JOURNAL OF AEROSPACE INFORMATION SYSTEMS
Vol. 13, No. 9, September 2016

Verification Guided Refinement of Flight Safety Assessment and Management System for Takeoff

Sweewarman Balachandran,* Necmiye Ozay,† and Ella M. Atkins‡
University of Michigan, Ann Arbor, Michigan 48109

DOI: 10.2514/1.I010408

Systems that make safety-critical decisions must undergo a rigorous verification and validation process to ensure automation decisions do not jeopardize the nominal safe state of operation. Flight safety assessment and management is a high-level decision-making system to reduce loss of control risk. This paper demonstrates how tools from formal verification can be used to guide the design of a takeoff flight safety assessment and management system implemented as a deterministic Moore machine. Finite state abstractions of simplified takeoff dynamics under different control authorities (i.e., pilot vs safety controller) are computed and composed with the Moore machine. By construction, the composition captures all behaviors of simplified takeoff dynamics. Then, a model checking tool analyzes whether this composition satisfies the takeoff safety requirements specified by federal aviation regulations. The results of model checking together with the abstraction are used to refine the Moore machine to ensure satisfaction of the specification. This paper contributes a novel approach to verification of a supervisory system specified by a Moore machine and applies this technique to flight safety assessment and management, which is itself an emerging flight management automation aid.

Nomenclature

$\mathcal{A}, \mathcal{G}$	=	deterministic Moore machine longitudinal takeoff, output function
C_{L_g}, C_{D_g}	=	coefficient of lift and drag with ground effects
$\mathcal{H}, \bar{T}$	=	observation map, abstraction function
s, σ	=	deterministic Moore machine state, alphabet symbol
P, EA	=	pilot, envelope-aware controller
q, θ, γ	=	pitch angular rate, pitch attitude, flight-path angle
$\bar{q}$, q	=	discrete state denoting a cell, discrete state denoting a facet of a cell
T, W, ρ	=	thrust, weight, atmospheric density
V_1, V_R, V_{lof}	=	go/no-go decision speed, rotation speed, liftoff speed
x, v, h	=	longitudinal position, airspeed, altitude
μ, g, S_{ref}	=	rolling friction coefficient, acceleration due to gravity, planform area

I. Introduction

LOSS of control (LOC) is the most common contributing factor to aviation accidents [1]. LOC results when an aircraft exits its safe flight envelope or collides with another aircraft, building, or surrounding terrain and has been widely addressed in previous research [2–8]. In our previous publications [8–10], a flight safety assessment and management (FSAM) capability was proposed to prevent LOC during the takeoff phase of flight. FSAM is an automation aid responsible for real-time assessment of LOC risk, activation of risk-based warnings, and resilient control override of the flight crew if necessary. Safety-critical flight systems such as FSAM must undergo a rigorous validation and verification process to ensure they meet the necessary safety certification criteria.

Verification and validation (V&V) are essential steps in the traditional V model [11] for system engineering, as illustrated in Fig. 1 [12]. The system is designed, built, and then tested comprehensively to ensure that all specified system requirements are satisfied. Validation asks the question "are we building the right system?" and verification asks the question "are we building the system correctly per the specifications?" This conventional approach (Fig. 1) can be labor-intensive and costly but has been shown to be an effective means to organize system development.

Formal methods such as model checking [13] and deductive techniques [14] efficiently augment the traditional simulation and testing-based V&V [11]. Formal methods help establish the correctness of a system design with respect to specified requirements before building and testing the system. Model checking identifies violations of the specified requirement set by exhaustively searching the state space of an abstract representation (model) of the system. Deductive techniques such as theorem-proving use mathematical arguments to prove or disprove the correctness of the design with respect to system requirements.

Formal verification tools are gaining traction in the aerospace industry. For example, Airbus used a model checking approach to validate the ground spoiler functionality on the A380 aircraft [15]. Rockwell Collins used a theorem-proving approach to verify the functionality of a flight guidance system [16,17]. Joshi et al. [18] proposed a model-based safety analysis that extends model checking with fault trees used to analyze safety-critical components. The idea of a sandbox controller was introduced by Bak et al. [19], in which a nominal system is augmented with a safety controller and a decision module to prevent the system from entering an unsafe state. Lygeros and Lynch [20] used an automaton-based method to verify their traffic collision avoidance system conflict resolution algorithm.

The preceding references focus only on the verification of the system and do not consider the influence of the operator. A human factors approach to model checking was adopted by Degani and Heymann [21]. In [21], interactions between a human operator and a machine are

Received 18 June 2015; revision received 16 April 2016; accepted for publication 19 May 2016; published online 5 August 2016.

*Graduate Student, Department of Aerospace Engineering. Student Member AIAA.
†Assistant Professor, Department of Electrical Engineering and Computer Science.
‡Professor, Department of Aerospace Engineering. Associate Fellow AIAA.

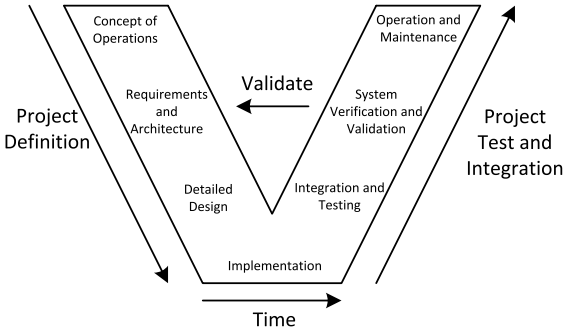

Fig. 1 V model for system engineering [12].

formally analyzed to guide the design of the interfaces between human and machine and to develop better training manuals. Bolton et al. [22] presented an approach to verify human automation interaction using task analytical models. In [22], the task analytic model capturing the human operator's behavior is combined with a model of the system under consideration and is verified using a model checking tool.

