A$ ;
9.    Set step $step_i$ = TSD_step($F_R, step_{i-1}$) ;
10.   Execute Trim State Discovery $s_{i+1} = (V_{i+1} \ \dot\psi_{i+1} \ \gamma_{i+1})^T = s_i + step_i$ ;
11.   Update Trim States Sequence $S = \{s_j\}_{j=0}^{i} \cup \{s_{i+1}\}$ ;
12. End while
13. Compute states $S_{ep}$ that should be explored around $s_{app}$ per eq. (16) (2D) or (26) (3D);
14. Explore $S_{ep}$ and save new trim states, $s_{i+1} = S_{ep}$ ;
15. Compute trajectory $T_{i+1} = (x_{i+1}, y_{i+1}, z_{i+1})^T$ = TSD_trajectory($T_i, s_i, s_{i+1}, t_{ti}$) ;
Return $S = \{s_j\}_{j=0}^{i}$ and $T_{i+1} = (x_{i+1}, y_{i+1}, z_{i+1})^T$ .
```

Fig. 10 Trim state discovery algorithm.

A_k and B_k are derived from the following equations:

$$A_{k,i} \approx \frac{\dot{x}_k|_{z_k^*+\varepsilon_i e_i,\mu_k^*} - \dot{x}_k|_{z_k^*,\mu_k^*}}{\varepsilon_i}, \qquad B_{k,i} \approx \frac{\dot{x}_k|_{z_k^*,\mu_k^*+\varepsilon_i e_i} - \dot{x}_k|_{z_k^*,\mu_k^*}}{\varepsilon_i} \tag{31}$$

where $A_{k,i}$ is the ith column of A, and $B_{k,i}$ is the ith column of B_k. A small, positive number is ε_i, and e_i is the ith column of an n-dimensional identity matrix where n is the size of the corresponding z_k^* or μ_k^*. In our research, without loss of generality for TSD, we used pole placement to design each controller $u_k = -K_k x_k$.

TSD searches in trim state space for a feasible sequence of trim states to the desired goal state. Given an open airspace with no obstacles or terrain, strict focus on exploration of trim state space is appropriate. However, in practice, conflicts in physical space may occur, and pilots as well as air traffic control would benefit from understanding the four-dimensional waypoint sequence $(x, y, h, t)_i$ as well as the flight (trim) state. Such a sequence can be generated in real time for TSD. As a baseline, each new trim state can be projected forward in time as it is selected to generate and communicate the "next waypoint" for TSD. If further predictions are required, estimates of waypoints over a longer horizon can be provided, although the potential field method could change these predictions as its model of the flight envelope evolves.

For the linear time-invariant model of an F-16, a variable-step Runge–Kutta method was used to forward-propagate aircraft state. The integration process generated a trajectory based on commanded trim states and provides the ability to generate physical-space waypoints. Equations (32) and (33) show the generated trajectory. Note that expected aircraft trajectories are dependent on transition time values, as will be discussed in the following case studies. The first case studies investigate results in which the potential field algorithm is applied to a model with strictly trim state-space constraints and TSD exploration goals, as previously discussed. This representation required augmentation, however, to enable consideration of hard physical constraints such as terrain, thus ensuring TSD trajectories are feasible in terms of the physical environment as well as the flight envelope:

$$v_B = [U, V, W]^T = [V_T \cos\alpha\cos\beta, V_T \sin\beta, V_T \sin\alpha\cos\beta]^T \tag{32}$$

IV. Trim State Discovery with Physical-Space Constraints

The TSD algorithm previously presented focused on trim state discovery without consideration of physical space. While an aircraft requiring TSD during an emergency will have priority handling, the craft still must ensure it flies to avoid terrain and adverse weather conditions that could worsen the impact of the failure or damage. Physical-space constraints must therefore be modeled and handled appropriately in TSD. In this work, physical constraints are incorporated into the artificial potential field TSD algorithm using two methods. Each is described next.

A. TSD with Physical-Space Constraints: Method 1

In our initial and most straightforward approach, additional repulsive force was applied to represent physical-space constraints. Figures 11 and 12 illustrate the potential sources of repulsive force constraints in physical and trim state spaces.

Constraints in physical space are described as

$$C = C_1 \cup C_2 \cup \ldots \cup C_{N_C} \tag{34}$$

where C_i represents a physical constraint, and N_C is the total number of physical constraints. For example, an altitude ceiling $h_{\max}$ can be defined as

$$C = \{(x, y, h) | h > h_{\max}\} \tag{35}$$

Projection of the current trim state transition to each neighboring trim state and the continued tracking of that next trim state, etc., yields a trajectory through physical space, which can be evaluated for conflict with physical-space constraints. Each physical-space conflict in turn generates a repulsive force to be incorporated into the selection of the next trim state. The possible trajectories of all neighboring trim states are calculated to determine physical constraint conflict parameter pc_i $i = 1, 2, \ldots 26$, indicating whether a physical constraint is present (represented by 1) or not present (represented by 0):

$$\mathrm{pc}_i = \begin{cases} 1 & T_i \cap C \neq \varnothing \\ 0 & T_i \cap C = \varnothing \end{cases} \tag{36}$$

where T_i is the trajectory corresponding to adjacent trim state s_i. The additional repulsive potential force based on physical-space constraints is then

$$\begin{cases} \dot{x} = U\cos\theta\cos\psi + V(-\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi) + W(\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi) \\ \dot{y} = U\cos\theta\sin\psi + V(\cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi) + W(-\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi) \\ \dot{h} = U\sin\theta - V\sin\phi\cos\theta - W\cos\phi\cos\theta \\ \dot{\psi} = \frac{q\sin\phi + r\cos\phi}{\cos\theta} \end{cases} \tag{33}$$

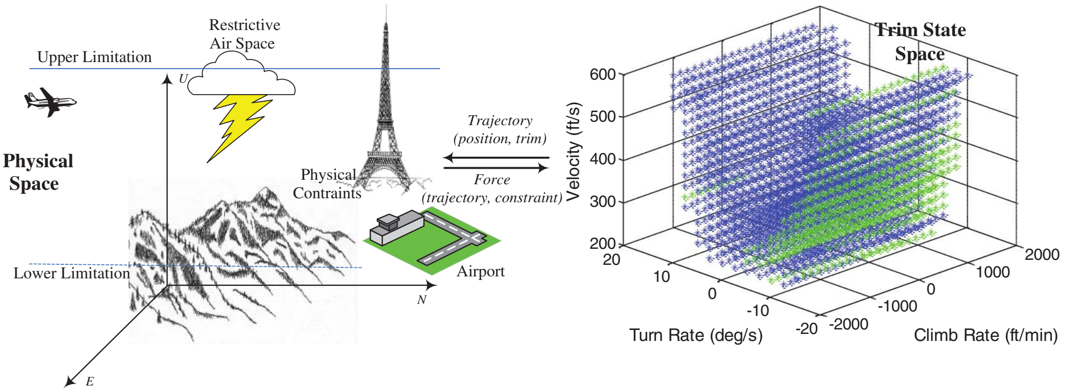

Fig. 11 TSD with physical (left) and trim (right) state-space constraints.

