JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 40, No. 4, April 2017

Markov Decision Process Framework for Flight Safety Assessment and Management

Sweewarman Balachandran* and Ella Atkins†
University of Michigan, Ann Arbor, Michigan 48109

DOI: 10.2514/1.G001743

Loss of control is the most common precursor to aircraft accidents. This paper presents a flight safety assessment and management system aimed at mitigating loss-of-control risks. Nominally, flight safety assessment and management serves as a passive watchdog system. When loss-of-control scenarios are encountered, flight safety assessment and management issues resilient control overrides to restore a safe operational state. This paper formulates flight safety assessment and management as a Markov decision process to account for uncertainties in state evolution and tradeoffs between passive monitoring and safety-based override. To ensure unsafe states are unreachable, probabilistic constraints are incorporated into the Markov decision process formulation. The Markov decision process framework is applied to prevent loss-of-control events during takeoff. An abstract representation of the underlying state space is specified to minimize Markov decision process computational overhead and to facilitate understanding of the resulting policy. Flight safety assessment and management is evaluated in a runway overrun case study motivated by a real-world incident.

Nomenclature

$\mathcal{A}$	=	Markov decision process action set
F	=	feature
h_{eng}	=	engine health status
$\mathcal{L}$	=	$y-\psi$ abstract state
M	=	mode
NOOP	=	no operation
$\mathcal{P}$	=	$\theta-h$ abstract state
p, q, r	=	angular rates
$\mathcal{Q}$	=	$v-x$ abstract state
$\mathcal{R}$	=	reward function
$\mathcal{S}$	=	Markov decision process state set
$\mathcal{T}$	=	transition probabilities
TOGL	=	toggle
u, v, w	=	velocities in the aircraft body frame
$\mathcal{V}$	=	utility/value
V_{lof}	=	liftoff speed
V_R	=	takeoff rotation speed
V_1	=	takeoff decision speed
V_2	=	safe takeoff speed
$\bar{v}$	=	true airspeed
x, y, z	=	three-dimensional position
Y_w	=	runway half-width
γ	=	discount factor
$\delta_e, \delta_a, \delta_r, \delta_t$	=	elevator, aileron, rudder, and throttle control inputs
$\theta_{\text{TS}}, h_{\text{TS}}$	=	tail-strike pitch angle, tail-strike altitude
χ	=	state distribution
ψ_0	=	runway heading
ϕ, θ, ψ	=	roll, pitch, and yaw angles

Presented as Paper 2015-0115 at the AIAA Infotech@Aerospace Conference, Kissimmee, FL, 5–9 January 2015; received 30 September 2015; revision received 23 May 2016; accepted for publication 24 May 2016; published online 3 August 2016.

*Graduate Research Assistant, Department of Aerospace Engineering; swee@umich.edu.

†Professor, Department of Aerospace Engineering; ematkins@umich.edu.

I. Introduction

ADVANCED capabilities such as fly-by-wire avionics, triply redundant systems, and envelope protection logic have dramatically reduced the accident rate in modern commercial transport aircraft [1,2]. However, loss of control (LOC) remains a primary contributing factor for commercial and general aviation accidents. Over 4000 fatalities have been attributed to LOC in the past decade alone [3–5]. LOC often results from a chain of events initiated by adverse environmental conditions and onboard anomalies/failures followed by inappropriate crew inputs and vehicle upset. The complex dependencies between LOC factors make it difficult to construct a single intervention strategy for LOC prevention [4,6].

Belcastro and Jacobson proposed the concept of Aircraft Integrated Resilient Safety Assurance and Failsafe Enhancement (AIRSAFE) [6]. AIRSAFE combines online modeling, safety assessment, and resilient control to recover a stable flight state in cases with significant LOC risk. This architecture concept provides a general LOC prevention capability consistent with this work. Other researchers have focused on developing automation aids to reduce specific LOC risks. Gingras et al. developed the Icing Contamination Envelope Protection (ICEPro) system [7]. ICEPro helps identify degradations in airplane performance and flying qualities resulting from ice contamination, providing cues to pilots. Borst et al. introduced the aircraft Safety Augmentation System (SafAS) [8]. SafAS is an automated pilot support system that prevents aircraft from veering off course into hazards such as terrain, severe weather, restricted airspace, etc. The Runway Overrun Prevention System was introduced by Airbus to warn flight crews about degraded landing performance during final approach. Srivatsan et al. [9], Milligan et al. [10], and Zammit-Mangion and Eshelby [11] proposed systems that aid the crew in making a safe go–no-go decision to avoid runway overrun accidents during takeoff.

The Envelope-Aware Flight Management System (EA-FMS) was introduced in previous publications [12,13] to mitigate LOC risk (see Fig. 1). EA-FMS augments existing flight management system (FMS) capabilities through online system identification and envelope estimation, envelope-aware flight planning, and resilient control. The Flight Safety Assessment and Management (FSAM) module of EA-FMS is responsible for monitoring system state with respect to LOC constraints and activating overrides only when necessary to avoid or recover from LOC. FSAM overrides the pilot or the nominal autopilot with envelope-aware planning and control logic that can suitably prevent or recover from the impending LOC scenario. Previous work modeled FSAM as a deterministic Moore machine [12–14]. This paper explores the use of decision-theoretic planning to allow override actions to be optimized over a probabilistic model of

the overall system. A reward or cost function explicitly trades the cost of inaction with the cost of automatically switching between pilot and (autonomous) envelope-aware control authorities.

Decision-theoretic techniques have been used for the development and enhancement of the traffic collision avoidance system (TCAS). Kochenderfer et al. [15,16], Temizer [17], and Winder [18] have used the Markov decision process (MDP) or partially observable Markov decision process to design alerting systems that could warn the flight crew about imminent conflicts with other aircraft and issue conflict resolution advisories.

Rules for FSAM to switch between available controllers to mitigate risk could be encoded as finite-state machines. FSAM finite-state machines can be manually constructed by a system designer based on domain knowledge and empirical simulations [14]. In general, tools such as hybrid automata and reachability analysis can further guide the designer in defining appropriate switching strategies/rules [19]. Finite-state machines can also be synthesized from linear temporal logic specifications [20]. However, manually specifying a state machine can be inefficient when the machine needs to address a broad class of scenarios. Use of a planner [21–23] to generate rules that serve as a state machine to be executed can aid a user in handling larger state-space sizes. The Markov decision process (MDP) is a compelling planning tool because it can model uncertainty, reward and cost, and arbitrary state-space features in an optimization framework. This paper therefore uses an MDP to generate a lookup table that effectively specifies switching decisions for each state of the system.

This work presents a fully observable MDP formulation to enable FSAM to make control mode override decisions that prevent LOC scenarios. A single comprehensive MDP formulation over all possible interacting LOC factors is computationally intractable due to the complexity associated with a very large state space. However, the full MDP can be decomposed into several sublevel MDPs, where each sublevel MDP is responsible for preventing LOC for a specific phase of flight or specific suite of elevated risk factors. This paper contributes an MDP formulation to address common takeoff LOC events associated with runway excursions and improper rotations. A novel abstract representation of the underlying state space is developed based on takeoff flight envelopes. This abstraction reduces the size of the original state space and promotes better understanding of the resulting policy. This MDP formulation is extended with constraints to ensure that unsafe states are unreachable. Note that this paper only focuses on developing an MDP formulation that will enable selecting the appropriate control authority (i.e., pilot/autopilot versus envelope-aware) to prevent LOC. Suitable envelope-aware control, flight planning, and guidance laws that prevent constraint violations or recover from LOC situations have been proposed by others [24–29] and are not the focus of this paper.

The rest of this paper is organized as follows. Section II reviews the MDP, whereas Sec. III specifies an MDP to address a suite of takeoff LOC risk factors. Section IV illustrates example policies obtained from the MDP formulation. Section V discusses a constrained MDP framework. Section VI applies the takeoff FSAM MDP formulation to a real-world aviation incident. Sections VII and VIII provide a discussion and conclusions, respectively.

II. Background

A discrete-time fully observable MDP [30,31] is represented as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$, where $\mathcal{S}$ represents a finite set of all possible discrete system states; $\mathcal{A}$ represents a finite set of actions that can be executed; $\mathcal{T}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ represents the transition probabilities associated with transitions from a given state to another state when executing an action; and $\mathcal{R}: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ represents a reward function that assigns a finite real value to each state–action pair. Actions $a \in \mathcal{A}$ for each state $s \in \mathcal{S}$ are chosen such that they maximize the expected cumulative discounted reward function of the form

$$\mathcal{V}^{\pi}(s) = \mathbb{E}\left[\sum_{n=0}^{\infty} \gamma^n \mathcal{R}(s_n, a_n) | \pi, s_0 = s\right] \tag{1}$$

$\gamma \in (0, 1]$ is a discount factor to specify the relative value of short versus long-term rewards. π is a policy defined as $\pi: \mathcal{S} \to \mathcal{A}$. $\mathcal{V}^{\pi}(s)$ is the utility of state s due to policy π. The optimal policy $\pi^*(s)$ is given by

$$\pi^*(s) = \arg\max_a \left(\mathcal{R}(s, a) + \gamma \sum_{s'} \mathcal{T}(s, a, s') \mathcal{V}(s')\right) \tag{2}$$

$$\mathcal{V}(s) = \max_a \left(\mathcal{R}(s, a) + \gamma \sum_{s'} \mathcal{T}(s, a, s') \mathcal{V}(s')\right) \tag{3}$$

The optimal policy can be obtained using algorithms such as value iteration, policy iteration, or linear programming [31].

III. Markov Decision Process Formulation for Flight Safety Assessment and Management

FSAM MDP state must capture all information necessary to make risk-optimal override decisions. State features relevant to LOC risk assessment and decision making can be broadly classified within four main categories: aircraft dynamics and control F_1, aircraft and subsystem health F_2, human operator characteristics F_3, and environment characteristics F_4. Each state $s \in \mathcal{S}$ of the FSAM MDP formulation is represented by its composition $s = [F_1, F_2, F_3, F_4]$. Detailed description of each state feature can be found in [32].

A complete MDP formulation over all flight phases would be unreasonably large, particularly if continuous-valued state features are discretized over a fine grid. Instead, the ideal MDP can be decomposed into several smaller context-appropriate MDPs. A phase-of-flight decomposition facilitates customizing the MDP to address LOC scenarios related to a particular phase of flight. Furthermore, state-space size can be significantly reduced by mapping baseline state features into abstract features for a particular phase of flight, as is illustrated later for takeoff. Abstract state features are based on flight envelopes and their translation to a suitable reward or cost function. This work specifies an MDP formulation to handle common takeoff-related loss-of-control risks.

Takeoff and landing are the highest-risk phases of flight due to their proximity to the ground. Ninety-seven rejected takeoff (RTO) runway overrun accidents and incidents have been reported from 1960 to 2000, resulting in over 400 fatalities. A survey of causal factors for takeoff-related accidents is provided in [33]. The most common contributing factors to LOC during takeoff are improper rejected takeoff procedures, poor directional control, and inappropriate takeoff configuration. This paper develops an MDP to address high-risk takeoff LOC scenarios, including runway excursion and unsafe liftoff states. The MDP presented in this paper is applicable given the following assumptions.

1) All actuators, sensors, and aircraft systems are functioning nominally.

2) There are no increased weather-related risks; other aircraft and obstacles remain clear of the runway and departure path.

3) Aircraft flight envelopes remain constant throughout takeoff, implying no change in airframe or performance characteristics.