Finite state machine formulations and tools from automata theory are used for the model checking of discrete systems (see [13,23] and references therein). Continuous time systems can be transformed into discrete systems using various abstraction techniques (see [24] for details) to exploit model checking methods used on discrete systems. The use of backward reachable sets [7] and forward reachable sets [25–27] have been widely used to verify safety properties for low-dimensional hybrid systems. The use of barrier certificates for the verification of hybrid systems was explored in [28]. A theorem-proving technique can also be used for the formal verification of hybrid systems [20]. The use of probabilistic approaches for model checking was explored in [29,30] to facilitate the verification of higher-order systems. This work employs an abstraction-based technique to facilitate the verification of FSAM.

The main contributions of this paper are 1) a general approach to verify a switched control system for which the switching policy is realized by a deterministic finite state Moore machine [31] and 2) a model checking framework to guide the refinement of and verify the FSAM system against safety requirements specified in the federal aviation regulations (FAR). Specifically, a suitable representation of the underlying state space for takeoff FSAM is established first. Next, a discrete transition system that encodes an overapproximation of the reachable states under the available control authorities is constructed. Finally, a composition of the discrete transition system and the finite state machine specifying the switching protocol is constructed. The composed transition system is then used to verify requirements are always satisfied with SPIN [32], an existing model checking tool.

In this work, simplified models that can adequately capture the takeoff dynamics for verification are used. Safety requirements for takeoff extracted from FAR part 25 are expressed in linear temporal logic (LTL) [33], which facilitate model checking. The results of verification are also compared with a Monte Carlo analysis that makes use of a higher-order nonlinear model describing takeoff dynamics. Counterexamples obtained from the model checking process identify necessary refinements of the underlying FSAM switching protocol.

Section II provides background on the tools necessary to perform model checking. Section III presents the FSAM formulation for takeoff, develops a simplified dynamics model for takeoff that facilitates verification, defines safety requirements to satisfy during takeoff, and outlines the proposed approach to model check the FSAM switching policy. Section IV describes the proposed approach for model checking and the results of verification. Section V discusses refinements to FSAM based on the results of verification, and Sec. VI considers validation of FSAM. Section VII provides a discussion on the proposed approach, and Sec. VIII presents conclusions and future extensions.

II. Background

Model checking is the process of ensuring that a system satisfies a set of requirements. This section introduces the modeling and specification formalisms used in this paper to enable FSAM model checking. FSAM is a switching control system and its switching policy is modeled as a deterministic Moore machine (DMM) [8]. DMMs are finite state machines in which each state has a prescribed output. The use of a deterministic specification for FSAM facilitates its verification using well-established tools in model checking. The underlying dynamics of the aircraft for takeoff is abstractly represented as a discrete transition system. The safety requirements for takeoff are expressed in linear temporal logic. Formal definitions are provided next.

A. Deterministic Moore Machine

A DMM [31] is defined by the tuple $(\mathcal{S}, \mathcal{S}_0, \Sigma, \Lambda, \mathcal{T}, \mathcal{G})$. Here, $\mathcal{S}$ represents a discrete set of states, $\mathcal{S}_0 \subset \mathcal{S}$ represents an initial state, Σ is a finite input alphabet, Λ is a finite output alphabet, and $\mathcal{T} \subseteq \mathcal{S} \times \Sigma \times \mathcal{S}$ represents a transition relation. $\mathcal{G}$: $\mathcal{S} \to \Lambda$ is the output function that maps each state to the output alphabet.

B. Finite Transition Systems

A finite transition system [13] is a tuple $(\mathcal{Q}, \mathcal{Q}_0, \mathcal{P}, \Gamma, \Pi, \mathcal{L})$ in which $\mathcal{Q}$ is a set of discrete states, and $\mathcal{Q}_0$ is the set of initial states. $\mathcal{P}$ represents a finite input/action set. $\Gamma \subseteq \mathcal{Q} \times \mathcal{P} \times \mathcal{Q}$ is a transition relation. Π is a set of atomic propositions and $\mathcal{L}$: $\mathcal{Q} \to 2^{\Pi}$ is a labeling function.

C. Linear Temporal Logic

LTL is a formal specification language [13,34] that can be used to describe a rich class of system properties. LTL is built upon a finite set of atomic propositions Π plus logical operators $\neg$ (negation), $\vee$ (disjunction), and temporal/modal operators $\bigcirc$ (next), and $\mathcal{U}$ (until). Properties such as safety, reachability, invariance, and combinations of these can be expressed using LTL. The set of LTL formulas over a finite set of atomic propositions Π can be inductively defined as follows: 1) Any atomic proposition $\pi \in \Pi$ is an LTL formula. 2) If $\bar{\varphi}$ and $\bar{\psi}$ are LTL formulas, then $\neg\bar{\varphi}$, $\bigcirc\bar{\varphi}$, $\bar{\varphi} \vee \bar{\psi}$, and $\bar{\varphi}\mathcal{U}\bar{\psi}$ are also LTL formulas. Additional operators such as $\wedge$ (conjunction), $\Rightarrow$(implication), $\diamond$ (eventually), and $\Box$ (always) can also be defined (see [34,35] and references therein for detailed discussions on LTL syntax and semantics). This work focuses on verification of properties expressed using the $\Box$ (always) operator. A sequence of truth assignments to the atomic propositions $\pi \in \Pi$ satisfy $\Box\varphi$ if φ is true in every position of the sequence. In this work, LTL formulas are interpreted over time-sampled trajectories of dynamic systems (i.e., discrete-time semantics of LTL) [36].