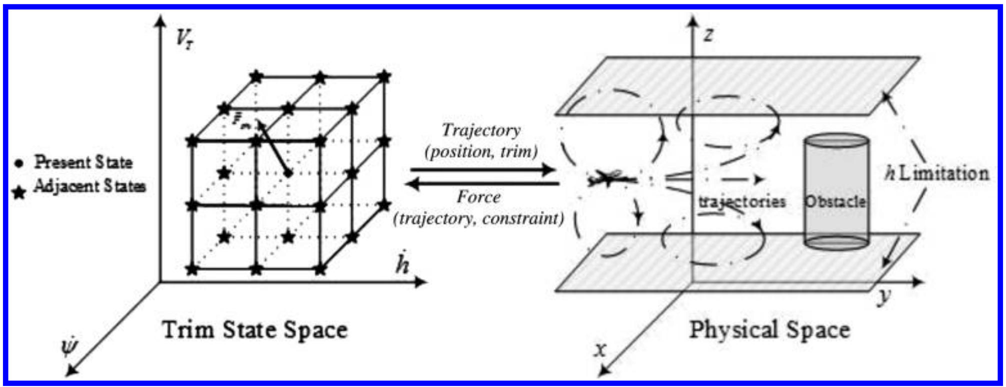

Fig. 12 Dual trim state and physical-space topologies.

$$\boldsymbol{F}_{\mathrm{pc}} = k_{\mathrm{pc}} \sum_{i=1}^{26} \mathrm{pc}_i \cdot \frac{\boldsymbol{s}_t - \boldsymbol{s}_i}{|\boldsymbol{s}_t - \boldsymbol{s}_i|} \tag{37}$$

where k_{pc} are weighting factors to adjust the level to which physical constraints impact choice of the next trim state. The total resultant force then becomes

$$\boldsymbol{F}_R = \boldsymbol{F}_P + \boldsymbol{F}_A + \boldsymbol{F}_{\mathrm{pc}} \tag{38}$$

In this method, for some cases, one step trajectory prediction may not be sufficient, because for some neighboring trim state points, the aircraft may encounter dead-end paths, i.e., all neighboring trim states not yet explored have conflicts with physical constraints. To ensure avoidance of such dead-end paths, multiple prediction steps are required. The corresponding search has branching factor 26, the number of adjacent trim states, and user-specified depth d, set to trade off confidence that dead-end paths are avoided with computational overhead required for the search. All 26^d trajectories must be explored, with each path pruned only when a dead end is encountered. For the ith ($i = 1, \ldots, 26$) search subtree, if the percentage of the trajectories that present physical conflicts in all the 26^{d-1} trajectories in this search subtree is larger than a user-specified risk threshold r_{sc}, $\mathrm{pc}_i = 1$; otherwise, $\mathrm{pc}_i = 0$. The influence of d is illustrated in test cases presented next.

B. TSD with Physical-Space Constraints: Method 2

In the previously described method 1, judicious selection of d may be difficult: if d is too low, the system may encounter a dead-end path, but if d is too high, calculation costs may be prohibitive. We therefore introduce a second method not based on combinatoric search to deal with physical-space constraints during TSD. In this method, instead of supplementing the repulsive force term in the potential field search process, the trim state sequence is optimized strictly through trim state-space considerations, but a postprocessing step is added to guarantee the resulting trajectory does not introduce physical conflict. While this two-step procedure no longer can guarantee even a locally optimal path through trim state space, given the potential for suboptimal path changes to resolve physical constraints, it can provide a computationally efficient, thus practical, means to generate obstacle-free TSD paths. Figure 13 describes this two-step algorithm that consists of first calculating the projected trajectory from a selected next trim state and then overriding with an "escape-trajectory" algorithm as needed to ensure any physical conflicts are resolved. In this algorithm, trim states sequences will not be influenced by physical constraints. Instead, the physical-space trajectory is augmented with escape trajectory segments specifically designed for avoidance purposes, after which the TSD process continues per the original algorithm (Fig. 10).

```
Escape Algorithm
Set initial state $s_0=\left(V^*_{T,0},\dot{\psi}^*_0,\gamma^*_0\right)^T$ with $\mu_0=\left(\mu_{t_0},\mu_{e_0},\mu_{a_0},\mu_{r_0}\right)^T$;
Compute the neighboring
 trim states of $s_0$;
Update trim states database TS;
Read initial position $T_0=\left(x_0,y_0,h_0,\psi_0\right)^T$;
Determine transition time sequence $t_{ti}=\left(t_{mi},t_{si}\right)^T$;
Set final approach trim state $s_{app}=\left(V^*_{T,app}\ \dot{\psi}^*_{app}\ \gamma^*_{app}\right)^T$;
while $|s_i-s_{app}|>\varepsilon$
    Compute attractive force $F_A=\mathrm{TSD\_attractive}(s_i,s_{app})$;
    Compute repulsive force $F_P=\mathrm{TSD\_repulsive}(s_i,\mathrm{explorerange})$;
    Compute resultant force $F_R=F_P+F_A$;
    Set step $step_i=\mathrm{TSD\_step}\left(F_R,step_{i-1}\right)$;
    Compute next trim state $s_{i+1}=\left(V_{i+1}\ \dot{\psi}_{i+1}\ \gamma_{i+1}\right)^T=s_i+step_i$;
    Add Trim States to valid set $S=\left\{s_j\right\}_{j=0}^{i}\cup\left\{s_{i+1}\right\}$;
    Predict $T_{i+1}=\mathrm{TSD\_trajectory}\left(T_i,s_i,s_{i+1},t_{ti}\right)$
    if ( $T_{i+1}$ is not safe)
        Compute escape direction:
        $D_{\mathrm{esc}}=\mathrm{TSD\_escape\_direction}(T_i,C)$;
        Compute viable $s_{R\max}$ and $s_{R\min}$ in all known trim states;
        Compute escape trajectory:
        $T_{\mathrm{esc}}$=Escape_trajectory($s_{R\max},s_{R\min},D_{\mathrm{esc}},s_i,s_{i+1},T_i$)
    end if
        Update physical trajectory T;
        Compute the nearby trim states of $s_{i+1}$;
        Update trim states database TS
end while
```

Fig. 13 TSD with physical constraints.

```
Algorithm
$T_{esc}$=Escape_trajectory($s_{R\max},s_{R\min},D_{\mathrm{esc}},s_i,T_i$)
Calculate current position $P_0$;
if (altitude has a conflict)
    Transform to the maximal climb rate trim state
     or minimal climb rate trim state;
    Keep this trim state for $3t_{ti}$;
    Transform to $s_{i+1}$;
else
    Transform to $s_{R\min}$;
    Keep $s_{R\min}$ for $t_{Rmin}$;
    Transform to $s_{R\max}$;
    Keep $s_{R\max}$ for $t_{Rmax}$;
     Transform to $s_{i+1}$;
end if
update $T_{esc}$
```

Fig. 14 Escape trajectory generation function for the TSD algorithm (method 2).

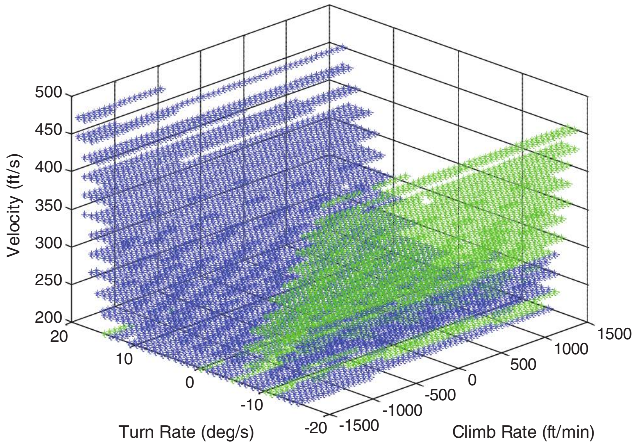

Fig. 15 Trim states for the damaged F-16 aircraft.