With these assumptions, the flight crew characteristics F_3 and environmental features F_4 remain constant for the takeoff MDP formulated in this paper. In the aircraft health feature F_2, this work models the engine status $\bar{E}$ to capture engine failure associated LOC events during takeoff. The remaining subfeatures in F_2 are assumed to remain constant. These assumptions must be relaxed in future work, which must also consider appropriate decompositions based on specific hazard scenarios to ensure that the takeoff MDP is tractable.

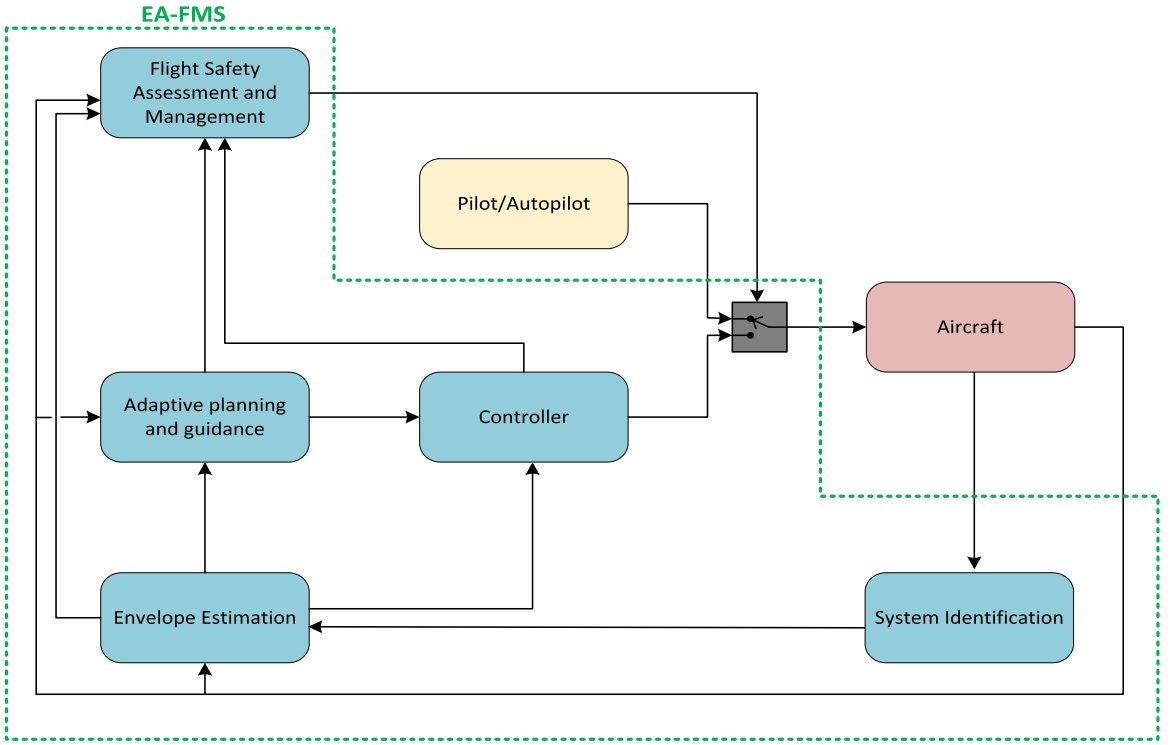

Fig. 1 Envelope-aware flight management system.

A. State Formulation for Takeoff

High-risk LOC scenarios such as runway overruns and improper rejected takeoffs are captured in aircraft longitudinal dynamics and runway position constraints. Events such as improper rotations and tail strikes are associated with pitch dynamics, whereas runway lateral excursion events are associated with lateral or directional dynamics. The relevant aircraft dynamics states considered for the takeoff MDP formulation are aircraft velocity $V = \sqrt{u^2 + v^2 + w^2}$; pitch θ; heading ψ; position x, y, z with respect to the runway; control mode M; mode select switch status $\bar{S}$; throttle control input $\bar{T}$; and engine health status $\bar{E}$. MDP state is given as

$$s \in \mathcal{S}, \quad s = [V, \theta, \psi, x, y, z, \bar{T}, \bar{M}, \bar{S}, \bar{E}] \tag{4}$$

This state-space formulation is infinite due to continuous variables such as position, airspeed, and pitch. Knowledge of aircraft takeoff dynamics and aircraft envelopes is exploited to combine the continuous-valued state variables into abstract state features.

Aircraft takeoff envelopes are analyzed with respect to translational, rotational, and lateral dynamics. In a nominal takeoff, the aircraft accelerates to liftoff speed from rest, lifting off and accelerating to speed V_2 before the end of the runway. In case of engine failure during the takeoff ground roll, a rejected takeoff is warranted unless airspeed is too high and insufficient runway distance remains. Rejecting versus continuing a takeoff following an engine failure was previously analyzed using simplified equations of motion for takeoff [14]. Figure 2 is a vector field that illustrates the evolution of the $V - x$ dynamics under a rejected takeoff scenario [14]. In Fig. 2, rejecting the takeoff at an airspeed–position state below the solid curve leads to trajectories that decelerate and stop within the available runway length. This action would correspond to a safe rejected takeoff. Rejecting the takeoff at a state above the solid curve results in the aircraft overrunning the remaining runway, representing an unsafe rejected takeoff. A similar analysis can be done for the continued takeoff case (see Fig. 3).

If an engine failure occurs at a point below the solid curve, the airplane has sufficient airspeed to accelerate to lift off speed and reach the V_2 airspeed. However, if an engine failure occurs at a point above the solid curve, the airplane has insufficient airspeed to accelerate to V_2 before the runway overrun. Combining the curves in Figs. 2 and 3 yields four distinct regions shown in Fig. 4. Clearly, a region exists

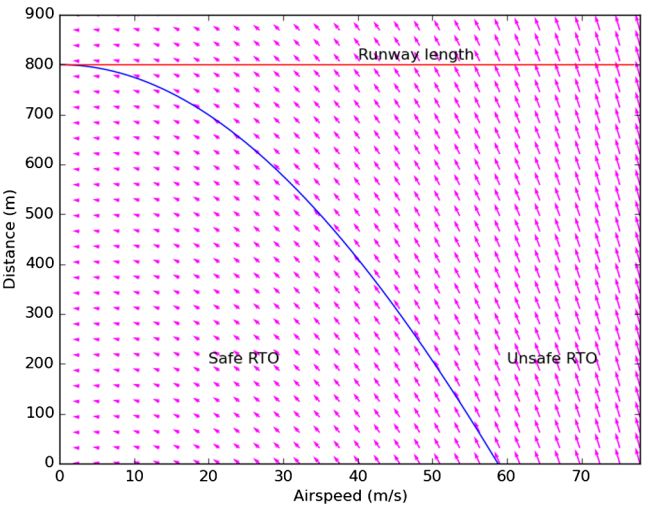

Fig. 2 Rejected takeoff envelope.

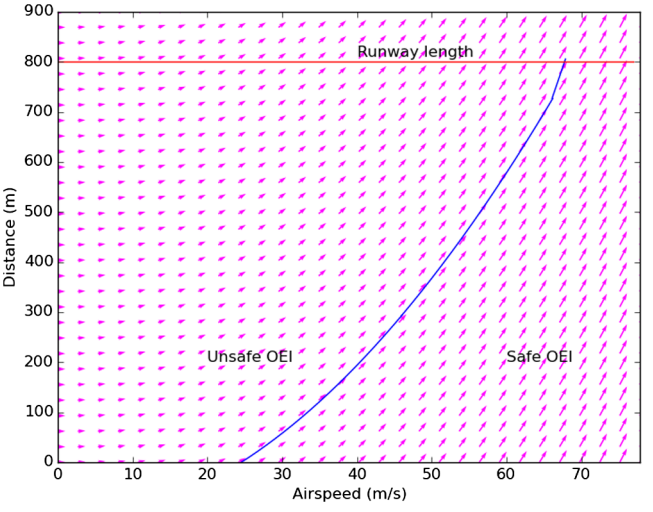

Fig. 3 One-engine inoperative envelope.

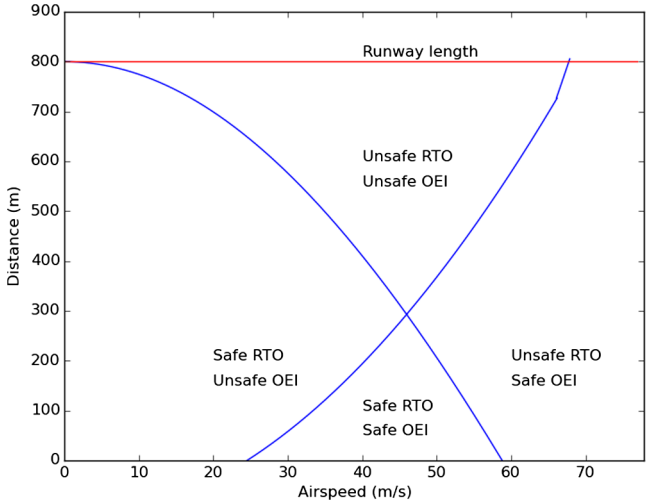

Fig. 4 RTO and OEI envelopes: safe vs unsafe zones.

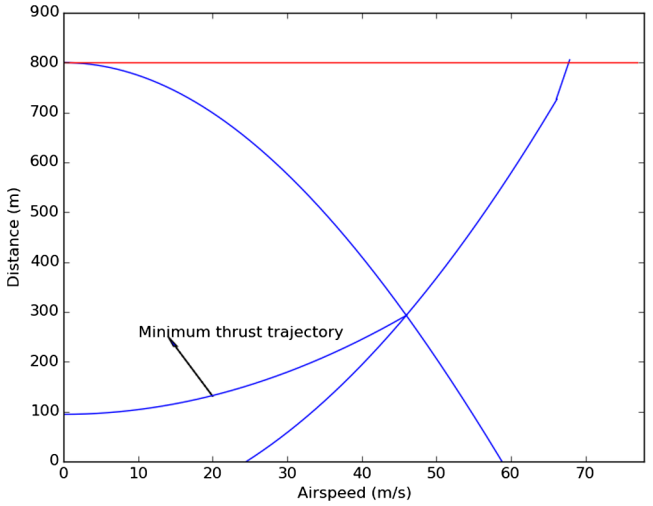

Fig. 5 Minimum thrust trajectory for safe takeoff.

where neither a rejected takeoff nor a continued takeoff is safe; this region must be avoided at all times. One can estimate the minimum thrust required to prevent the aircraft from entering this unsafe region. The resulting minimum thrust trajectory is shown in Fig. 5. Each curve in Fig. 5 can be described by polynomials of the form $x = \bar{a}_0 + \bar{a}_1 V + \bar{a}_2 V^2 + \bar{a}_3 V^3$, with coefficients $\bar{a}_0, \ldots, \bar{a}_3$ chosen appropriately. Let V_{EF} denote the smallest airspeed at which a takeoff can be continued following an engine failure at $x = 0$. Let V_1 denote the airspeed at the intersection of the three curves. Let X_{V_1} denote the corresponding distance on the runway, and let X_{length} denote the length of the runway. With these parameters, the V and x states can be aggregated into 17 abstract states, as shown in Fig. 6a. Note that states 15 and 16 in Fig. 6a represent runway overrun scenarios where the aircraft has crossed the available takeoff distance with inappropriate airspeed to either take off or stop safely.

Envelopes for the rotational and lateral dynamics are constructed based on geometric constraints. Increasing the pitch attitude beyond a certain pitch angle results in a tail strike. Thus, care must be taken to prevent tail strikes during rotation. Let $\theta \geq \theta_{\mathrm{TS}}$, $z \leq h_{\mathrm{TS}}$ denote the condition at which a tail strike occurs, where θ_{TS} is the tail-strike pitch attitude when the aircraft is below altitude h_{TS}. Let $\theta_1 = 0.2\theta_{\mathrm{TS}}$ and $\theta_2 = 0.8\theta_{\mathrm{TS}}$. With these parameters, pitch-altitude space is aggregated as shown in Fig. 6b.