III. Problem Formulation

This section formulates the model checking problem for FSAM. First, the takeoff FSAM DMM is introduced. Next, an approximate dynamic model for the takeoff phase is specified. Then, the safety requirements for the takeoff phase extracted from FAR part 25 are discussed. Using these three components, the model checking problem for FSAM is formally defined and the solution strategy used in this paper is outlined.

A. Takeoff Flight Safety Assessment and Management

Takeoff is one of the most hazardous phases of flight, second only to final approach and landing. Current takeoff regulations require that the flight crew follow standard operating procedures to configure the aircraft appropriately, obtain clearances, and manually fly the aircraft through initial departure climb [37].

FSAM is a high-level flight management system decision aid responsible for real-time assessment of LOC risk, activation of risk-based warnings, and resilient control override of the flight crew if necessary. In previous work, FSAM was applied to the prevention of LOC during takeoff [8–10] but the FSAM capability is still in early stages of development. Figure 2 illustrates a manually constructed DMM $\mathcal{A}$: $(\mathcal{S}, \mathcal{S}_0, \Sigma, \Lambda, \mathcal{T}, \mathcal{G})$ that represents FSAM logic for the longitudinal dynamics of takeoff. A formal representation of the DMM is provided in the Appendix of this paper. Each state $s \in \mathcal{S}$ or node in Fig. 2 represents a segment of takeoff. The input alphabet Σ consists of symbols that depend on the physical state of the aircraft (see Table A1). Each edge in Fig. 2 represents a transition (s, σ, s') over input alphabet symbol $\sigma \in \Sigma$ labeling the edge between states $s, s' \in \mathcal{S}$. $\Lambda = \{P, \text{EA}\}$ is the output alphabet, in which P denotes that the pilot is in control and EA denotes that an envelope-aware safety controller is active. The output of each state $\mathcal{G}(s) \in \Lambda$, indicated in the lower half of each node, determines the current control authority.

The DMM in Fig. 2 was constructed after analysis of aviation accident and incident reports, aircraft manuals, and safety briefings [8]. The goal was to develop a switching strategy that could recognize high-risk states during the takeoff ground roll and activate an envelope-aware controller to mitigate risk. The top row in Fig. 2 represents the nominal progression of states during the takeoff ground run with the pilot in control. As shown in Fig. 2, the aircraft starts from an initial state of rest s_1 at $x = 0$, $v = 0$. If the aircraft is configured for takeoff c and takeoff thrust $T_{\max}$ is established, the aircraft accelerates down the runway and the DMM state transitions through the nominal V-speed state progression. The top row of states in Fig. 2 represents the nominal V-speed sequence. The additional states represent off-nominal conditions with LOC risk. If the aircraft is inappropriately configured, the DMM enters Takeoff Configuration Warning (TOCW) state s_8, inducing a corresponding alert to the crew. If the configuration problem persists, the DMM transitions into the abort state s_{13}, where it overrides and rejects the takeoff. During the initial ground roll ($V_{\text{mcg}} < V \leq V_1$), if the aircraft has inadequate acceleration, FSAM rejects the takeoff f to prevent entry into unsafe regions of the state space with respect to rejected takeoff and one engine inoperative [8]. At higher speeds, the DMM monitors crew inputs to avoid premature rotation and tail strike (s_4 and s_5). After liftoff, conventional envelope protection features such as angle of attack (stall) and overspeed become active. Pushing the aircraft to the stall boundary during the climb (s_6, s_7) results in override, where the FSAM-activated envelope-aware controller (s_{11}, s_{12}) effectively offers the stall or envelope protection capabilities found on the existing aircraft. FSAM reverts control to the flight crew after the aircraft is stabilized on climbout. This paper focuses on verifying, validating, and refining the manually constructed longitudinal DMM developed in our previous work to ensure the satisfaction of takeoff safety requirements.

B. Longitudinal Dynamics for Takeoff

To illustrate verification of the longitudinal FSAM DMM, the following simplifying assumptions are made:

1) The lateral dynamics is well behaved (i.e., there are no lateral disturbances, and so the aircraft can maintain runway heading while staying on runway centerline throughout takeoff).

2) The engines, control surfaces, instruments, and all subsystems function nominally.

3) There is no runway incursion risk.

4) The only pilot behaviors impacting FSAM decisions are related to captured configuration settings and control inputs.

5) Envelope-aware control and guidance algorithms are capable of maintaining or recovering a safe state in any DMM in which the envelope-aware controller is active.

Verification of a system like FSAM requires the consideration of human pilot behavior. Several authors have developed models to describe human pilot behavior under different scenarios [21,22,38]. The preceding assumptions enable the usage of simple pilot behavior models (human operator transfer functions [39]) in verifying the FSAM DMM.

The full nonlinear equations describing the dynamics of the aircraft during takeoff were discussed in our previous publications [8,9]. The preceding assumptions allow this work to ignore lateral or directional dynamics. Furthermore, takeoff is decomposed into two segments: ground roll and climb. In the ground roll segment, the aircraft accelerates down the runway while the pitch attitude stays almost constant until achieving rotation airspeed V_R. At or above V_R, the pilot applies control inputs to rotate the nose of the aircraft. When liftoff speed V_{lof} is reached, the aircraft climbs (note $V_{\text{lof}} > V_R$). Thus, the longitudinal dynamics for takeoff can be split into the segments $V < V_{\text{lof}}$ and $V \geq V_{\text{lof}}$:

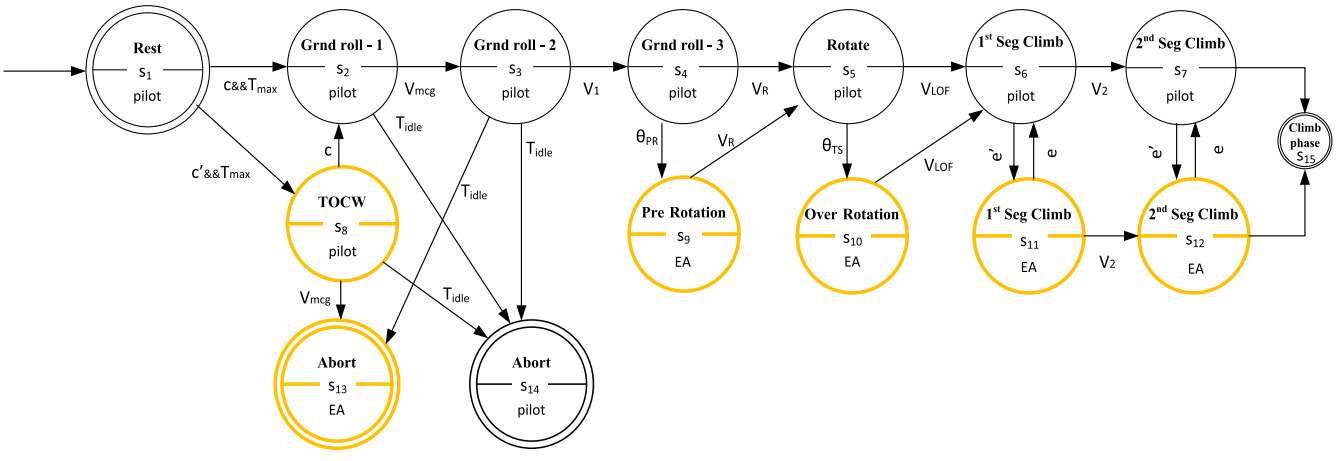

Fig. 2 Longitudinal FSAM Moore machine for takeoff.

Table 1 Requirements and their LTL specifications

No.	Requirement	LTL specification
1	During acceleration to speed V_2, the nose gear may be raised off the ground at a speed not less than V_R [FAR 25.111.(b)].	$\square((\theta \geq \theta_{ng}) \rightarrow (V \geq V_R))$
2	The pitch attitude of the airplane must not exceed an attitude that leads to the minimum tail clearance during rotation [FAR 25.107.(e).4].	$\square((h \leq h_{lof}) \rightarrow (\theta < \theta_{tail}))$
3	The slope of the airborne part of the takeoff path must be positive at each point [FAR 25.111.(c).1].	$\square((h > h_{lof}) \rightarrow (\theta > \theta_0))$
4	The airplane must reach V_2 before it is 35 feet above the takeoff surface [FAR 25.111.(c).2].	$\square((h \geq h_{obs}) \rightarrow (V \geq V_2))$

$$\dot{x} = \begin{cases} v & v < V_{\text{lof}} \\ v\cos(\gamma_0) & v \geq V_{\text{lof}} \end{cases}, \quad \dot{v} = \begin{cases} A_1 - B_1 v^2 & v < V_{\text{lof}} \\ A_2 - B_2 v^2 & v \geq V_{\text{lof}} \end{cases}, \quad \dot{h} = \begin{cases} 0 & v < V_{\text{lof}} \\ v\sin(\gamma_0) & v \geq V_{\text{lof}} \end{cases}, \quad \dot{q} = \begin{cases} A_3 q + B_3 u_e & v < V_R \\ A_4 q + B_4 u_e & v \geq V_R \end{cases}, \quad \dot{\theta} = q \tag{1}$$

in which x is the longitudinal position of the aircraft, v is airspeed, θ is pitch, q is angular rate, h is altitude, and γ is flight-path angle. Terms A_1, B_1, A_2, and B_2 are defined next using a formulation from [40]:

$$A_1 = g\left(\frac{T}{W} - \mu\right), \quad B_1 = \frac{g}{W}\left[\frac{1}{2}\rho S_{\text{ref}}(C_{D_g} - \mu C_{\text{Lg}})\right], \quad A_2 = g\left(\frac{T}{W}\cos(\alpha_0) - \sin(\gamma_0)\right), \quad B_2 = \frac{g}{W}\left(\frac{1}{2}\rho S_{\text{ref}} C_{D_g}\right)$$

Here, T represents the takeoff thrust, W is aircraft takeoff weight, ρ is atmospheric density, μ is the wheel rolling friction coefficient, γ_0 is the flight-path angle after lift off, S_{ref} is the planform area, and C_{L_g} and C_{D_g} are the coefficients of lift and drag, respectively, including aerodynamic ground effect and the impact of nominal takeoff flaps/slat settings. Pitch dynamics are approximated as a piecewise linear system defined by the pair (A_3, B_3) when $v < V_R$ and (A_4, B_4) when $v \geq V_R$.

For convenience, Eq. (1) is represented compactly as $\dot{X} = f(X, U)$. Here, $X \in \mathcal{X} \subseteq \mathbb{R}^5$ represents the state vector $[x, v, h, \theta, q]^T$, in which $\mathcal{X}$ is a compact hyperrectangle; $U \in \Omega \subseteq \mathbb{R}$ represents the elevator control input u_e; and $\{P, \text{EA}\}$ are the available control authorities. The elevator input provided by the pilot $(u_e(t)|_p)$ is given as

$$u_e(t)|_P = \begin{cases} K_p(\theta_{\text{ref}_1} - \theta(t-\tau)) + K_d q & \text{if } (v \geq V_r) \\ K_p(\theta_{\text{ref}_2} - \theta(t-\tau)) + K_d q & \text{if } (v < V_r) \end{cases} \tag{2}$$