As an example, illustrating use of the Figure 13 algorithm, assume the aircraft satisfies the following constraints:

$$\begin{cases} x < x_{\max} \\ y < y_{\max} \\ h_{\min} < h < h_{\max} \end{cases} \tag{39}$$

Each time a new trim state is chosen, the physical-space trajectory (step 14) will be projected to make sure this trajectory is safe. If this trajectory is deemed safe, it must satisfy

$$\begin{cases} \max(x_{i+1}) < x_{\max} - x_{\mathrm{sd}\_i} \\ \max(y_{i+1}) < y_{\max} - y_{\mathrm{sd}\_i} \\ \max(h_{i+1}) < h_{\max} - h_{\mathrm{sd}\_\mathrm{high}\_i} & \text{if } \dot{h}_{i+1} > 0 \\ \min(h_{i+1}) > h_{\min} + h_{\mathrm{sd}\_\mathrm{low}\_i} & \text{if } \dot{h}_{i+1} < 0 \end{cases} \tag{40}$$

where x_{sd_i}, y_{sd_i}, $h_{\mathrm{sd}\_\mathrm{high}\_i}$, $h_{\mathrm{sd}\_\mathrm{low}\_i}$ are safety distances for longitude, latitude, upper altitude, and lower altitude, respectively, chosen to guarantee that, if the aircraft must execute an escape trajectory in the next TSD step, the aircraft has sufficient maneuvering clearance. Generally safe clearance distances may be a function of the ith trim state being followed. For the examples presented in this paper, we define clearance distance values common across all trim states i to satisfy the following constraints:

$$\begin{cases} x_{\mathrm{sd}\_i} > \min\{2R_{\min}, V_{\min} t_s\} \\ y_{\mathrm{sd}\_i} > \min\{2R_{\min}, V_{\min} t_s\} \\ h_{\mathrm{sd}\_\mathrm{high}\_i} > \max\{\dot{h}_{\max} t_s, 0\} \\ h_{\mathrm{sd}\_\mathrm{low}\_i} > -\min\{\dot{h}_{\min} t_s, 0\} \end{cases} \tag{41}$$

where $R_{\min}$ is the minimum possible turning radius, and $V_{\min}$ is the minimum possible airspeed based on the trim state space discovered thus far. The maximum and minimum possible climb rates are $\dot{h}_{\max}$, $\dot{h}_{\min}$, respectively, based on the trim state space discovered thus far.

Figure 14 describes computation of an escape trajectory (step 18 in Fig. 13). When an altitude conflict is recognized, the aircraft transitions to its maximal or minimal climb rate trim state to "escape." When a lateral conflict in the forward direction is also present, trim states with maximal turning radius and minimal turning radius are combined to compose this escape trajectory, as has been proposed in other work [28,29]. First, the goal is to transition to a trim state with minimal radius from the known trim state database. This will allow the aircraft to turn away from the obstacle as quickly as possible.

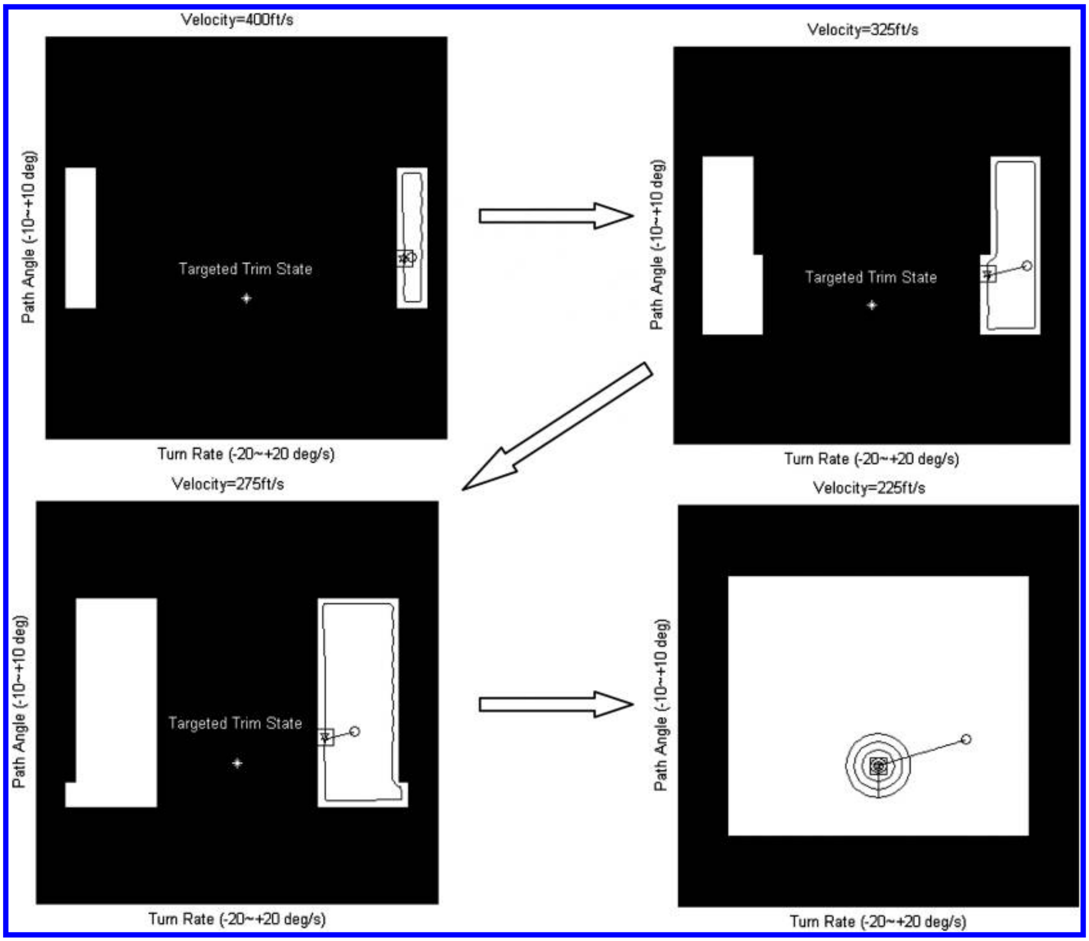

Fig. 16 2-D TSD; dark regions are envelope constraint obstacles.

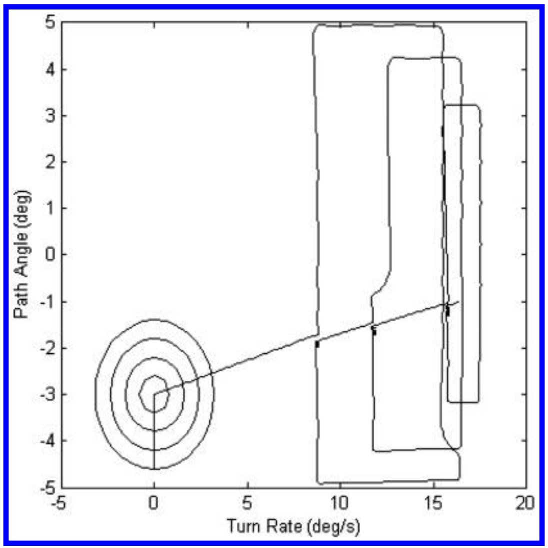

Fig. 17 Full solution path for 2-D TSD.

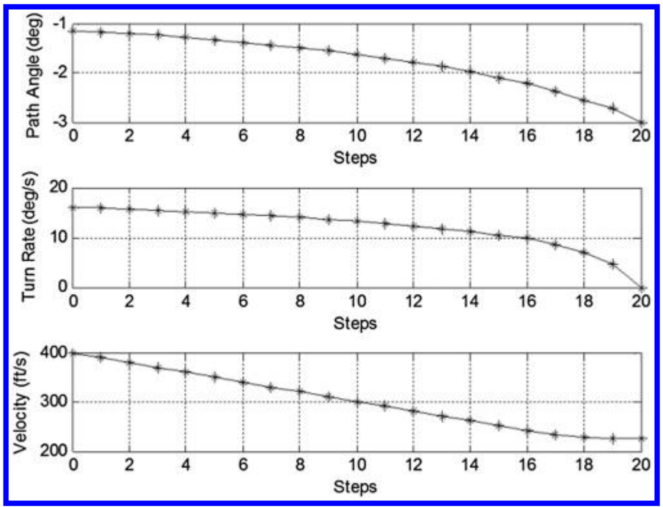

Fig. 19 Trim state time history (in TSD steps) for 3-D TSD.