Figure 7 illustrates geometric constraints for the lateral dynamics imposed by the available runway width. Here, the cross-track position and heading are combined into a single feature. Let Y_w represent the half-width of the runway. Let $Y_1 = Y_w$, $Y_2 = 0.5Y_w$. Let ψ_0 represent the runway heading. Let $\psi_1 = \psi_0 + 4$ deg and $\psi_2 = \psi_0 + 10$ deg. With these parameters, partitions of the lateral displacement and yaw space are obtained as shown in Fig. 8.

Thrust control inputs for takeoff are discretized as $\bar{T} \in \{T_{\mathrm{idle}}, T_{\mathrm{max}}\}$. In this work, the two available control authorities are the nominal pilot/autopilot P and the envelope-aware controller EA such that $\bar{M} \in \{P, \mathrm{EA}\}$. The engine health status is discretized as $\bar{E} \in \{E_{\mathrm{AEO}}, E_{\mathrm{OEI}}, E_{\mathrm{AEI}}\}$, where E_{AEO} represents "all engines operational" (AEO), E_{OEI} represents "one engine inoperative" (OEI), and E_{AEI} represents "all engines inoperative" (AEI). With the compact state features described previously, the initial state formulation in Eq. (4) is transformed into

$$
\begin{aligned}
&s \in \mathcal{S},\\
&s = [\bar{Q}, \bar{P}, \bar{L}, \bar{T}, \bar{E}, \bar{S}, \bar{M}],\\
&\bar{Q} \in \{q_1, q_2, \ldots, q_{16}\},\\
&\bar{P} \in \{p_1, p_2, \ldots, p_8\},\\
&\bar{L} \in \{l_1, l_2, \ldots, l_{25}\},\\
&\bar{T} \in \{T_{\mathrm{idle}}, T_{\mathrm{max}}\},\\
&\bar{E} \in \{E_{\mathrm{AEO}}, E_{\mathrm{OEI}}, E_{\mathrm{AEI}}\},\\
&\bar{S} \in \{P, \mathrm{EA}\},\\
&\bar{M} \in \{P, \mathrm{EA}\}
\end{aligned}
\tag{5}
$$

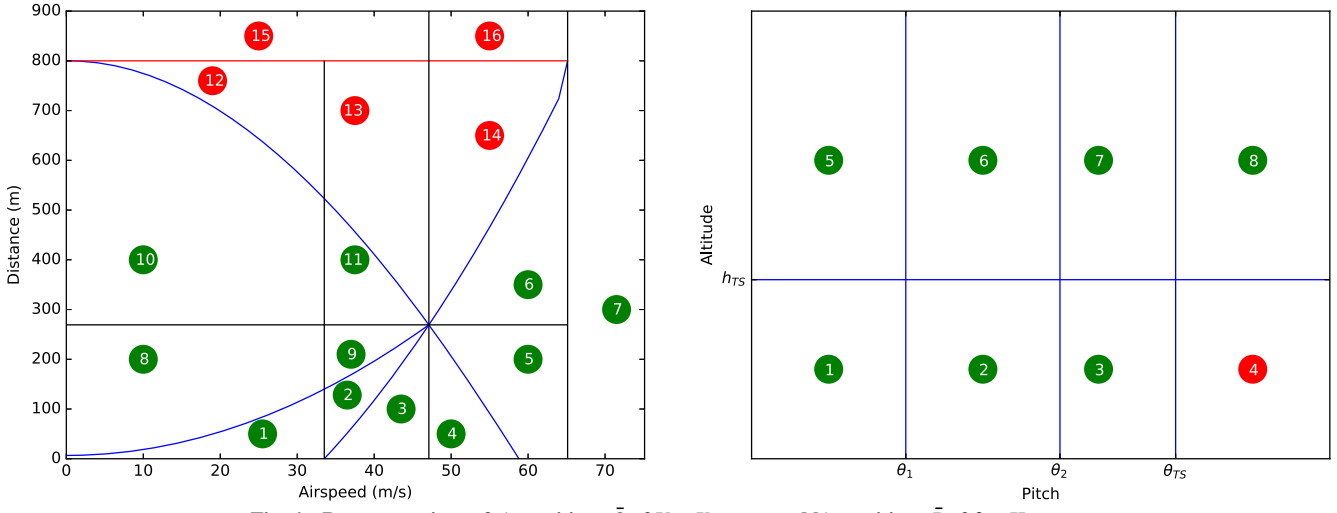

Fig. 6 Representations of a) partitions $\bar{Q}$ of $V - X$ space, and b) partitions $\bar{P}$ of $\theta - H$ space.

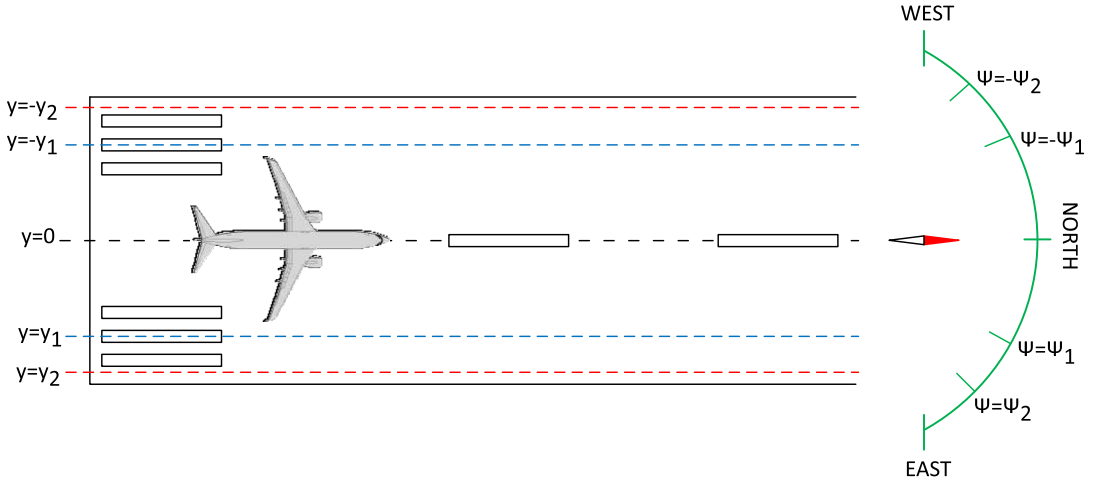

Fig. 7 Lateral constraints.

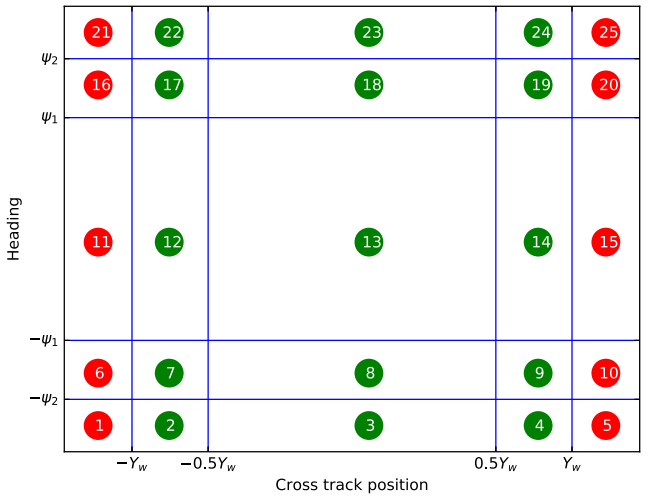

Fig. 8 Partitions $\bar{L}$ of $Y - \psi$ space.

Note that $\bar{Q}$ is an abstraction of aircraft velocity V and longitudinal position x. $\bar{P}$ is an abstraction of pitch attitude θ and altitude z. $\bar{L}$ is an abstraction of cross-track position y and heading ψ.

B. Action Formulation for Takeoff

FSAM is a high-level watchdog system that passively monitors the various state features for LOC risk. If sufficient time and margin exist for the flight crew to mitigate any elevated LOC risk factors, FSAM continues to remain passive. FSAM issues override decisions only when switching to the envelope-aware controller would enable LOC prevention or recovery. FSAM then returns control back to the pilot and nominal autopilot once LOC risk is lowered to acceptable levels.

The FSAM MDP selects from two actions: no operation (NOOP) and toggle (TOGL). Any time FSAM selects NOOP, current control mode $\bar{M}$ is likely to remain engaged. If the current control mode $\bar{M}$ indicates nominal pilot/autopilot authority and FSAM selects the TOGL action, FSAM activates the envelope-aware controller. If the current control mode $\bar{M}$ is the envelope-aware controller and FSAM selects the TOGL action, authority is returned to the nominal pilot/autopilot system. The pilot could also manually request activation of the envelope-aware controller or transfer of control authority from the envelope-aware control via the mode select switch $\bar{S}$. The MDP actions are described as follows:

$$\bar{A} \in \{\text{NOOP}, \text{TOGL}\} \tag{6}$$

C. Reward Formulation for Takeoff

FSAM MDP reward is formulated as a cost function (negative reward) that penalizes unsafe aircraft states but also discourages the routine selection of the toggle action. A weighted sum reward formulation is proposed:

$$\mathcal{R}(s,a) = \sum_{i=0}^{n} \eta_i \mathcal{R}_i(s,a) \tag{7}$$

$\mathcal{R}_i(s,a)$ penalize unsafe states and unnecessary toggle actions, whereas η_i represent tunable weighting parameters that may vary depending as a function of flight mode. For example, the penalty for violating an airspeed or angle of attack stall constraint at high altitude can be lower than the stall penalty at low altitude due to the availability of recovery margin. Weighting parameters may be learned from accident flight data and investigation board recommendations.

In this work, the additive reward formulation is defined as in Eq. (7):

$$\mathcal{R}(s,a) = \eta_1 \mathcal{R}_1(\bar{Q}) + \eta_2 \mathcal{R}_2(\bar{P}) + \eta_3 \mathcal{R}_3(\bar{L}) + \eta_4 \mathcal{R}_4(\bar{M}, \bar{A}) \tag{8}$$

Here, $\mathcal{R}_1(\mathcal{Q})$ penalizes unsafe states with respect to the translational dynamics (see Fig. 6) and is given by

$$\mathcal{R}_1(\bar{Q}) = \begin{cases} -1 & \text{if } \bar{Q} \in \{q_{15}, q_{16}\} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$\mathcal{R}_2(\mathcal{P})$ penalizes unsafe states with respect to the rotational dynamics:

$$\mathcal{R}_2(\bar{P}) = \begin{cases} -1 & \text{if } \bar{P} \in \{p_4\} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$\mathcal{R}_3(\bar{L})$ penalizes unsafe states with respect to the lateral dynamics (see Fig. 8):

$$\mathcal{R}_3(\bar{L}) = \begin{cases} -1 & \text{if } \bar{L} \in \{l_1, l_5, l_6, l_{10}, l_{11}, l_{15}, l_{16}, l_{20}, l_{21}, l_{25}\} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$\mathcal{R}_4(\bar{M}, \bar{A})$ penalizes unnecessary toggle actions to discourage frequent mode switches and the resulting mode confusion. Staying in the envelope-aware control mode when the pilot requests pilot mode is also penalized to encourage transfer of control authority to the pilot once the high-risk LOC scenario is averted. Thus,

$$\mathcal{R}_4(\bar{M},\bar{A}) = \begin{cases} -1 & \text{if } \bar{M} = P \wedge \bar{S} = P \wedge \bar{A} = \text{TOGL} \\ -o_1 & \text{if } \bar{M} = \text{EA} \wedge \bar{S} = P \wedge \bar{A} = \text{NOOP} \\ -o_2 & \text{if } \bar{M} = \text{EA} \wedge \bar{S} = \text{EA} \wedge \bar{A} = \text{TOGL} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

where $0 \leq o_1 \leq 1$ and $0 \leq o_2 \leq 1$. η_i represent positive weights per Eq. (8). For this work, the authors manually tuned reward function weights to ensure that policies favored pilot control but did not allow the system to violate constraints.