Equation (2) represents a simple human operator model [39,41] that treats the pilot as a proportional-derivative feedback law with a time delay. 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 [39,41]; θ_{ref_1} is the appropriate pitch reference attitude during rotation; θ_{ref_2} is the reference pitch attitude before rotation (ideally zero); and $\theta_{\text{ref}} - \theta(t-\tau)$ is the error in tracking the appropriate rotation attitude θ_{ref}. V_r denotes the rotation airspeed perceived by the pilot and ideally would be equal to V_R. Equation (2) represents a typical pilot behavior during takeoff. Although specification of expected values would be possible for certain parameters, such as θ_{ref} and V_r, specific parameter values such as K_p, K_d, and τ would be pilot dependent. For example, it is rare for any two pilots to have the same response time, thus τ varies between pilots. The delay τ can also be influenced by several other factors, such as time of day, runway conditions, etc. The specific parameters are also different for each takeoff due to pilot input and environmental differences. In this work, it is assumed that the values of θ_{ref}, V_r, K_p, K_d, and τ lie within a bounded interval $[\theta_{\text{ref}_{\min}}, \theta_{\text{ref}_{\max}}]$, $[V_{r_{\min}}, V_{r_{\max}}]$, $[K_{p_{\min}}, K_{p_{\max}}]$, $[K_{d_{\min}}, K_{d_{\max}}]$, and $[\tau_{\min}, \tau_{\max}]$, respectively, for a given pilot.

When off-nominal conditions are encountered during takeoff, FSAM transfers control to the envelope-aware (EA) safety controller that attempts LOC prevention or recovery. To ensure a safe rotation during takeoff, elevator inputs of safety EA controller $u_e(t)|_{\text{EA}}$ are modeled as follows:

$$u_e(t)|_{\text{EA}} = \begin{cases} \bar{K}_1(\bar{\theta}_{\text{ref}_1} - \theta(t)) + \bar{K}_2 q & \text{if } (\theta(t) < \theta_{\text{PR}} \;\&\; v < V_R) \\ \bar{K}_3(\bar{\theta}_{\text{ref}_2} - \theta(t)) + \bar{K}_4 q & \text{if } (\theta(t) \geq \theta_{\text{PR}} \;\&\; v < V_R) \\ \bar{K}_5(\bar{\theta}_{\text{ref}_3} - \theta(t)) + \bar{K}_6 q & \text{if } (\theta(t) < \theta_{\text{TS}} \;\&\; v \geq V_R) \\ \bar{K}_7(\bar{\theta}_{\text{ref}_4} - \theta(t)) + \bar{K}_8 q & \text{if } (\theta(t) \geq \theta_{\text{TS}} \;\&\; h(t) < h_{\text{TS}} \;\&\; v \geq V_R) \end{cases} \tag{3}$$

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 strikes; θ_{TS}, h_{TS} represents the threshold when tail strike protection is activated, whereas θ_{PR} represents the threshold when prevention against premature rotation is activated. Specific numerical values of parameters used in this work are given in the Appendix (see Table A3).

C. Safety Requirements for Takeoff Phase

The goal of the takeoff FSAM system is to prevent LOC during takeoff. Thus, the primary requirement for FSAM is to ensure that the system does not ever enter an unsafe state. A discussion of safe and unsafe states during takeoff can be found in [8]. For the purpose of illustration, in this paper, the primary focus is on verifying safety requirements or properties specified in part 25 [42] of the FAR (Airworthiness Standards: Transport Category Aircraft). This paper verifies that the longitudinal FSAM DMM (Fig. 2) meets the requirements listed in Table 1. These requirements can be found in the FAR part 25 under subpart 25.111. Table 1 also provides the LTL expression for each requirement. In Table 1, θ_{ng} is the pitch attitude at which the nose gear first leaves the ground, θ_0 is the pitch attitude that provides a nonnegative flight-path angle[§], θ_{tail} is the pitch attitude at which the tail contacts the ground before liftoff (i.e., when $h \leq h_{\text{lof}}$), and h_{obs} is the nominal obstacle clearance height, typically 35 ft for commercial aircraft [42].

[§]Note that, during takeoff, angle of attack α is positive and hence, a positive pitch attitude corresonds to a positive flight-path angle.

Table 2 Atomic propositions

Π_V	Π_θ	Π_H
$\pi_{v1} \coloneqq 0 \le v < V_{\text{mcg}}$	$\pi_{\theta 1} \coloneqq \theta_1 \le \theta < \theta_2$	$\pi_{H1} \coloneqq h_1 \le h < h_2$
$\pi_{v2} \coloneqq V_{\text{mcg}} \le v < V_1$	$\pi_{\theta 2} \coloneqq \theta_2 \le \theta < \theta_3$	$\pi_{H2} \coloneqq h_2 \le h < h_3$
$\pi_{v3} \coloneqq V_1 \le v < V_{r_{\min}}$	$\pi_{\theta 3} \coloneqq \theta_3 \le \theta < \theta_4$	$\pi_{H3} \coloneqq h_3 \le h < h_4$
$\pi_{v4} \coloneqq V_{r_{\min}} \le v < V_R$	$\pi_{\theta 4} \coloneqq \theta_4 \le \theta < \theta_5$	— —
$\pi_{v5} \coloneqq V_R \le v < V_{r_{\max}}$	$\pi_{\theta 5} \coloneqq \theta_5 \le \theta < \theta_6$	— —
$\pi_{v6} \coloneqq V_{r_{\max}} \le v < V_{\text{lof}}$	— —	— —
$\pi_{v7} \coloneqq V_{\text{lof}} \le v < V_2$	— —	— —
$\pi_{v8} \coloneqq V_2 \le v < V_{\text{fp}}$	— —	— —