Once achieved, the minimum radius turn is maintained for time $t_{R\ \min}$ and then transformed to a trim state with maximal radius. This sequence ensures the aircraft will move away from the obstacle as effectively as possible [27,28]. To ensure the escape traversal direction leads away from the lateral obstacle, $t_{R\ \min}$ is chosen. The escape trajectory is maintained until vertical and lateral physical-space separations are sufficient to prevent another conflict, at least in the near term when the distances from the current position to the nearest physical-space constraint satisfy

$$\begin{cases} d_x > v_{i+1}t_s \\ d_y > v_{i+1}t_s \\ d_h > |\dot{h}_{i+1}|t_s \end{cases} \tag{42}$$

where d_x d_y, and d_h are the longitude distance, latitude distance, and altitude distance, respectively.

V. Case Study

A. Two-Dimensional Trim State Discovery

The trim state discovery algorithms were applied to an F-16 aircraft with the aileron jammed at -10 deg. Figure 15 shows the flight envelope of the damaged F-16 aircraft. The color of each trim state indicates its characteristics, with green indicating a stable and controllable trim state and blue representing an unstable but controllable trim state. For TSD, the initial trim state is defined as $h = 10,000$ ft, $V_T = 400$ ft/s, $\dot{\psi} = 16.5$ deg/s, and $\gamma = -1$ deg. The final (targeted approach) trim state is $\dot{\psi} = 0$ deg/s, $\gamma = -3$ deg. First, TSD is conducted in two dimensions, with turn rates from -25 to $+25$ deg and flight-path angles from -10 to $+10$ deg. Note that the envelope characteristics can be exactly obtained if the state belongs to a certain neighborhood of the present trim state, with TSD configured to take steps of ± 0.2 deg in the path angle dimension and ± 0.4 deg/s in the turn rate dimension. Figure 16 shows the search process over 2-D trim space augmented to explore different airspeeds. When velocity is changed from 400 to 225 ft/s, TSD is caught on a dead-end path, and thus cannot steer to the final approach trim state. In this case, the velocity must also be changed. For simplicity, TSD keeps path angle and turn rate constant during velocity (airspeed) excursions to maintain the continuity. Figure 17 shows the full path of augmented 2-D TSD with $k_a = 1$, $k_p = 40$, $k_f = 1$, and $k_s = 5$. These trim states on this sequence are all in the flight envelope, so these trim states can build a path from the initial trim state to the targeted trim state, which means all the trim states on this sequence can be used for further path plans. In the 2-D TSD, as illustrated, a dead-end path can result, and that will reduce the TSD efficiency.

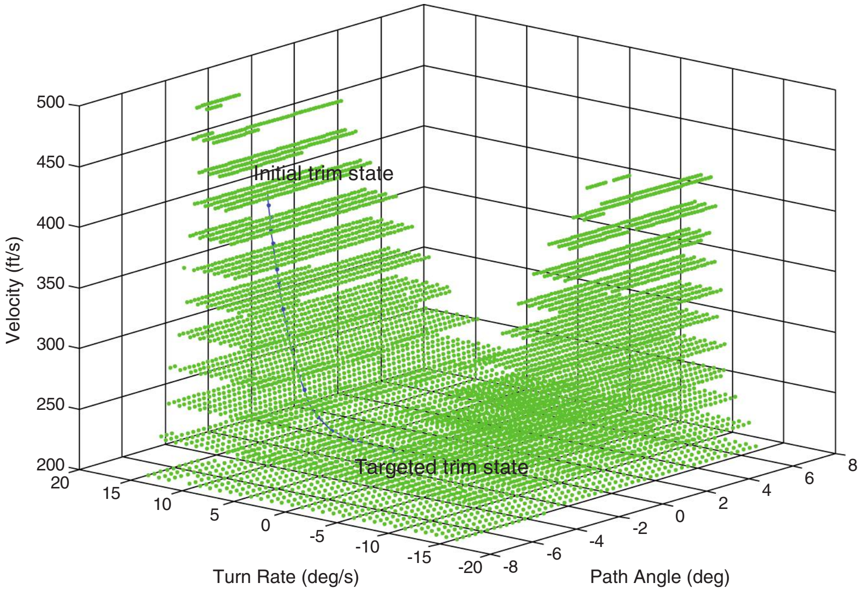

Fig. 18 3-D TSD in trim state space.

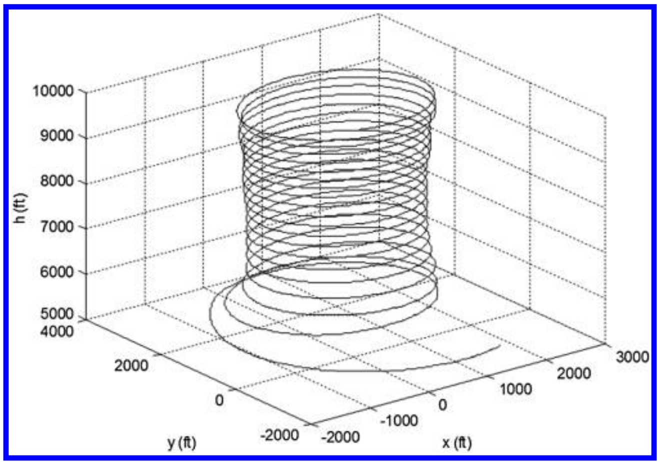

Fig. 20 Flight trajectory in physical space for 3-D TSD.

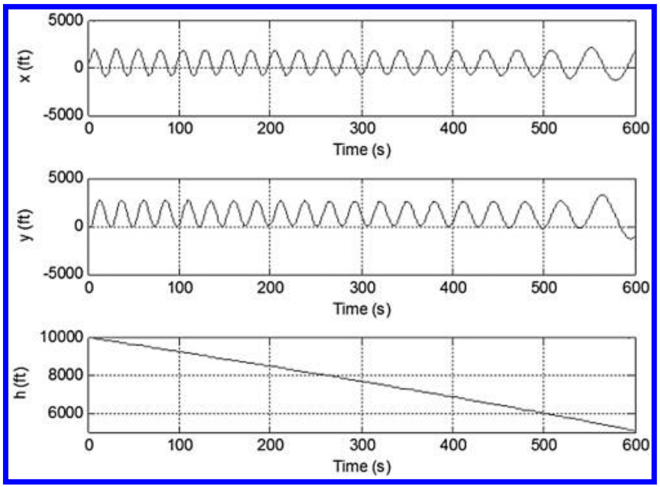

Fig. 21 3-D position (x,y,h) time history in physical space for 3-D TSD (one **TSD** step **= 30 s).**

B. Three-Dimensional Trim State Discovery

As shown, 2-D TSD will explore some states more than once, a pattern that cannot be avoided due to the limited ability to traverse the full flight envelope at different airspeeds. This 2-D TSD simulation motivates the inclusion of velocity as well as turn rate and path angle

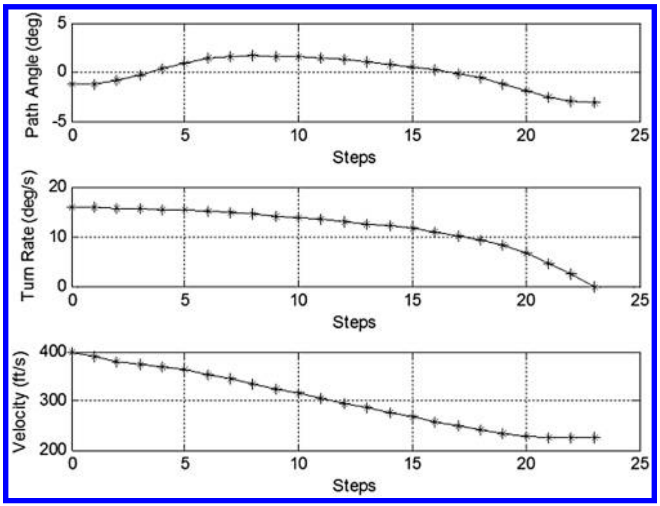

Fig. 23 Trim state time history (in TSD steps) for 3-D TSD with physical constraints (method 1).