Statistics can assist in computing reward weights. For example, the Flight Safety Foundation [33] reports that runway overruns and lateral runway excursions have given rise to a larger number of fatal accidents than tail-strike events during takeoff. Consequently, for the takeoff MDP, the values of the weighting parameters on $\mathcal{R}_1$ and $\mathcal{R}_3$ are set significantly higher than the weight on $\mathcal{R}_2$. The choice of the weight on $\mathcal{R}_4$ may be guided by human subject experiments and pilot preferences; for this work, it is assumed the pilot will prefer to assume control whenever constraints are not otherwise violated. Methods presented in [34] can also be adapted to compute reward function parameters in future work.

D. Transition Probabilities

The transition probabilities are obtained using Monte Carlo simulations. The Monte Carlo simulation framework used in this work is documented in Appendix B. States are sampled from the MDP state

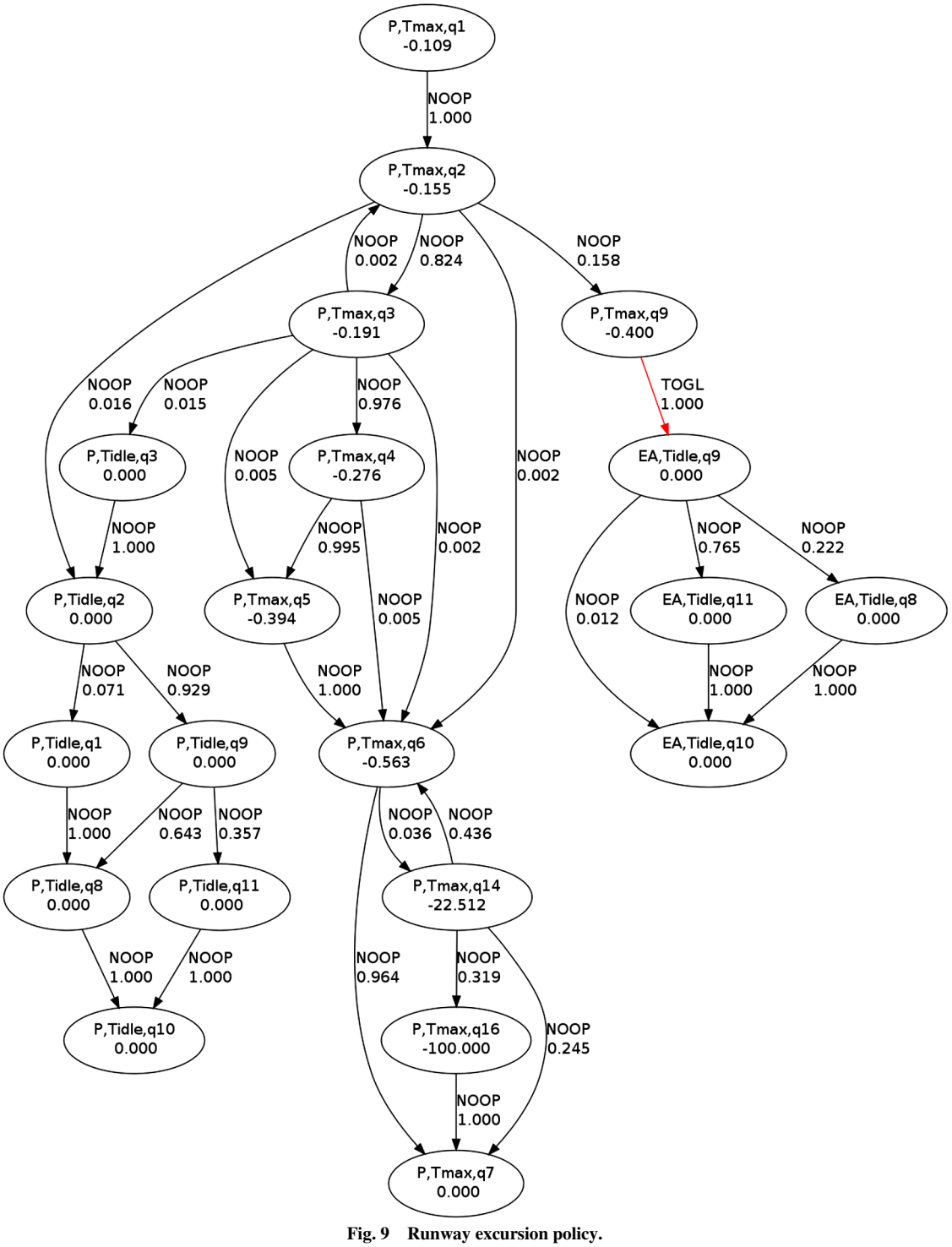

Fig. 9 Runway excursion policy.

space $\mathcal{S}$ described in Eq. (5), and their corresponding transition probabilities $\mathcal{T}$ under a given control authority are estimated as follows:

$$\mathcal{T}(s_k|s_i)|_M = \frac{N(s_i, s_k)}{\sum_{s_l \in \mathcal{S}} N(s_i, s_l)} \tag{13}$$

where $N(s_i, s_k)$ is the total number of transitions from state s_i to s_k under control authority $M \in \{P, \text{EA}\}$. With the aforementioned information, the transition matrix for each action is defined as follows. Let $\mathcal{T}_{\text{NOOP}}$ denote the transition probability matrix for $a = \text{NOOP}$. Let $\mathcal{T}_{\text{TOGL}}$ denote the transition probability matrix for $a = \text{TOGL}$. The state features in Eq. (5) are permuted such that $\mathcal{T}_{\text{NOOP}}$ and $\mathcal{T}_{\text{TOGL}}$ can be viewed as block diagonal matrices of the form

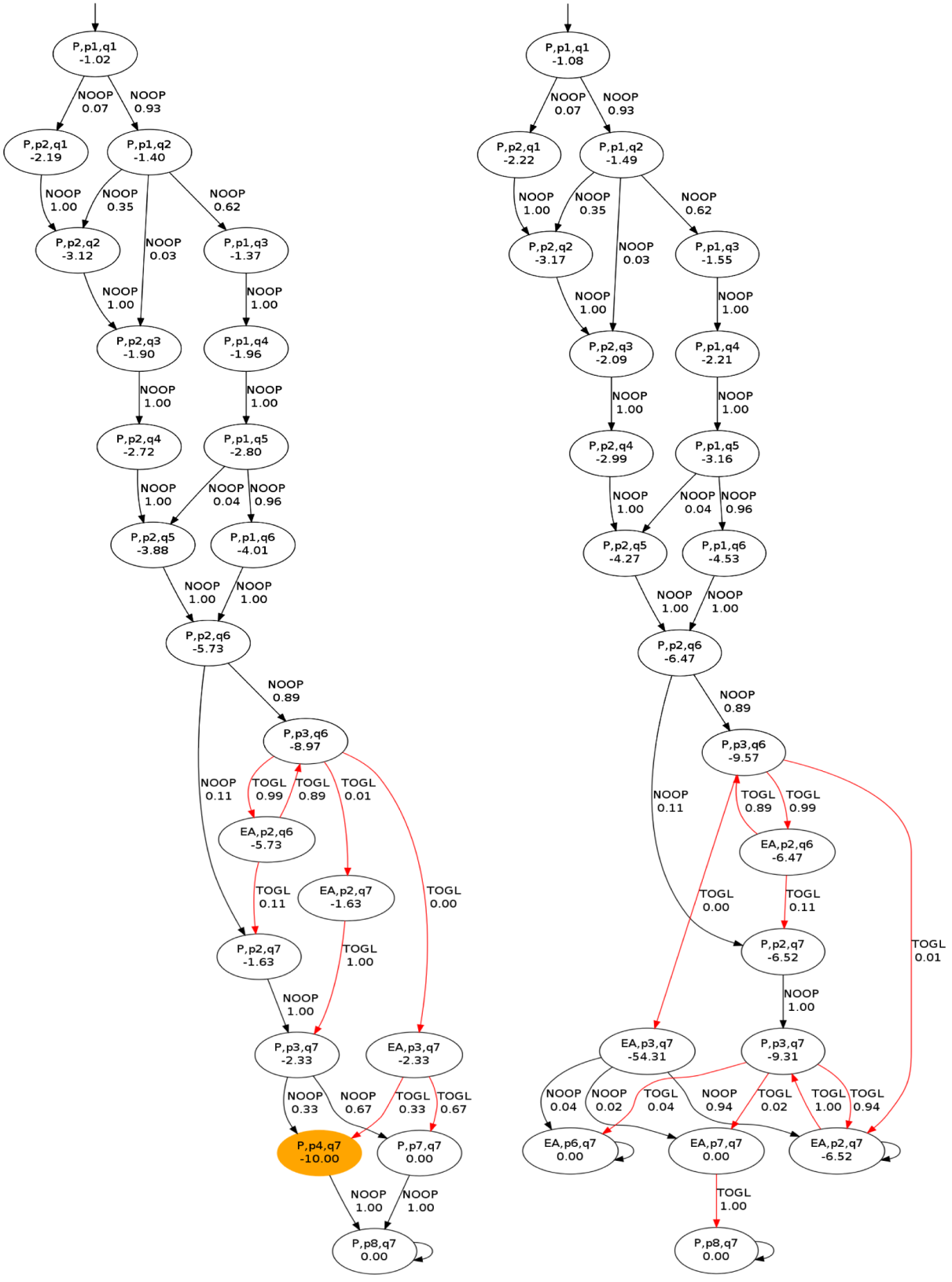

Fig. 10 Tail-strike policy: a) policy obtained from MDP and b) policy obtained from CMDP.

$$\mathcal{T}_{\text{NOOP}} = \begin{bmatrix} \mathcal{T}|_{M=P} & 0 \\ 0 & \mathcal{T}|_{M=\text{EA}} \end{bmatrix} \tag{14}$$

$$\mathcal{T}_{\text{TOGL}} = \begin{bmatrix} 0 & \mathcal{T}|_{M=\text{EA}} \\ \mathcal{T}|_{M=P} & 0 \end{bmatrix} \tag{15}$$

With the aforementioned states, actions, rewards, and transition probabilities, the takeoff MDP is optimized using value iteration.

IV. Takeoff Markov Decision Process Policies

The total number of states in the takeoff MDP formulation is given by the product of sizes of the individual state features. Thus, there are 76,800 states in the preceding takeoff MDP formulation. The optimal policy for the takeoff MDP is stored as a lookup table mapping an optimal action to each state. This section constructs a Markov chain to facilitate MDP policy understanding.