D. Verification Problem Specification and Approach

Let $f(\cdot)$ denote the dynamics of takeoff and let $\text{Reach}(f, I)_{\mathcal{A}}$ denote the set of states reachable from the set of initial conditions I as governed by the switching strategy imposed by the FSAM DMM $\mathcal{A}$. Let $\mathcal{U}$ denote the set of unsafe states identified by the requirements (e.g., Table 1). The safety verification problem then reduces to checking the validity of the following expression [27]:

$$\text{Reach}(f, I)_{\mathcal{A}} \cap \bar{\mathcal{U}} = \phi \tag{4}$$

Computation of the reachable set $\text{Reach}(f, I)_{\mathcal{A}}$ can be challenging, especially if the underlying dynamics f is nonlinear [24]. Several authors have developed different approaches to compute reachable sets. The successes of these approaches are typically determined by the representations used to approximate the reachable sets. In this paper, a discrete overapproximation [27,43] of the dynamics in the form of a finite transition system is developed with the following steps: 1) Define a set of atomic propositions over the state space of the dynamics. These atomic propositions are used to express the requirements and also constitute inputs received by the FSAM DMM. 2) Abstract the dynamics as a finite transition system that takes into account the behavior of the pilot and the EA controller. 3) Compose the abstraction with FSAM to obtain an overapproximation of the closed-loop behavior. To verify Eq. (4), an automaton-theoretic approach is used wherein the overapproximation of the closed-loop behavior and system requirements (constraints) Φ are used as inputs to a model checker. The model checker searches for any violation of requirements Φ in the state space of the given model. If violations are detected, the model checker returns a counterexample (a sequence of states in the given model) that illustrates how a requirement is violated. In this work, the existing model checker SPIN [32] is used. The three steps of this model checking process are discussed in detail next.

IV. Verification of Takeoff FSAM

A. State-Space Abstraction

The first step to verification requires defining a set of atomic propositions for model checking. These preceding propositions capture thresholds essential to verify requirements. The requirements defined earlier are only related to airspeed V, pitch θ, and altitude h, yielding propositions Π_V, Π_θ, Π_H. Here, $\Pi_V = \{\pi_{v1}, \ldots, \pi_{v8}\}$ is the set of propositions that defines a discrete set of airspeed values, $\Pi_\theta = \{\pi_{\theta 1}, \ldots, \pi_{\theta 5}\}$ defines a discrete set of pitch values, and $\Pi_H = \{\pi_{H1}, \ldots, \pi_{H4}\}$ defines a discrete set of altitude values. The propositions (shown in Table 2) are chosen such that they partition the state space with sufficient resolution to capture safe versus unsafe states relevant to the requirements. The airspeed is partitioned with respect to the various V-speed constraints. The pitch and altitude states are partitioned to capture unsafe states such as tail strikes and premature rotations.[¶] For example, a tail strike (a state in which the tail of the aircraft strikes the runway) is identified by the propositions indicated in Fig. 3.

Next, an observation map $\mathcal{H}$: $\mathcal{X} \to 2^{\Pi}$ maps each state $X \in \mathcal{X}$ to atomic propositions in 2^{Π}. For example, let $X^* = [x, v^*, h^*, \theta^*, q]$, in which $0 \le v^* < V_{\text{mcg}}$, $\theta_1 \le \theta^* < \theta_2$, and $h_1 \le h^* < h_2$. In this case, $\mathcal{H}(X^*) = \{\pi_{v1}, \pi_{\theta 1}, \pi_{h1}\}$. Note that each state $X \in \mathcal{X}$ belongs to a polytope (a hyperrectangle) formed by the set $\{\mathcal{H}^{-1}(\pi_{vi}) \times \mathcal{H}^{-1}(\pi_{\theta j}) \times \mathcal{H}^{-1}(\pi_{\text{hk}})\}$. Thus, the set of states in $\mathcal{X}$ mapped to the same set of atomic propositions by $\mathcal{H}$ leads to a partition of the state space $\mathcal{X}$. A discrete state from a finite set $Q \coloneqq \{\bar{q}_1, \ldots, \bar{q}_n\}$ is associated with each element of this partition. As a result, the observation map induces the abstraction $\bar{T}$: $\mathcal{X} \to Q$, which maps each state $X \in \mathcal{X}$ into the finite set Q. The map $\bar{T}$ is proposition preserving if and only if

$$\bar{T}(X_1) = \bar{T}(X_2) \Rightarrow \mathcal{H}(X_1) = \mathcal{H}(X_2), \qquad \forall\, X_1, \qquad X_2 \in \mathcal{X} \tag{5}$$

Equation (5) indicates that any two states belonging to the same cell satisfy the same set of atomic propositions. Let $\mathcal{F}(\bar{T}^{-1}(\bar{q}))$ denote the set of $X \in \mathcal{X}$ that belongs to the facets of the polytope $\bar{T}^{-1}(\bar{q})$. (A facet of a polytope of n dimensions is a face that has $n-1$ dimensions.) In this paper, an element $\bar{q} \in Q$ is referred to as a cell instead of explicitly denoting a cell as $\bar{T}^{-1}(\bar{q})$, and $\text{q}_i \in \mathcal{F}(\bar{q})$ is referred to as the ith facet of cell $\bar{q}$.

The atomic propositions in Table 2 are chosen such that a switch to a different control authority dictated by FSAM always occurs at a facet of cell $\bar{q} \in Q$. Furthermore, a transition to an unsafe cell must pass through the facet of the unsafe cell. Hence, for the verification of FSAM according to Eq. (4), it is sufficient to assure that all reachable facets from a given initial facet do not contain any facets of unsafe cells. Thus, the goal is to find all possible transitions between facets of all cells in Q.