in a 3-D TSD search, which slows the search process but improves TSD's exploration capability.

For the full 3-D TSD (Fig. 10), the same F-16 damage condition is simulated with the initial trim state defined as $h = 10,000$ ft, $V_T = 400$ ft/s, $\dot{\psi} = 16$ deg/s, and $\gamma = -1.2$ deg. The final (targeted) trim state is defined as $\dot{\psi} = 0$ deg/s, $\gamma = -3$ deg, and $V_T = 225$ ft/s. Figure 18 shows the results of 3-D TSD. The green dots indicate stabilizable trim states, and blue path overlaid shows the path through 3-D trim space generated by TSD. Compared with the 2-D cases, 3-D TSD has the ability to traverse the envelope more effectively, as it can explore velocity space in addition to the climb rate and turn rate. Figure 19 depicts the changes in V_T, $\dot{\psi}$, and γ over time for 3-D TSD, whereas Fig. 20 presents the flight trajectory in physical space for this same 3-D TSD solution. Figure 21 shows the 3-D physical position time history for 3-D TSD. Here, we choose $k_a = 1$, $k_p = 40$, $k_f = 1$, and $k_s = 5$, selected from a manual tuning process to balance efficient progress toward the goal trim state with successful obstacle avoidance. Obviously, 3-D TSD can expand the fight envelope more efficiently and reach the targeted final trim state more quickly.

C. Trim State Discovery with Physical-Space Constraints

The same damaged F-16 scenario was studied with a 3-D TSD simulation using method 1 to avoid physical constraints. For this example, altitude is constrained to be higher than 9000 ft throughout

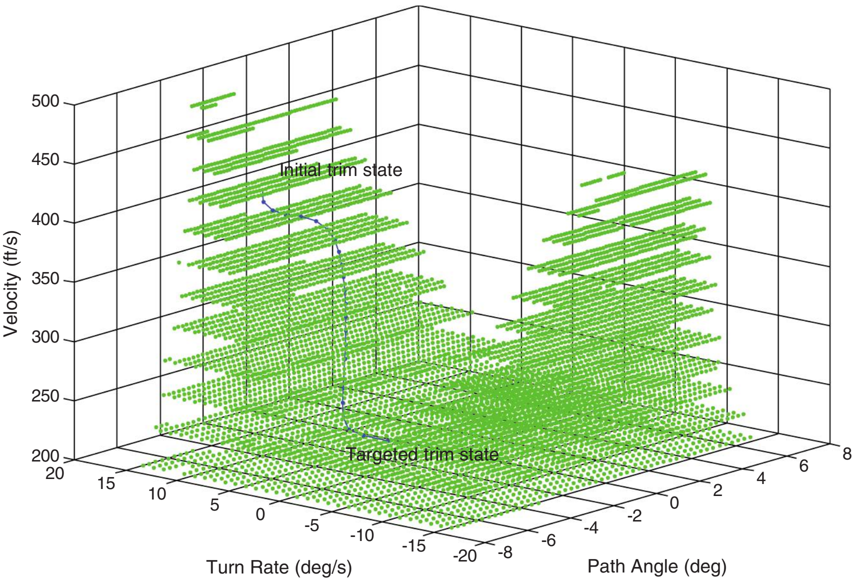

Fig. 22 3-D TSD with physical constraints (method 1).

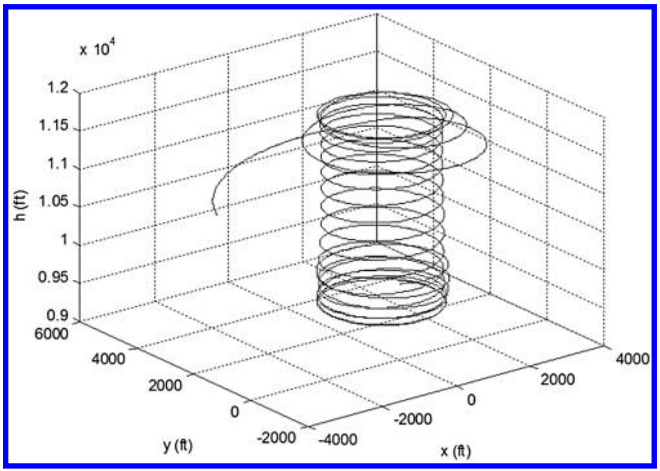

Fig. 24 Flight trajectory for 3-D TSD with physical constraints (method 1).

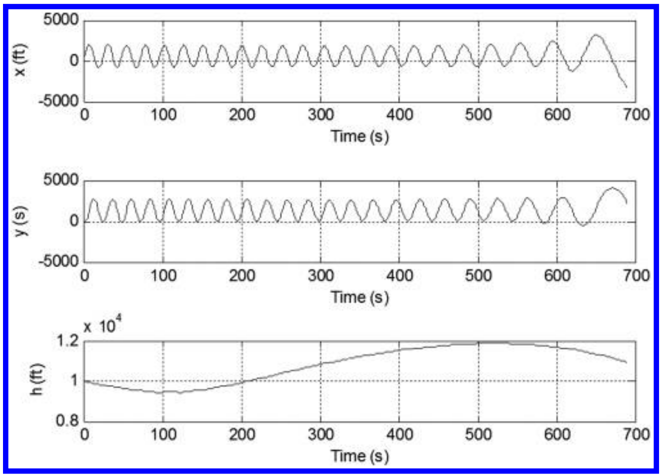

Fig. 25 3-D position (x,y,h) time history for 3-D TSD with physical constraints (method 1).

TSD. In this simulation, we choose $k_a = 1$, $k_p = 40$, $k_f = 1$, $k_s = 5$, $k_{pc} = 0.08$, and $d = 1$, the same baseline APF weights as for the previous example but with collision search "depth" d of 1. The results of this case are shown in Figs. 22–24, with Fig. 22 showing the path through 3-D trim state space; Fig. 23 plotting V_T, $\dot{\psi}$, and γ time histories; and Fig. 24 illustrating the physical-space trajectory. Figure 25 shows the corresponding position time history. In contrast with the previous 3-D case, the physical constraint resulted in selection of trim states with positive flight-path angles to obtain sufficient altitude to avoid the 9000 ft altitude floor. The additional repulsive potential force based on physical-space constraints is introduced and successfully avoided the aircraft from physical constraints.

Next, we consider the same damaged F-16 condition but with the initial trim state defined as $h = 5000$ ft, $V_T = 225$ ft/s, $\dot{\psi} = 12$ deg/s, and $\gamma = -4$ deg; and the final (targeted) trim state defined as $V_T = 225$ ft/s, $\dot{\psi} = -12$ deg/s, and $\gamma = 4$ deg. We choose $k_a = 1$, $k_p = 40$, $k_f = 1$, $k_s = 5$, $k_{\text{pc}} = 0.08$, and $d = 1$, the same parameter value set used in the previous case. In our simulation, we assume that y should be less than 6000 ft, representing a no-fly zone or "wall" in the longitudinal (east–west) direction. Figure 26 shows the results of 3-D TSD with physical constraints using method 1, whereas Fig. 27 presents the 3-D physical-space trajectory. Note that these figures are the same as for the case without physical constraints, except for the overlaid gray shading indicating the physical obstacle. As shown, the aircraft does in fact violate the no-fly zone constraint for this case ($y < 6000$ ft). The reason for this failure is that a dead-end path is encountered, which means all neighboring trim states not yet explored have conflicts with physical constraints and $\boldsymbol{F}_{\text{pc}} = 0$, representing a situation where opposing repulsive potential fields cancel, enabling the aircraft to believe it can pass through the physical obstacle constraints. To address this problem, the search depth d must be increased.