Let $\mathcal{T}^i$ represent the ith row of transition matrix $\mathcal{T}$. The transition probability matrix for the MDP policy is constructed as follows:

$$\mathcal{T}^i_\pi = \begin{cases} \mathcal{T}^i_{\text{NOOP}} & \text{if } \pi(i) = \text{NOOP} \\ \mathcal{T}^i_{\text{TOGL}} & \text{if } \pi(i) = \text{TOGL} \end{cases} \tag{16}$$

Transition matrix $\mathcal{T}_\pi$ represents the Markov chain of policy π. The probability distribution over the states reached after n steps (χ_n) while starting from a given initial state distribution χ_0 and following policy π is

$$\chi_n = \chi_0^T \mathcal{T}_\pi^n \tag{17}$$

The Markov chain representing the complete policy is also difficult to visualize, and so segments of the policies as used to illustrate their properties. Figure 9 presents a policy segment that illustrates FSAM MDP policy response to an imminent runway excursion risk.[‡] For ease of illustration, only transitions in $\bar{M}$, $\bar{T}$ and $\bar{Q}$ are shown. Each node represents a discrete state s annotated with features and optimal value $\mathcal{V}(s)$. Edges represent transitions between discrete states and are labeled with the optimal action and transition probability. The policy chooses NOOP if the pilot is in control and the aircraft remains inside the safe takeoff envelope with sufficient margin. When the aircraft enters an unsafe region (e.g., $\bar{Q} = q_9$) with imminent runway overrun risk, the policy chooses TOGL to transfer authority to the envelope-aware controller, which then rejects the takeoff by reducing thrust to idle (T_{idle}) to ensure that the aircraft remains within the safe operating envelope. Policy behavior vary depending on the choice of weighting factors η in Eq. (8). For example, increasing the penalty on envelope-aware states (i.e., $\bar{M} = \text{EA}$) in Eq. (12) can result in transfer of control back to the pilot immediately. The following example illustrates the tradeoff between increasing the cost of NOOP versus the cost of TOGL.

Figure 10a presents an FSAM MDP policy segment showing response to a tail-strike risk. For ease of visualizing this policy, the runway excursion/overrun risks states have been pruned in Fig. 10a.[§] Only state transitions impacting risk level, specifically $\bar{M}$, $\bar{P}$, and $\bar{Q}$, are shown. The aircraft starts from rest at the beginning of the runway, with the pilot in control, and accelerates as throttles are set to takeoff thrust (P, p_1, q_1). Figure 10a illustrates the probable transitions from this initial state. From rest until the rotation airspeed V_R is reached, the policy does not interfere with the crew operations because the aircraft is within the safe $V - x$ and $\theta - h$ envelopes. However, during rotation, FSAM activates envelope-aware control at (P, p_3, q_6) to prevent excessive rotation and the subsequent tail strike (i.e., $\bar{P} = p_3$). The envelope-aware control law reduces pitch attitude to prevent tail strike during rotation. However, because of a large penalty on the envelope-aware controller state (i.e., $\bar{M} = \text{EA}$), control is returned to the pilot immediately, as would be the case in an automobile when antilock brakes or traction control systems temporarily engage. This policy favors the pilot control model, but note that the aircraft still has a tail-strike risk. The tail-strike risk can be eliminated by choosing a higher weighting factor for the R_2 term than R_4 in Eq. (8).

The preceding policy segments illustrated the FSAM responses to specific loss-of-control situations. The complete FSAM MDP policy manages combinations of elevated risks associated with runway excursions and overruns as well as potential tail strikes by assuring that inappropriate longitudinal and lateral control inputs are overridden in time to avoid LOC. The full policy must ultimately be verified to ensure that unsafe states are unreachable. For the nominal MDP formulation, this requires manually tuning reward weighting factors and regenerating policies to ensure that the desired behavior is obtained. This process can be cumbersome, especially if the underlying state space is large. To overcome this difficulty, the following section proposes an MDP formulation with constraints.

V. Constrained Markov Decision Process

A constrained Markov decision process (CMDP) formulation enables FSAM to make risk-optimal decisions subject to upper bounds on the probability of entering a LOC risk state. The CMDP policy aims to maximize the expected cumulative discounted reward function [Eq. (1)] subjected to constraints of the form

$$\begin{aligned} &\mathcal{T}(s_1^*|s_0^*) \le c_1, \\ &\mathcal{T}(s_2^*|s_0^*) \le c_2, \\ &\vdots \\ &\mathcal{T}(s_n^*|s_0^*) \le c_n \end{aligned} \tag{18}$$

where $\mathcal{T}(s_i^*|s_0)$ is the conditional probability of entering an unsafe state s_i^* from a given initial state s_0. Each $c_i \in [0, 1]$ represents a probability upper bound. Equation (18) can be expressed in terms of state–action frequencies. This facilitates solving the constrained MDP formulation using a linear programming framework (see Appendix A for more details on the linear programming CMDP formulation for FSAM). Figure 10b illustrates the tail-strike policy segment constructed using the CMDP. The probability of entering a tail-strike state (i.e., $\bar{P} = p_4$) is constrained to be zero rather than imposing a cost penalty on tail-strike states. The CMDP policy is

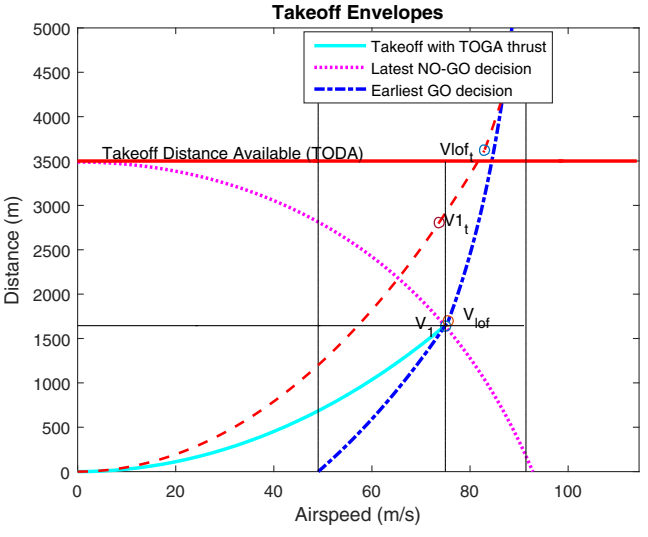

Fig. 11 Trajectory of Flight 407 (TOGA, takeoff/go around).

[‡]The weight parameters for this policy were $\eta_1 = 100$, $\eta_4 = 0.4$, $\eta_2 = \eta_3 = o_1 = o_2 = 0$. A discount factor of $\gamma = 0.7$ was used. $\bar{S} = P$ remained constant throughout the Monte Carlo simulations.

[§]The transition probabilities for this policy were reconstructed such that there were no runway excursions/overrun risks. The weight parameters used were $\eta_2 = 10, \eta_4 = 5, \eta_1 = \eta_3 = 0, o_1 = 0.5, o_2 = 0$. Thus, Figs. 9 and 10 represent segments of two different policies.

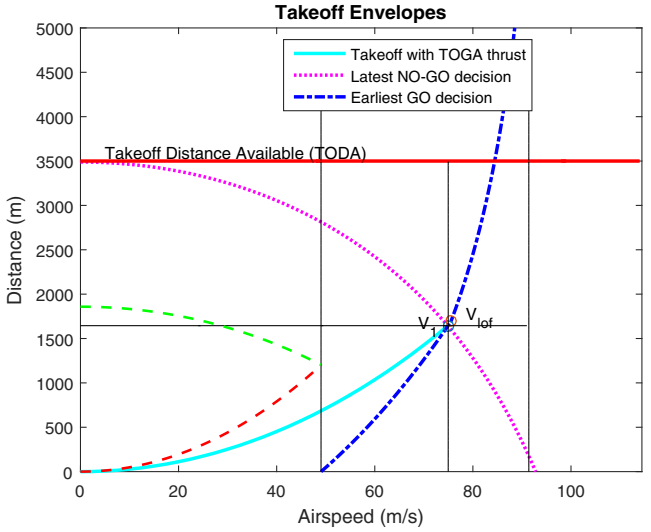

Fig. 12 MDP policy applied to Flight 407 (TOGA, takeoff/go around).

similar to the MDP policy in Fig. 10a until the initial override from *P* to EA. However, the CMDP policy overrides the pilot when there is an imminent tail-strike risk and retains control until the probability of entering a tail-strike state is zero.

VI. Case Study

On 20 March 2009, an Airbus A340 operated by Emirates Airlines failed to take off safely from Melbourne Airport, Australia [35]. The flight crew had programmed the flight computer with the wrong weight calculations, which resulted in poor takeoff performance due to inadequate thrust. Consequently, the aircraft overshot the runway during the initial takeoff roll and experienced a tail strike due to overrotation. The subsequent departure was uneventful, and the aircraft returned to the airport for an emergency landing. The actual weight of the aircraft was 362.9 tons, but the weight entered into the flight computer was 262.9 tons. Figure 11 illustrates the takeoff envelopes of the aircraft for the weight that was entered into the flight computer (262.9 tons). If the aircraft was actually loaded at 262.9 tons, it would have followed the green trajectory in Fig. 11, remaining within safe operating envelope regions. The dashed curve in Fig. 11 indicates the actual aircraft trajectory (weighing 362.9 tons) from

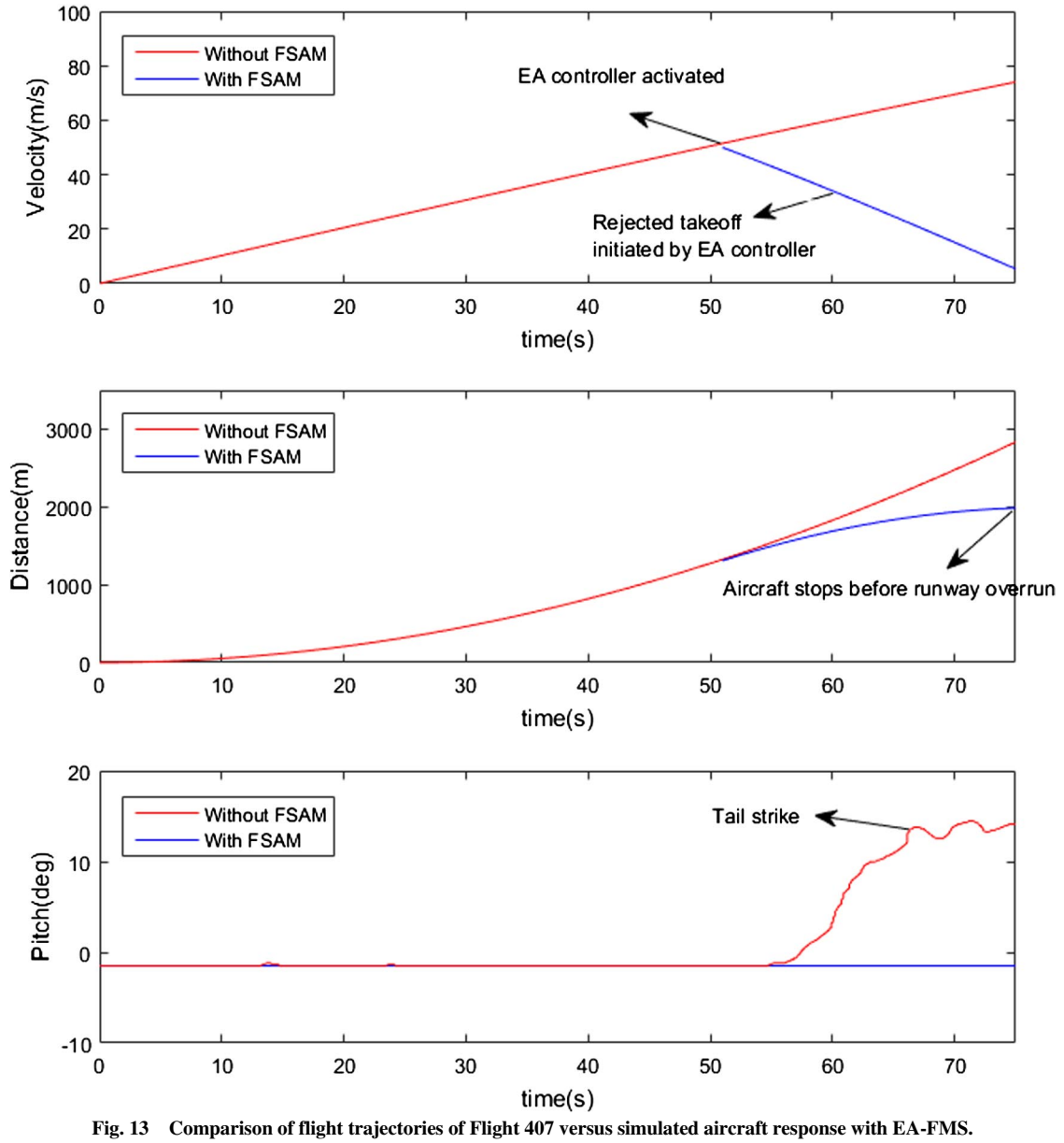

Fig. 13 Comparison of flight trajectories of Flight 407 versus simulated aircraft response with EA-FMS.

flight recorder data. Because of the data entry error, the aircraft began its ground roll with a thrust setting that was too low for the higher takeoff weight, resulting in insufficient acceleration to attain liftoff speed V_{lof} before overshooting the runway.