B. Discrete Representation of Reachable States

In principle, given a proposition-preserving partition and the dynamics, it is possible to use the methods in [7,25,34,44–46] to compute a discrete abstraction of the reachable states based on system dynamics. However, it is possible to simplify construction of the discrete abstraction by exploiting structural properties of the underlying dynamics and requirements. For instance, the states $v(t)$ and $h(t)$ as described by the dynamics in Eq. (1) are monotonically increasing functions of time in the region of interest. Furthermore, pitch response is governed by a linear system. These dynamics do not contain invariant sets in $\mathcal{X}$. The requirements discussed in Sec. III.C are only invariance requirements [13]. These properties simplify construction of a discrete abstraction of the reachable states.

[¶]Though this work uses an $8 \times 5 \times 3$ partition, any proposition-preserving partition with sufficient resolution could be used.

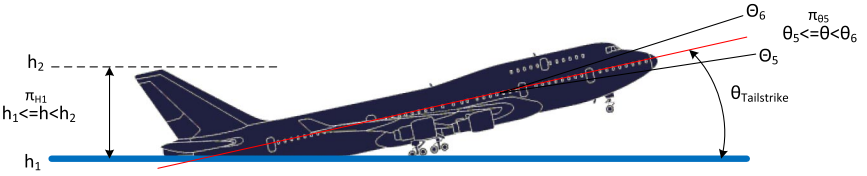

Fig. 3 Partitions that enable identification of a tail strike.

The method used to construct the discrete abstraction is shown in Algorithm 1. Inputs include discrete state-space partition Q, the takeoff dynamics model f, and the two controller formulations (P, EA) described by Eqs. (2) and (3). The algorithm returns a discrete transition system $\mathcal{B} \coloneqq (\mathcal{Q}, \mathcal{Q}_0, \mathcal{P}, \Gamma_{\mathcal{B}}, \Pi, \mathcal{L})$ for which the states $\mathcal{Q}$ represent facets of the cells in the partition. The actions $\mathcal{P}\{P, \text{EA}\}$ denote the two control authorities and $\Gamma_{\mathcal{B}}$ describes the transitions between facets under the two control authorities. The function *isReach* (q, k, q′) returns true if, for $k \in \{P, \text{EA}\}$, there exists $t_0, t_1, X(t_0) \in \text{q}, X(t_1) \in \text{q}'$ such that $X(t) \in \bar{q}$ for all $t \in [t_0, t_1]$, in which $\bar{q}$ is the cell containing the two facets q and q′. The function *isReach* (q, k, q′) can in general be evaluated using methods described in [25,26,46,47]. A description of the *isReach* (q, k, q′) function used in this work and specific numerical values describing the state-space partition can be found in the Appendix (see Table A3).

Algorithm 1 Algorithm to construct the discrete transition system

Inputs: state-space partitions Q, dynamics f, control inputs $u_e(t)|_P$, and $u_e(t)|_{\text{EA}}$
Initialize transition system $\mathcal{B} = (\mathcal{Q}, \mathcal{Q}_0, \mathcal{P}, \Gamma_{\mathcal{B}}, \Pi, \mathcal{L})$ in which $\mathcal{Q} = \text{q}|\text{q} \in \mathcal{F}(q), \forall\, q \in Q, P = P, \text{EA}, \Pi = \{\Pi_V, \Pi_H, \Pi_\Theta\}, \Gamma_{\mathcal{B}} = \{\}$
for k **in** $\{P, \text{EA}\}$
  for $\bar{q}$ **in** Q
    for q_i **in** $\mathcal{F}(\bar{q})$
      for q_j **in** $\mathcal{F}(\bar{q})$
        if (*isReach*$(\text{q}_i, k, \text{q}_j)$)
          //Add transition to discrete system $\mathcal{B}$ if valid transition exists.
          $\Gamma_{\mathcal{B}} = \Gamma_{\mathcal{B}} \cup \{(\text{q}_i, k, \text{q}_j)\}$
for $\bar{q}_i$ **in** Q
  for $\bar{q}_j$ **in** Q
    for q_m **in** $\mathcal{F}(q_i)$
      for q_n **in** $\mathcal{F}(q_j)$
        if $(\text{q}_m \equiv \text{q}_n)$
          //Add transitions between facets that are common to adjacent cells.
          $\Gamma_{\mathcal{B}} = \Gamma_{\mathcal{B}} \cup \{(\text{q}_m, k, \text{q}_n)\}$
Return $\mathcal{B}$

Each state $\text{q} \in \mathcal{Q}$ in $\mathcal{B}$ contains transitions induced by the pilot P and the envelope-aware safety controller. However, the goal of this paper is to verify transitions at each state that are governed by FSAM. Therefore, those transitions in $\mathcal{B}$ that are induced by the control authority dictated by the FSAM DMM at each discrete state $\text{q} \in \mathcal{Q}$ are extracted by constructing the composition (product) of the transition system $\mathcal{B}$ and FSAM DMM $\mathcal{A}$.