For the next case study, the same problem was formulated, except that we now choose $d = 2$ and risk threshold $r_{\text{sc}} = 40\%$. Figures 28–30 show the results of 3-D TSD with physical constraints using method 1. Figure 29 provides time histories of V_T, $\dot{\psi}$, and γ for both $d = 1$ and $d = 2$. As shown, even the depth $d = 2$ provides sufficient lookahead to avoid the no-fly zone obstacle. To do this, the trim state sequence is altered; for this case, the number of trim states increases from 19 to 24, as shown in Figure 29. Since d is increased from 1 to 2, the TSD algorithm can successfully avoid the obstacle. As a result, the calculation cost (time) increases substantially.

Next, physical-space collision avoidance method 1 is applied to solve the same TSD case. Figure 31 shows the position history in physical space for 3-D TSD with physical constraints (method 2) given $y_{\text{sd}\_i} = 2000$ ft. For all these cases, the altitude of the aircraft must remain above sea level so we choose $h_{\text{sd}\_\text{low}\_i} = 1000$ ft for all trim states i. Note that, for this example, the only physical obstacles present are a wall in y plus an altitude floor, so safe distances $h_{\text{sd}\_\text{high}\_i}$ and x_{sd_i} are not relevant. In Fig. 31, the red trajectory depicts the new

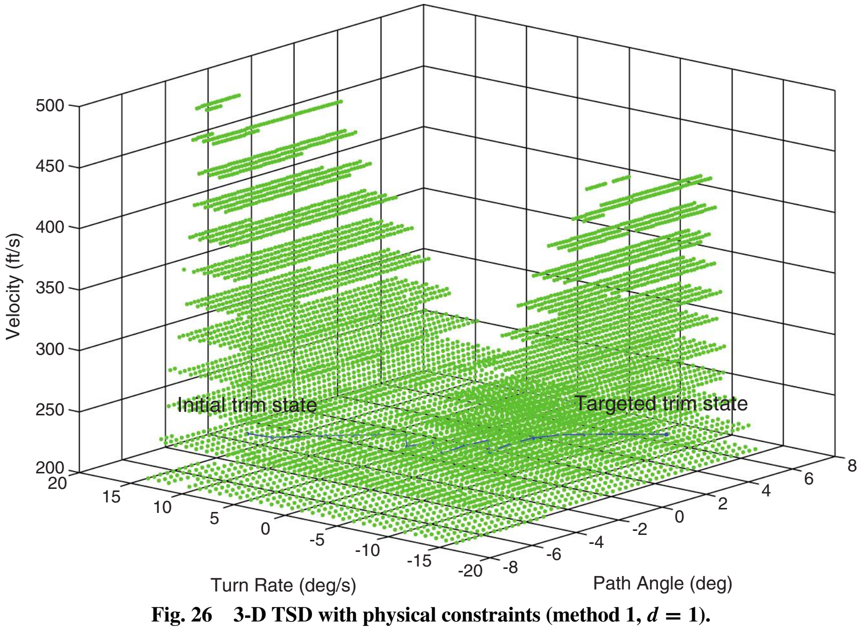

Fig. 26 3-D TSD with physical constraints (method 1, $d = 1$).

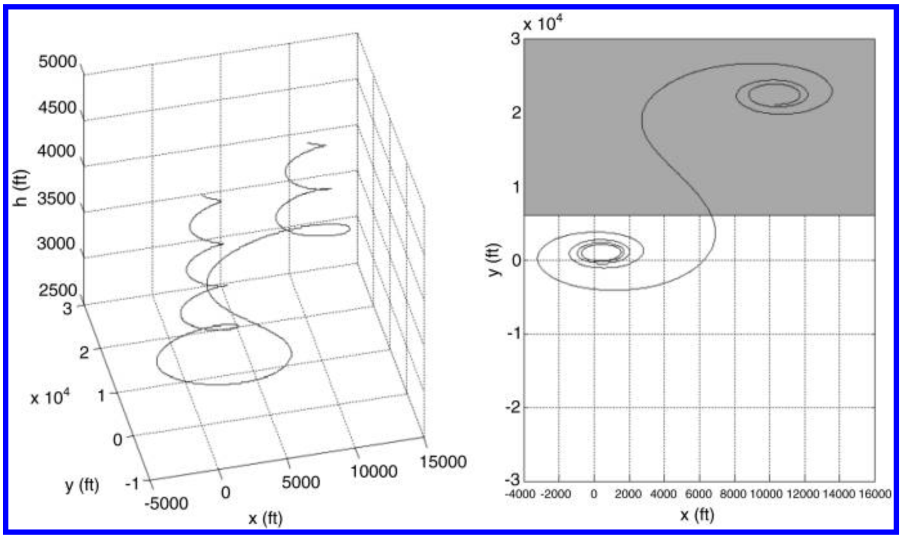

Fig. 27 Position history in physical space for 3-D TSD (method 1, $d = 1$).

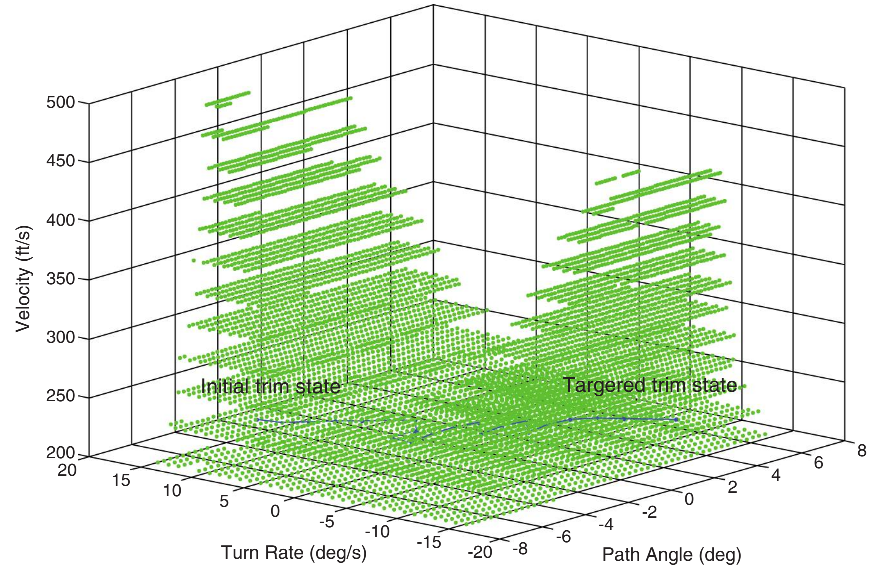

Fig. 28 3-D TSD with physical constraints (method 1, $d = 2$).

escape trajectory segment. With method 2, the trim state sequence is the same as was shown in Fig. 26, but an escape trajectory segment is inserted into the physical trajectory to enable the aircraft to avoid collision with defined physical constraints, after which TSD (from the last explored trim state) is continued. In contrast to method 1, method 2 is more efficient and can guarantee the aircraft can avoid the physical constraints.