Figure 12 illustrates the application of the policy developed using the MDP framework described for the preceding accident scenario. As the accelerating aircraft enters $\bar{Q} = q_9$, an unsafe region in the $V - x$ envelope, FSAM overrides the pilot with the envelope-aware controller, which then rejects the takeoff. The aircraft then decelerates and stops safely in $\bar{Q} = q_{17}$ well before the runway threshold. Figure 13 compares the aircraft states of Flight 407 modeled from data obtained from accident reports [35] and the simulated aircraft response to the takeoff MDP policy developed in this paper.

VII. Discussion

The preceding case study illustrates how an MDP can be constructed for a particular set of LOC risks during takeoff. Separate FSAM MDPs can be constructed for each phase of flight because only one phase will be active at a time. When reducing the set of assumptions, the remaining suite of state features in F will increase complexity substantially relative to this paper's case study. Conditional independence will be critical to exploit as will additional state-space abstractions.

FSAM must sufficiently capture the capabilities and limitations of the envelope-aware controller to recognize high-risk situations needing envelope-aware control and situations, where it is best to leave the pilot in control. Interaction between pilot and FSAM is also important to more fully characterize. Certainly, if FSAM and flight crew agree on control mode, the decision is straightforward. FSAM override of the pilot's designated mode, whether toggling to envelope-aware (for recovery) or back to pilot control (following recovery), requires further research in human factors as well as in autonomous system development, validation, and verification.

For the purpose of illustration, Monte Carlo simulations used in this paper were constructed such that the probability of entering unsafe states was higher under pilot control. This may not always be the case. For example, instrument malfunctions may render the safety controller ineffective, in which case the sensor health feature can bias the MDP away from an autonomous control mode selection. In addition to Monte Carlo simulations, accurate state transition probabilities can be constructed from flight data. Care must be taken to model the pilot's inputs adequately because flight data only represent one crew input case. Mining larger data sets (e.g., data from all of an airline's flights over a multiyear period) can be used to determine statistically significant state-space transition dynamics and probabilities.

The policies obtained from the MDP/CMDP formulations are stored in the form of lookup tables. Verifying large lookup table policies can be computationally intensive. Deterministic MDP policies can be verified using model checking tools such as SPIN [36] and NuSMV [37]. For large-scale MDPs, probabilistic verification algorithms described in [38,39] can be used to establish formal guarantees. PRISM [40] can be used to verify stochastic MDP policies. Note that FSAM is only an overriding mechanism. Thus, if the available control authorities cannot mitigate a given LOC risk, the MDP policy can result in unsafe states.

In this work, optimal policies were constructed using a value iteration algorithm that explicitly enumerates all states. This may be infeasible for large state spaces. Instead, a modified form of value iteration can be used to only enumerate states that are reachable from a given initial state [41]. The availability of a simulation model for takeoff dynamics makes it possible to use reinforcement-learning techniques such as temporal difference/Q-learning to solve the underlying MDP [41,42]. The use of a Monte Carlo tree search algorithm to solve the ideal MDP formulation in an online fashion was explored in a separate publication [32].

VIII. Conclusions

This paper contributes a decision-theoretic formulation of a Flight Safety Assessment and Management (FSAM) system that monitors each flight and activates an envelope-aware controller under high-risk conditions. A generalized suite of MDP state features and reward formulation were proposed, and a takeoff case study was formulated in detail. Specifically, this paper develops a takeoff MDP capable of preventing LOC events such as runway excursion and tail strike and demonstrates its ability to avoid LOC on a real-world accident case. Intuitive state-space abstractions enabled the FSAM takeoff MDP to remain computationally tractable. A CMDP formulation eliminates the need to iteratively refine MDP policies by imposing probabilistic constraints on high-risk states.

Previous work [14] formulated FSAM as a suite of manually constructed finite-state machines to govern control authority switching. Manually generating finite-state machines can be cumbersome if the underlying state space is large and requires significant experience to ensure that the override directives are chosen appropriately. This paper has shown that an MDP or CMDP FSAM formulation can eliminate the need to manually design finite-state machines for managing control authority switches. Furthermore, any MDP formulation enables each policy to be optimized over uncertainties and generalized reward functions.

Despite the generality of the initial FSAM MDP formulation, this paper makes several assumptions about pilot models, environment, and aircraft health in the takeoff case study. Extending the specific FSAM MDP models to not require these assumptions is essential to ensure that FSAM policies do not actually increase risk when assumptions no longer hold. Future research will formally analyze additional scenarios over the full state space and develop strategies to ensure that the actions of FSAM will not jeopardize nominal operations of the aircraft. Case studies demonstrating the integration of multiple EA-FMS modules to address LOC scenarios related to an in-flight rudder jam are presented in complementary work [43].

Appendix A: Constrained Markov Decision Process

The expected value or utility of state s_0 when acting according to policy π is given by

$$\mathcal{V}(s_0)_\pi = \mathbb{E}\left[\sum_{n=0}^{\infty} \lambda^n \mathcal{R}(s_n, a_n)\right]_{s_0} \tag{A1}$$

For a Markov process, Eq. (A1) can be expressed as

$$\begin{aligned}\mathcal{V}(s_0) &= \sum_{s_i \in s} \sum_{a_j \in \mathcal{A}} \sum_{n=0}^{\infty} \lambda^n \mathcal{T}(s_n = s_i, a_n = a_j | s_0) \mathcal{R}(s_n = s_i, a_n = a_j) \\ &= \sum_{s_i \in s} \sum_{a_j \in A} \rho(s_i, a_j)^\pi_{s_0} \mathcal{R}(s_n = s_i, a_n = a_j)\end{aligned} \tag{A2}$$

Here, $\rho(s_i, a_j)^\pi_{s_0}$ is defined as the occupational measure of the state–action pair (s_i, a_j):

$$\rho(s_i, a_j)^\pi_{s_0} := \sum_{n=0}^{\infty} \lambda^n \mathcal{T}(s_n = s_i, a_n = a_j | s_0) \tag{A3}$$

The occupational measure is the discounted total probability of reaching a state s_i and executing an action a_j as a result of starting in state s_o and acting according to policy π. The sum of the occupational measure of state a_i over all possible actions $a_j \in \mathcal{A}$ is obtained from Eq. (A3):

$$\sum_{a_j\in\mathcal{A}}\rho(s_i,a_j)=\sum_{a_j\in\mathcal{A}}\sum_{n=0}^{\infty}\lambda^n\mathcal{T}(s_i,a_j|s_0)$$
$$=\mathcal{T}(s_0)+\sum_{s_x\in\mathcal{S}}\sum_{a_y\in\mathcal{A}}\sum_{n=1}^{\infty}\lambda^{n-1}\mathcal{T}(s_x,a_y|s_0)\mathcal{T}(s_i|s_x,a_y)$$
$$=\mathcal{T}(s_0)+\sum_{s_x\in\mathcal{S}}\sum_{a_y\in\mathcal{A}}\rho(s_x,a_y)^{\pi}_{s_0}\mathcal{T}(s_i|s_x,a_y) \tag{A4}$$

Here, $\mathcal{T}(s_0)=1$ is the probability of starting in the initial state s_0. Equation (A4) leads to the expression

$$\sum_{a_j\in\mathcal{A}}\rho(s_i,a_j)-\sum_{s_x\in\mathcal{S}}\sum_{a_y\in\mathcal{A}}\rho(s_x,a_y)^{\pi}_{s_0}\mathcal{T}(s_i|s_x,a_y)=\mathcal{T}(s_0) \tag{A5}$$

Equations (A2) and (A5) can be expressed in their respective matrix forms:

$$\mathcal{V}=\mathcal{R}^T\rho \tag{A6}$$

$$([I\quad I\quad\ldots\quad I]-[\mathcal{T}^T_{a_1}\mathcal{T}^T_{a_2}\ldots\mathcal{T}^T_{a_n}])\rho=\xi \tag{A7}$$

Here, $\mathcal{V}\in\mathbb{R}^{|\mathcal{S}|}$ and $\mathcal{R}$, $\rho\in\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$. $I\in\mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ is the identity matrix, and $\mathcal{T}_{a_i}\in\mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ is the transition probability matrix for each action $a_i\in\mathcal{A}$. $\xi\in\mathbb{R}^{|\mathcal{S}|}$ is the initial state distribution with $\xi(s_0)=1$, and all other states $\xi(s_i)$ are zeros. Using Eqs. (A6) and (A7), the problem of maximizing the cumulative reward [Eq. (1)] is formulated as a linear program (LP):

$$\max\ \mathcal{R}^T\rho \tag{A8}$$

subject to the constraints

$$([I\quad I\quad\ldots\quad I]-[\mathcal{T}^T_{a_1}\mathcal{T}^T_{a_2},\ldots\mathcal{T}^T_{a_n}])\rho=\xi$$
$$\rho\geq 0 \tag{A9}$$

Note that the solution to Eqs. (A8) and (A9) corresponds to the MDP without constraints [Eq. (1)]. The additional constraints imposed by Eq. (18) are expressed in terms of the occupational measures. For example, consider the constraint

$$\mathcal{T}(s_i|s_0)\leq\bar{p}_i$$

The preceding constraint can be expressed as

$$\sum_{a_j\in\mathcal{A}}\mathcal{T}(s_i,a_j|s_0)\leq\bar{p}_i$$
$$\sum_{n=0}^{\infty}\lambda^n\sum_{a_j\in\mathcal{A}}\mathcal{T}(s_n=s_i,a_n=a_j|s_0)\leq\sum_{n=0}^{\infty}\lambda^n\bar{p}_i \tag{A10}$$
$$\sum_{a_j\in\mathcal{A}}\rho(s_i,a_j),\leq\sum_{n=0}^{\infty}\lambda^n\bar{p}_i \tag{A11}$$
$$\sum_{a_j\in\mathcal{A}}\rho(s_i,a_j)\leq\frac{1}{1-\lambda}\bar{p}_i$$
$$\bar{z}^T\rho\leq\frac{1}{1-\lambda}\bar{p}_i \tag{A12}$$

Here, $\bar{z}$ is a vector of zeros, with ones in the positions corresponding to the occupational measures of state s_i. Equations (A8), (A9), and (A12) comprise the LP formulation for the constrained MDP or CMDP [44]. For each state s_i, a probability distribution over actions (policy) is obtained from the occupational measures:

$$\mathcal{T}(a_j|s_i)=\frac{\rho(s_i,a_j)^{\pi}_{s_0}}{\sum_{a_j}\rho(s_i,a_j)^{\pi}_{s_0}} \tag{A13}$$