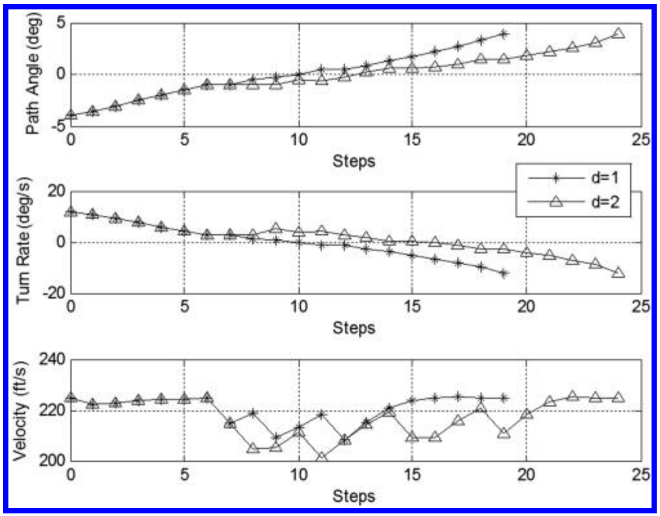

Fig. 29 Trim state time histories.

VI. Conclusions

A trim state discovery algorithm has been proposed to guide a physically disabled aircraft to explore feasible trim states that can be subsequently sequenced by an adaptive flight planner into a safe landing trajectory. TSD is of importance in situations where damage or failures cannot be explicitly characterized and matched to a known model. Emphasis is placed on efficient and correct identification of trim states at airspeed, turn rate, and flight-path angle values advantageous for landing approaches. Local flight envelope estimates enable safe exploration of the local neighborhood of each achieved trim state to determine stability of the states in that neighborhood. An artificial potential field motion planning method has been adapted to effectively explore the target trim state neighborhood while respecting envelope constraints in trim state space. Two-dimensional TSD is first defined as a search over the space of turn rates and flight-path angles, both of which can transition between values faster than would be possible with appreciable airspeed changes. However, dead-end paths can result, as illustrated.

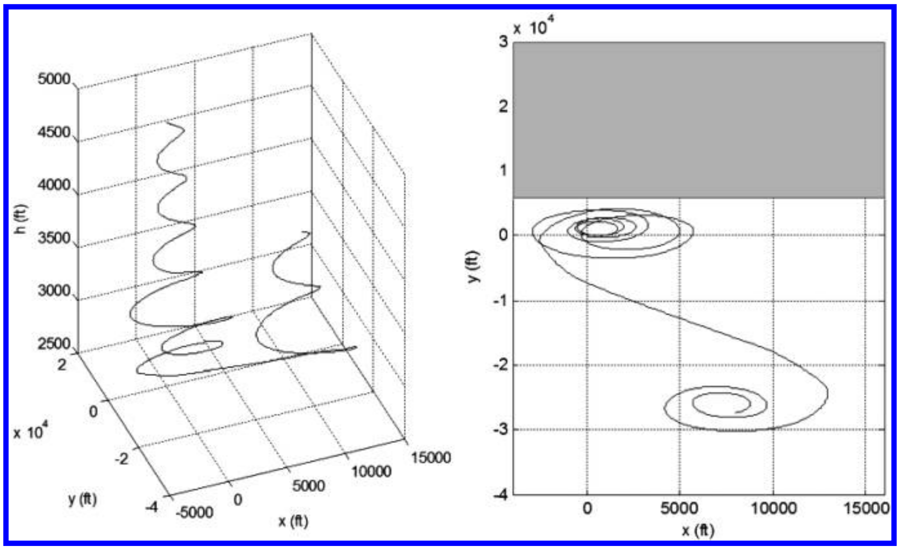

Fig. 30 Position history in physical space for 3-D TSD (method 1, $d = 2$).

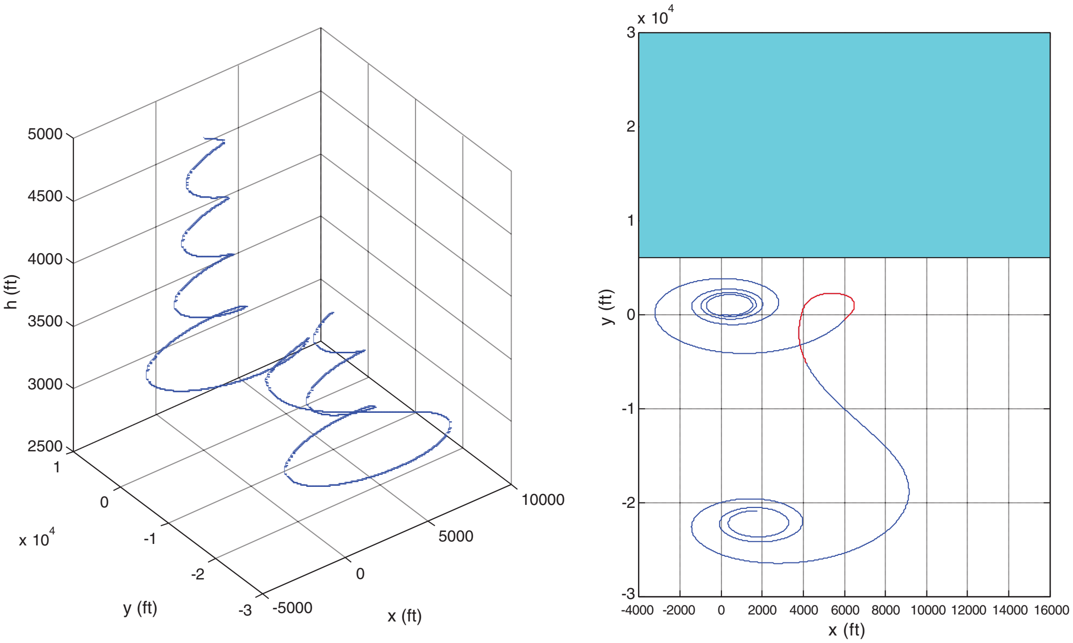

Fig. 31 Flight trajectory for 3-D TSD with physical constraints (method 2).

To expand the fixed-airspeed flight envelope more efficiently, a 3-D TSD algorithm explores an airspeed, turn rate, and flight-path angle search space. To cope with physical constraints such as terrain or restricted airspace, two methods are proposed. In the first method, an additional repulsive force is added to each trim state-space exploration option based on conflicts identified in physical space by projecting each trim-state option into physical space. While simple, this method fails when adjacent trim states all have conflicts with physical constraints or when repulsive terms cancel. In a second method devised to circumvent this issue, an escape-trajectory calculation algorithm is employed to make sure the damaged aircraft consistently avoids physical obstacles. These two methods are not proven optimal, but they are intuitive real-time strategies by which an aircraft can effectively avoid physical constraints as it identifies its performance envelope.

In future work, further consideration of factors such as path distance, transition time, and control rates can improve the efficiency and accuracy of the trim state discovery process. Trajectory optimization algorithms may also be introduced over a finite horizon to migrate from the segmented model of trim state space to a formulation supporting use of accelerated flight segments as part of an optimal solution. The TSD problem may also in future work be transformed to a parameter optimization problem that can be solved by nonlinear programming. The impacts of system identification and flight envelope estimation must be factored into the decision on the amount of time each trim state must be held. If it can be confirmed which estimated parameter(s) in the aircraft model are changing, the system model can be updated for only those parameters that can decrease the amount of time devoted to model identification.

While the adaptive flight planner, including trim state discovery, is cast in the context of flight management to be a reference for the flight crew, it can also be used in the context of future autonomous (or remotely piloted) operations of unmanned as well as manned aircraft. Although specifics of merging an AFP into a flight deck or unmanned aircraft ground control station are still under development, such a capability represents a new frontier in adaptation for recovery that, once certified and deployed, can move the air transportation system beyond simply reverting control to the pilot in the most dangerous degradation scenarios.

References

[1] "Fatal Airliner Hull-Loss Accidents," *Flight Safety Foundation Accident Information*, http://aviation-safety.net/statistics/period/stats.php?cat=A1 [retrieved 10 Jan. 2008].

[2] Lansdall, M., Lewis, L., and Bezdek, W. J., "The History of Commercial Simulators and the Boeing 777 Systems Integration Lab," *AIAA Modeling and Simulation Technologies Conference and Exhibit*, AIAA Paper 2004-5150, Aug. 2004.