Appendix B: Monte Carlo Simulation Framework

B1. Aircraft Dynamics

The aircraft is modeled by ordinary differential equations given by [45]

$$m(\dot{u}-vr+wq)=-(\sin\theta)mg-(\cos\beta)(\cos\alpha)\bar{\mathcal{D}}$$
$$+(\sin\alpha)\bar{\mathcal{L}}+(\cos\phi_T)F_T+F_{x_{\text{gear}}} \tag{B1}$$

$$m(\dot{v}+ur-wp)=(\sin\phi)(\cos\theta)mg-(\sin\beta)\bar{\mathcal{D}}+F_{y_{\text{gear}}} \tag{B2}$$

$$m(\dot{w}-uq+vp)=(\cos\phi)(\cos\theta)mg-(\cos\beta)(\sin\alpha)\bar{\mathcal{D}}$$
$$-(\cos\alpha)\bar{\mathcal{L}}-(\sin\phi_T)F_T+F_{z_{\text{gear}}} \tag{B3}$$

$$I_{xx}\dot{p}+(I_{zz}-I_{yy})qr-I_{xz}(\dot{r}+pq)=\bar{L}_{\text{aero}}+\bar{L}_{\text{thrust}}+\bar{L}_{\text{gear}} \tag{B4}$$

$$I_{yy}\dot{q}+(I_{xx}-I_{zz})pr+I_{xz}(p^2-r^2)=\bar{M}_{\text{aero}}+\bar{M}_{\text{thrust}}+\bar{M}_{\text{gear}} \tag{B5}$$

$$I_{zz}\dot{r}+(I_{yy}-I_{xx})pq+I_{xz}(qr-\dot{p})=\bar{N}_{\text{aero}}+\bar{N}_{\text{thrust}}+\bar{N}_{\text{gear}} \tag{B6}$$

$$\dot{x}=u\cos\psi\cos\theta+v(\cos\psi\sin\theta\sin\phi-\sin\psi\cos\phi)$$
$$+w(\cos\phi\sin\theta\cos\phi+\sin\psi\sin\phi) \tag{B7}$$

$$\dot{y}=u\sin\psi\cos\theta+v(\sin\psi\sin\theta\sin\phi+\cos\psi\cos\phi)$$
$$+w(\sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi) \tag{B8}$$

$$\dot{z}=u\sin\theta-v\cos\theta\sin\phi-w\cos\theta\cos\phi \tag{B9}$$

The translational motion is captured by Eqs. (B1–B3) and (B7–B9). Rotational dynamics are modeled by Eqs. (B4–B6). u, v, w represent aircraft body velocities. ϕ, θ, ψ represent roll, pitch, and yaw, respectively. p, q, r denote the angular rates. x, y, z represent aircraft position. $\bar{\mathcal{L}}$, $\bar{\mathcal{D}}$ represent lift and drag forces, respectively. $\bar{L}$, $\bar{M}$, $\bar{N}$ represent roll, pitch, and yawing moments, respectively, due to aerodynamic, thrust, and gear contact forces. Unlike conventional aircraft equations of motion, modeling takeoff dynamics requires knowledge of the gear forces and moments. Detailed description of the landing-gear dynamics can be found in [13].

B2. Pilot Control Inputs

The pilot's elevator u_e and rudder u_r inputs are given next, whereas the aileron input is assumed to be zero. Note that this work assumes that the pilot's control column and rudder inputs are translated directly to control surface deflections (direct law [1]):

$$u_e = \begin{cases} k_{p_e}(\theta_{\text{ref}_1} - \theta(t-\tau)) + k_{d_e} q & \text{if } (v \geq V_r) \\ k_{p_e}(\theta_{\text{ref}_2} - \theta(t-\tau)) + k_{d_e} q & \text{if } (v < V_r) \end{cases},$$
$$u_r = k_{p_r}(\psi_{\text{ref}} - \psi(t-\tau)) + k_{d_r} r,$$
$$\psi_{\text{ref}} = k_Y y \tag{B10}$$

Equation (B10) represents a simple human operator model [46,47] that treats the pilot as a proportional-derivative feedback law with time delay. Elevator input is modeled such that the pilot increases aircraft pitch attitude after rotation speed is reached. The rudder input is modeled such that the pilot tries to track the runway centerline. Here, k_p is a proportional feedback gain, k_d is a derivative gain, and τ is the time delay. $\theta(t-\tau)$ represents the inherent lag in pilot response due to time taken for perception of and reaction to external stimuli and neuromuscular interactions [46]. θ_{ref_1} is the appropriate pitch reference attitude during rotation. θ_{ref_2} is the reference pitch attitude before rotation (ideally zero). $\theta_{\text{ref}} - \theta(t-\tau)$ is the error in tracking the appropriate rotation attitude (θ_{ref}). V_r denotes the rotation airspeed perceived by the pilot, ideally V_R.

Equation (B10) represents a typical pilot behavior during takeoff. Although nominal values for θ_{ref} and V_r could be specified, actual parameter values such as k_{p_e}, k_{d_e}, and τ will be pilot-dependent. For example, it is rare for any two pilots to have the same response time; thus, τ varies between pilots [46]. The delay τ can also be influenced by other factors such as time of day and runway conditions. Parameter values are also different for each takeoff due to pilot input and environmental differences. For this work, θ_{ref}, V_r, k_p, k_d, and τ are uniformly sampled from bounded intervals $[\theta_{\text{ref}_{\min}}, \theta_{\text{ref}_{\max}}]$, $[V_{r_{\min}}, V_{r_{\max}}]$, $[k_{p_{\min}}, K_{p_{\max}}]$, $[K_{d_{\min}}, K_{d_{\max}}]$, and $[\tau_{\min}, \tau_{\max}]$, respectively.

The pilot's throttle control input is modeled as a function of engines' operational state. The engines can be all operational (E_{AEO}), one engine can be inoperative (E_{OEI}), all engines can be inoperative (E_{AEI}). For each takeoff sequence, the operational state of the engine $E \in \{E_{\text{AEO}}, E_{\text{OEI}}, E_{\text{AEI}}\}$ is sampled according to a specified distribution called as the engine failure distribution. If the sampled engine status denotes one or more engine failure(s) (i.e., $E_{\text{OEI}}/E_{\text{AEI}}$), then an engine failure is simulated by initializing the aircraft with all engines operational (E_{AEO}) and then triggering the engine failure event E_{OEI} or E_{AEI} at time $t_{\text{fail}} \in [0, t_f]$ by setting the thrust in the failed engines to zero. Note that $[0, t_f]$ denotes the takeoff time interval, and t_{fail} is sampled uniformly within this time interval. If the sampled engine status is nominal (E_{AEO}), it is assumed that the appropriate takeoff thrust $T_{\max}$ is used. Rejected takeoffs can also be caused by other factors besides engine failures. To account for these scenarios, a rejected takeoff scenario at time t_{fail} is triggered depending on the value of a Bernoulli random variable.

B3. Envelope-Aware Controller

When off-nominal conditions are encountered during takeoff, FSAM transfers control to the envelope-aware controller that attempts LOC prevention or recovery. Note that FSAM is only an overriding mechanism that selects the appropriate control authority in a LOC situation. In this work, the envelope-aware controller is designed to ensure that the aircraft states remain within safe operating envelopes. The design of control laws to prevent constraint violation is beyond the scope of this work (see [48,49] and references therein for related work). In this work, elevator input for the envelope-aware feedback control law is modeled as

$$u_e = \begin{cases} \bar{K}_1(\bar{\theta}_{\text{ref}_1} - \theta(t)) + \bar{K}_2 q & \text{if } (\theta(t) < \theta_{\text{PR}} \,\&\, v < V_R) \\ \bar{K}_3(\bar{\theta}_{\text{ref}_2} - \theta(t)) + \bar{K}_4 q & \text{if } (\theta(t) \geq \theta_{\text{PR}} \,\&\, v < V_R) \\ \bar{K}_5(\bar{\theta}_{\text{ref}_3} - \theta(t)) + \bar{K}_6 q & \text{if } (\theta(t) < \theta_{\text{TS}} \,\&\, v \geq V_R) \\ \bar{K}_7(\bar{\theta}_{\text{ref}_4} - \theta(t)) + \bar{K}_8 q & \text{if } (\theta(t) \geq \theta_{\text{TS}} \,\&\, z(t) < h_{\text{TS}} \,\&\, v \geq V_R) \end{cases} \tag{B11}$$

Here, $\bar{K}_i$, $i = 1, \ldots, 8$ and θ_{ref_j}, $j = 1, \ldots, 4$ are chosen such that the closed-loop response of the aircraft is free from high-risk states such as premature rotation and tail strike. $\theta_{\bar{T}S}$, $h_{\bar{T}S}$ represents the threshold when tail-strike protection should be activated, whereas θ_{PR} represents the threshold when prevention against premature rotation is activated.¶

The rudder input of the envelope-aware (EA) controller is modeled as

$$\psi_{\text{ref}} = \begin{cases} \psi_{\text{runway}} + 10 \text{ deg} & \text{if } \left(y < -\frac{y_{\text{width}}}{4}\right) \\ \psi_{\text{runway}} - 10 \text{ deg} & \text{if } \left(y > \frac{y_{\text{width}}}{4}\right) \end{cases} \tag{B12}$$

$$u_r = K_{p_r}(\psi_{\text{ref}} - \psi) + K_{d_r} r \tag{B13}$$

When all engines are operational ($E = E_{\text{AEO}}$), the EA controller's thrust input is modeled as

$$T = \begin{cases} T_{\max} & \text{if } \bar{Q} \in \{q_1, q_2, q_3, q_4, q_5, q_6, q_7\} \\ T_{\text{idle}} & \text{otherwise} \end{cases} \tag{B14}$$

When $E = E_{\text{OEI}}$, thrust is

$$T = \begin{cases} \frac{T_{\max}}{2} & \text{if } \bar{Q} \in \{q_3, q_4, q_5, q_6, q_7\} \\ T_{\text{idle}} & \text{otherwise} \end{cases} \tag{B15}$$

A total of 100,000 trials were run. The initial conditions for x_0, u_0 were sampled uniformly over intervals $[0, R_{\max}]$ and $[0, V_2]$, respectively. Here, $R_{\max}$ denotes the available runway length, and V_2 denotes the takeoff safety V speed. Initial conditions for the remaining state variables were set to zero.

Acknowledgment

This work was supported in part by NASA under cooperative agreement NNX12AM54A.

References

[1] Gregg, F. B., "Boeing B-777: Fly-By-Wire Flight Controls," *The Avionics Handbook*, CRC Press, Boca Raton, FL, 2015, Chap. 29.

[2] Brière, D., and Traverse, P., "AIRBUS A320/A330/A340 Electrical Flight Controls—A Family of Fault-Tolerant Systems," *Proceedings of the 23rd International Symposium on Fault-Tolerant Computing, FTCS-23. Digest of Papers*, IEEE Publ., Piscataway, NJ, 1993, pp. 616–623.
doi:10.1109/FTCS.1993.627364

[3] "Statistical Summary of Commercial Jet Airplane Accidents," Boeing Commercial Airplanes, Seattle, WA, 2015, http://www.boeing.com/news/techissues/pdf/statsum.pdf [retrieved Sept. 2015].