[3] Helmreich, R. L., and Merritt, A. C., "Safety and Error Management: The Role of Crew Resource Management," *Aviation Resource Management*, edited by Hayward, B. J., and Lowe, A. R., Ashgate, Aldershot, U.K., 2000, pp. 107–119.

[4] Bodson, M., and Groszkiewicz, J. E., "Multivariable Adaptive Algorithms for Reconfigurable Flight Control," *IEEE Transactions on Control Systems Technology*, Vol. 5, No. 2, 1997, pp. 217–229. doi:10.1109/87.556026

[5] Johnson, E. N., and Kannan, S. K., "Adaptive Flight Control for an Autonomous Unmanned Helicopter," *AIAA Guidance, Navigation and Control Conference*, AIAA Paper 2002-4439, Aug. 2002.

[6] Nguyen, N. T., Krishnakumar, K. S., Kaneshige, J. T., and Nespeca, P. P., "Flight Dynamics and Hybrid Adaptive Control of Damaged Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 751–764. doi:10.2514/1.28142

[7] Schutte, P. C., and Trujillo, A. C., "Flight Crew Task Management in Non-Normal Situations," *Proceedings of the Human Factors and Ergonomics 40th Society Annual Meeting*, SAGE Publ., Thousand Oaks, CA, 1996, pp. 244–248.

[8] Chen, T. L., and Pritchett, A. R., "Development and Evaluation of a Cockpit Decision Aid for Emergency Trajectory Generation," *Journal of Aircraft*, Vol. 38, No. 5, Sept.–Oct. 2001, pp. 935–943. doi:10.2514/2.2856

[9] Atkins, E., Portillo, A., and Strube, I., "Emergency Flight Planning Applied to Total Loss of Thrust," *Journal of Aircraft*, Vol. 43, No. 4, July–Aug. 2006, pp. 1205–1216. doi:10.2514/1.18816

[10] Tang, Y., Atkins, E., and Sanner, R., "Emergency Flight Planning for a Generalized Transport Aircraft with Left Wing Damage," *AIAA Guidance, Navigation and Control Conference and Exhibit*, AIAA Paper 2007-6873, Aug. 2007.

[11] Belcastro, C. M., and Jacobson, S., "Future Integrated Systems Concept for Preventing Aircraft Loss-of-Control Accidents," *AIAA Guidance, Navigation and Control Conference*, AIAA Paper 2010-8142, 2010.

[12] Meuleau, N., Plaunt, C., Smith, D., and Smith, T., "An Emergency Landing Planner for Damaged Aircraft," *Proceedings of the 21st Innovative Applications of Artificial Intelligence Conference*, AAAI Press, Pasadena, CA, 2009, pp. 114–121.

[13] Strube, M. J., "Post-Failure Trajectory Planning from Feasible Trim State Sequences," M.S. Thesis, Aerospace Engineering, Univ. of Maryland, College Park, MD, 2005.

[14] Michel, A. N., Nam, B. H., and Vittal, V., "Computer Generated Lyapunov Functions of Interconnected Systems: Improve Results with Applications to Power Systems," *IEEE Transactions on Circuits Systems*, Vol. 31, No. 2, Feb. 1984, pp. 189–198. doi:10.1109/TCS.1984.1085483

[15] Vannelli, A., and Vidyasagar, M., "Maximal Lyapunov Functions and Domains of Attraction for Autonomous Nonlinear Systems," *Automatica*, Vol. 21, No. 1, Jan. 1985, pp. 69–80. doi:10.1016/0005-1098(85)90099-8

[16] Chiang, H., Hirsch, M. W., and Wu, F. F., "Stability Regions of Nonlinear Autonomous Dynamical Systems," *IEEE Transactions on Automatic Control*, Vol. 33, No. 1, Jan. 1998, pp. 16–27. doi:10.1109/9.357

[17] Chesi, G., Garulli, A., Tesi, A., and Vicino, A., "LMI-Based Computation of Optimal Quadratic Lyapunov Functions for Odd Polynomial Systems," *International Journal of Robust and Nonlinear Control*, Vol. 15, No. 1, Dec. 2004, pp. 35–49. doi:10.1002/rnc.967

[18] Tibken, B., "Estimation of the Domain of Attraction for Polynomial Systems via LMIs," *Proceedings of the 39th IEEE Conference of Decision and Control*, Vol. 4, IEEE, Piscataway, NJ, Dec. 2000, pp. 3860–3864.

[19] Amato, F., Cosentino, C., and Merola, A., "On the Region of Attraction of Nonlinear Quadratic Systems," *Automatica*, Vol. 43, No. 12, 2007, pp. 2119–2123. doi:10.1016/j.automatica.2007.03.022

[20] Kohout, B., "Challenges in Real Time Obstacle Avoidance," *AAAI Spring Symposium on Real-Time Autonomous Systems*, Palo Alto, CA, 2000.

[21] Lumelsky, V., and Skewis, T., "Incorporating Range Sensing in the Robot Navigation Function," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 5, 1990, pp. 1058–1068. doi:10.1109/21.59969

[22] Lumelsky, V., "Path-Planning Strategies for a Point Mobile Automaton Amidst Unknown Obstacles of Arbitrary Shape," *Algorithmic*, Vol. 2, Nos. 1–4, 1987, pp. 403–430. doi:10.1007/BF01840369

[23] 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

[24] Yi, G., and Atkins, E. M., "Trim State Discovery for an Adaptive Flight Planner," *Proceedings of the 48th AIAA Aerospace Science Meeting*, AIAA Paper 2010-0416, Jan. 2010.

[25] McDonough, K., Kolmanovsky, I., and Atkins, E. M., "Recoverable Sets of Initial Conditions and Their Use for Aircraft Flight Planning After a Loss of Control Event," *AIAA Guidance, Navigation, and Control Conference (at Scitech)*, AIAA Paper 2014-0786, Jan. 2014.

[26] Liddy, T., and Lu, T., "Waypoint Navigation with Position and Heading Control Using Complex Vector Fields for an Ackermann Steering Autonomous Vehicle," *Australasian Conference on Robotics and Automation*, ARAA Paper 2007-6249, Dec. 2007.

[27] Borenstein, J., and Koren, Y., "Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 4, 1991, pp. 535–539. doi:10.1109/70.86083

[28] Choi, H. J., and Atkins, E. M., "An Analytic Trajectory Planner for Aircraft with Severe Damage or Failures," *Infotech@Aerospace Conference*, AIAA Paper 2009-2018, April 2009.

[29] 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, pp. 255–267, 2010. doi:10.2514/1.45264

This article has been cited by:

1. Stefan Schuet*, Thomas Lombaerts*, Diana Acosta†, John Kaneshige‡ and Kevin Wheeler§NASA Ames Research Center, Moffett Field, California 94035Kimberlee Shish§Millennium Engineering and Integration Company, Moffett Field, California 94035. 2017. Autonomous Flight Envelope Estimation for Loss-of-Control Prevention. *Journal of Guidance, Control, and Dynamics* **40**:4, 847-862. [Abstract] [Full Text] [PDF] [PDF Plus]
2. Flight Safety 311-402. [CrossRef]
3. Sweewarman Balachandran, Ella. M. Atkins. 2015. Flight Safety Assessment and Management for Takeoff Using Deterministic Moore Machines. *Journal of Aerospace Information Systems* **12**:9, 599-615. [Abstract] [Full Text] [PDF] [PDF Plus]
4. Xieyu Xu, Lingyu Yang, Jing Zhang. 2015. MRAC Control with Prior Model Knowledge for Asymmetric Damaged Aircraft. *The Scientific World Journal* **2015**, 1-10. [CrossRef]