[4] BelSrcastro, C. M., and Foster, J. V., "Aircraft Loss-of-Control Accident Analysis," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2010-8004, Aug. 2010.
doi:10.2514/6.2010-8004

[5] Belcastro, C. M., Newman, R. L., Crider, D. A., Groff, L., Foster, J. V., Klyde, D. H., and Huston, A. M., "Preliminary Analysis of Aircraft Loss of Control Accidents: Worst Case Precursor Combinations and Temporal Sequencing," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2014-0612, 2014.
doi:10.2514/6.2014-0612

[6] Belcastro, C. M., and Jacobson, S. R., "Future Integrated System Concepts for Preventing Aircraft Loss-of-Control Accidents," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2010-8142, Aug. 2010.
doi:10.2514/6.2010-8142

[7] Gingras, D. R., Barnhart, B., Ranaudo, R., Ratvasky, T. P., and Morelli, E., "Envelope Protection for In-Flight Ice Contamination," *47th AIAA Aerospace Sciences Meeting*, AIAA Paper 2009-1458, Jan. 2009.
doi:10.2514/6.2009-1458

[8] Borst, C., Grootendorst, F. H., Brouwer, D. I. K., Bedoya, C., Mulder, M., and van Paassen, M. M., "Design and Evaluation of a Safety

¶These thresholds were selected by trial and error. However, they can be computed formally by techniques described in [49].

Augmentation System for Aircraft," *Journal of Aircraft*, Vol. 51, No. 1, 2013, pp. 12–22.
doi:10.2514/1.C031500

[9] Srivatsan, R., Downing, R. D., and Bryant, H. W., "Development of Takeoff Performance Monitoring System," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 5, 1987, pp. 433–440.
doi:10.2514/3.20237

[10] Milligan, M. W., Zhou, M. M., and Wilkerson, H. J., "Monitoring Airplane Takeoff Performance: Prototype Instrument with Learning Capability," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 4, 1995, pp. 768–772.
doi:10.2514/3.46789

[11] Zammit-Mangion, D., and Eshelby, M., "Simplified Algorithm to Model Aircraft Acceleration During Takeoff," *Journal of Aircraft*, Vol. 45, No. 4, 2008, pp. 1090–1097.
doi:10.2514/1.22966

[12] Balachandran, S., and Atkins, E. M., "Flight Safety Assessment and Management During Takeoff," *AIAA Infotech@Aerospace Conference*, AIAA Paper 2013-4805, 2013.
doi:10.2514/6.2013-4805

[13] Balachandran, S., and Atkins, E. M., "An Evaluation of Flight Safety Assessment and Management to Avoid Loss of Control During Takeoff," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2014-0785, 2014.
doi:10.2514/6.2014-0785

[14] Balachandran, S., and Atkins, E. M., "Flight Safety Assessment and Management for Takeoff Using Deterministic Moore Machines," 2015, http://dx.doi.org/10.2514/1.I010350 [retrieved Sept. 2015].

[15] Kochenderfer, M. J., and Chryssanthacopoulos, J. P., "A Decision-Theoretic Approach to Developing Robust Collision Avoidance Logic," *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, IEEE Publ., Piscataway, NJ, 2010, pp. 1837–1842.
doi:10.1109/ITSC.2010.5625063.

[16] Kochenderfer, M. J., Chryssanthacopoulos, J. P., Kaelbling, L. P., and Lozano-Perez, T., "Model-Based Optimization of Airborne Collision Avoidance Logic," Lincoln Lab. Project Rept. ATC-360, Massachusetts Inst. of Technology, Cambridge, MA, 2010, http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA513420 [retrieved Sept. 2015].

[17] Temizer, S., "Planning Under Uncertainty for Dynamic Collision Avoidance," Ph.D. Dissertation, Massachusetts Inst. of Technology, Cambridge, MA, 2011, http://dspace.mit.edu/handle/1721.1/64487 [retrieved Sept. 2015].

[18] Winder, L. F., "Hazard Avoidance Alerting with Markov Decision Processes," Ph.D. Dissertation, Massachusetts Inst. of Technology, Cambridge, MA, 2004, http://hdl.handle.net/1721.1/28860 [retrieved Feb. 2013].

[19] Tomlin, C., Pappas, G. J., and Sastry, S., "Conflict Resolution for Air Traffic Management: A Study in Multiagent Hybrid Systems," *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, 1998, pp. 509–521.
doi:10.1109/9.664154

[20] Sun, F., Ozay, N., Wolff, E., Liu, J., and Murray, R., "Efficient Control Synthesis for Augmented Finite Transition Systems with an Application to Switching Protocols," *Proceedings of the American Control Conference*, IEEE Publ., Piscataway, NJ, 2014.
doi:10.1109/ACC.2014.6859428

[21] Bonet, B., and Geffner, H., "Planning as Heuristic Search," *Artificial Intelligence*, Vol. 129, No. 1, 2001, pp. 5–33.
doi:10.1016/S0004-3702(01)00108-4

[22] Peter, B. R., James, F. R., Gat, E., Kortenkamp, D., Miller, D. P., and Slack, M. G., "Experiences with an Architecture for Intelligent, Reactive Agents," *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 9, Nos. 2–3, 1997, pp. 237–256.
doi:10.1080/095281397147103

[23] Laird, J. E., Newell, A., and Rosenbloom, P. S., "Soar: An Architecture for General Intelligence," *Artificial Intelligence*, Vol. 33, No. 1, 1987, pp. 1–64.
doi:10.1016/0004-3702(87)90050-6

[24] Gregory, I. M., Cao, C., Xargay, E., Hovakimyan, N., and Zou, X., "L1 Adaptive Control Design for NASA AirSTAR Flight Test Vehicle," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2009-5738, Aug. 2009.
doi:10.2514/6.2009-5738

[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*, AIAA Paper 2014-0786, 2014.
doi:10.2514/6.2014-0786

[26] Yu, M.-J., McDonough, K., Bernstein, D. S., and Kolmanovsky, I., "Retrospective Cost Model Refinement for Aircraft Fault Signature Detection," *Proceedings of the American Control Conference (ACC)*, IEEE Publ., Piscataway, NJ, 2014, pp. 2486–2491.
doi:10.1109/ACC.2014.6858876

[27] Meuleau, N., Plaunt, C., Smith, D. E., and Smith, T. B., "An Emergency Landing Planner for Damaged Aircraft," *Innovative Applications of Artificial Intelligence*, IAAI Paper 09268-2009, Pasadena, CA, 2009, http://aaai.org/ocs/index.php/IAAI/IAAI09/paper/view/268.

[28] Atkins, E. M., "Emergency Landing Automation Aids: An Evaluation Inspired by US Airways Flight 1549," *AIAA Infotech@Aerospace Conference*, AIAA Paper 2010-3381, April 2010.
doi:10.2514/6.2010-3381

[29] Di Donato, P. F. A., and Atkins, E. M., "An Off-Runway Emergency Landing Aid for a Small Aircraft Experiencing Loss of Thrust," *AIAA Infotech@Aerospace Conference*, AIAA Paper 2015-1798, 2015.
doi:10.2514/6.2015-1798

[30] Russell, S. J., and Norvig, P., *Artificial Intelligence: A Modern Approach*, Pearson Education, Upper Saddle River, NJ, 2014, Chap. 17.

[31] Puterman, M. L., *Markov Decision Process: Discrete Stochastic Dynamic Programming*, Wiley, Hoboken, NJ, 1994, Chaps. 2–6.

[32] Balachandran, S., and Atkins, E. M., "An Autonomous Override System to Prevent Airborne Loss of Control," *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence and the Twenty-Eighth Conference on Innovation Applications of Artificial Intelligence*, The AAAI Press, Palo Alto, CA, 2016, pp. 3991–3396.

[33] "Reducing the Risk of Runway Excursions, Report of the Runway Safety Initiative," Flight Safety Foundation, Alexandria, VA, May 2009, http://www.skybrary.aero/bookshelf/books/900.pdf [retrieved Sept. 2012].

[34] Lepird, J. R., Owen, M. P., and Kochenderfer, M. J., "Bayesian Preference Elicitation for Multiobjective Engineering Design Optimization," *Journal of Aerospace Information Systems*, Vol. 12, No. 10, 2015, pp. 634–645.
doi:10.2514/1.I010363

[35] "Tailstrike and Runway Overrun, Melbourne Airport, Victoria," Australian Transportation Safety Bureau Accident Rept. AO-2009-012, 2009, Australian Capital Territory, Australia, http://www.atsb.gov.au/publications/investigation_reports/2009/aair/ao-2009-012.aspx [retrieved 01 Sept. 2014].

[36] Holzmann, G. J., "The Model Checker SPIN," *IEEE Transactions on Software Engineering*, Vol. 23, No. 5, May 1997, pp. 279–295.
doi:10.1109/32.588521

[37] Cimatti, A., Clarke, E., Giunchiglia, F., and Roveri, M., "NuSMV: A New Symbolic Model Verifier," *Computer Aided Verification*, Springer, Berlin, 1999, pp. 495–499.

[38] Grosu, R., and Smolka, S. A., "Monte Carlo Model Checking," *Tools and Algorithms for the Construction and Analysis of Systems*, Springer, Berlin, 2005, pp. 271–286.
doi:10.1007/978-3-540-31980-1_18

[39] Sankaranarayanan, S., and Fainekos, G., "Falsification of Temporal Properties of Hybrid Systems Using the Cross-Entropy Method," *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control*, ACM, New York, 2012, pp. 125–134.
doi:10.1145/2185632.2185653

[40] Kwiatkowska, M., Norman, G., and Parker, D., "PRISM 4.0: Verification of Probabilistic Real-Time Systems," *Computer Aided Verification*, Springer, Berlin, 2011, pp. 585–591.
doi:10.1007/978-3-642-22110-1_47

[41] Bertsekas, D. P., and Tsitsiklis, J. N., "Neuro-Dynamic Programming: An Overview," *Proceedings of the 34th IEEE Conference on Decision and Control*, Vol. 1, IEEE Publ., Piscataway, NJ, 1995, pp. 560–564.
doi:10.1109/CDC.1995.478953

[42] Sutton, R. S., and Barto, A. G., *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998, Chaps. 6–10.

[43] Donato, P., Balachandran, S., McDonough, K., Atkins, E. M., and Kolmanovsky, I., "Envelope-Aware Flight Control Recovery and Emergency Landing with a Rudder Jam Case Study," *Journal of Guidance, Navigation, and Control* (submitted for publication).

[44] Altman, E., *Constrained Markov Decision Processes*, Vol. 7, CRC Press, Boca Raton, FL, 1999, pp. 19–55.

[45] Stevens, L. B., and Lewis, L. F., *Aircraft Control and Simulation*, Wiley, Hoboken, NJ, 2003, pp. 1–137.

[46] McRuer, D. T., and Krendel, E. S., "Mathematical Models of Human Pilot Behavior," Advisory Group for Aerospace Research and Development, NATO, Distributed by National Technical Information Service (NTIS), Sprinfield, VA, 1974, https://www.cso.nato.int/Pubs/rdp.asp?RDP=AGARD-AG-188 [retrieved Sept. 2015].

[47] Hess, R. A., "Unified Theory of Aircraft Handling Qualities and Adverse Aircraft-Pilot Coupling," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 6, 1997, pp. 1141–1148. doi:10.2514/2.4169

[48] McDonough, K., and Kolmanovsky, I., "Integrator Resetting for Enforcing Constraints in Aircraft Flight Control Systems," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2015-1995, 2015. doi:10.2514/6.2015-1995

[49] Mitchell, I., and Tomlin, C. J., "Level Set Methods for Computation in Hybrid Systems," *Hybrid Systems: Computation and Control*, Springer, Berlin, 2000, pp. 310–323. doi:10.1007/3-540-46430-1_27