C. Composite Transition System

The composition of the discrete transition system $\mathcal{B} \coloneqq (\mathcal{Q}, \mathcal{Q}_0, \mathcal{P}, \Gamma_{\mathcal{B}}, \Pi, \mathcal{L})$ and the FSAM DMM $\mathcal{A} \coloneqq (\mathcal{S}, \mathcal{S}_0, \Sigma, \Lambda, \mathcal{T}, \mathcal{G})$ yields a new transition system $\mathcal{C} \coloneqq (\mathcal{D}, \mathcal{D}_0, \mathcal{P}, \Gamma_{\mathcal{C}}, \Pi, \mathcal{L})$, in which $\mathcal{D} = \mathcal{Q} \times \mathcal{S}$ and $\mathcal{D}_0 = \mathcal{Q}_0 \times \mathcal{S}_0$. Let $d_i, d_j \in \mathcal{D}$, in which $d_i = (\text{q}, s)$ and $d_j = (\text{q}', s')$. Then, $(d_i, p, d_j) \in \Gamma_{\mathcal{C}}$ if and only if $(\text{q}, p, \text{q}') \in \Gamma_{\mathcal{B}}$ and $(s, \sigma, s') \in \mathcal{T}$, in which $p = \mathcal{G}(s)$ and $\sigma = \mathcal{L}(\text{q}')$. In other words, the composite transition system denotes the parallel evolution of the states in the transition system $\mathcal{B}$ and FSAM DMM $\mathcal{A}$. Note that, in the composite transition system $\mathcal{C}$, the inputs p to discrete states q are the outputs of DMM state s. This ensures the composite transition system $\mathcal{C}$ contains only those transitions in $\mathcal{B}$ that are governed by the control authority selected by FSAM. This composite transition system serves as the model for the model checking process because it depicts the behavior of the aircraft during takeoff as governed by FSAM. Thus, the goal in verification is to ensure that model $\mathcal{C}$ satisfies the requirements imposed on takeoff in Sec. III.C. Figure 4 illustrates this composition process.

Proposition: If the composed transition system $\mathcal{C}$ does not violate the specifications, then the simplified dynamic model governed by the FSAM switching control law does not violate the specification.

Proof: By construction, the composed transition system $\mathcal{C}$ contains all behaviors of the simplified dynamics under FSAM's switching control law. Therefore, the reachable set of the composed system $\mathcal{C}$ contains the reachable set of the simplified dynamics under the switching control law governed by FSAM.

D. Model Checking

The requirements $\mathbf{\Phi}$ for model checking expressed in LTL with the propositions defined in the previous section are as follows:

$$
\begin{aligned}
\Phi_1 &\coloneqq \square((\theta \geq \theta_{\text{ng}}) \rightarrow (V \geq V_R)) = \square((\pi_{\theta 4} \vee \pi_{\theta 5}) \rightarrow (\pi_{v4} \vee \pi_{v5} \vee \pi_{v6})) \\
\Phi_2 &\coloneqq \square((h < h_{\text{lof}}) \rightarrow (\theta < \theta_{\text{tail}})) = \square((\pi_{H1} \vee \pi_{H2}) \rightarrow (\pi_{\theta 1} \vee \pi_{\theta 2} \vee \pi_{\theta 3} \vee \pi_{\theta 4})) \\
\Phi_3 &\coloneqq \square((h \geq h_{\text{lof}}) \rightarrow (\theta \geq \theta_0)) = \square[\neg(\pi_{H1} \vee \pi_{H2}) \rightarrow \neg\, \pi_{\theta 1}] \\
\Phi_4 &\coloneqq \square((h \geq h_{\text{obs}}) \rightarrow (V \geq V_2)) = \square((\pi_{H4}) \rightarrow \pi_{v6})
\end{aligned} \tag{6}
$$

The composed transition system $\mathcal{C}$ and requirements Φ are input into the SPIN model checker. Figure 5 illustrates an overview of model checking. If the model satisfies all requirements, the verification is considered complete. If violations exist, analysis of each counterexample is essential to understand why requirements are violated, as well as what changes to the logic or control laws are needed to prevent such violations. Analysis can help distinguish counterexamples that could be false positives. Most often, false positives are artifacts of the abstraction technique itself, and so it is possible to use these counterexamples to refine the abstractions in a manner that eliminates false positives [48,49].

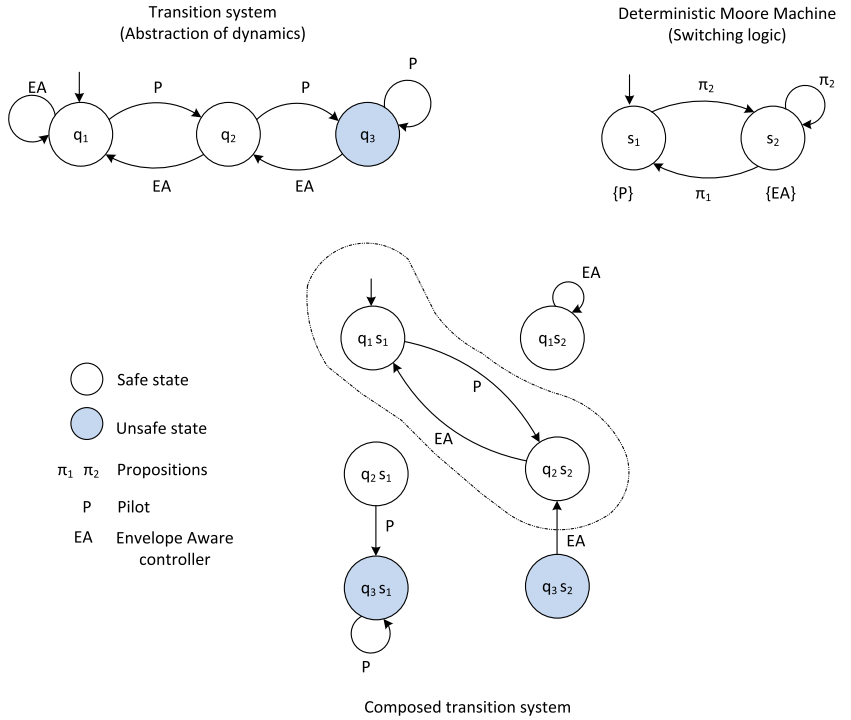

Fig. 4 Composition of a transition system with a DMM.

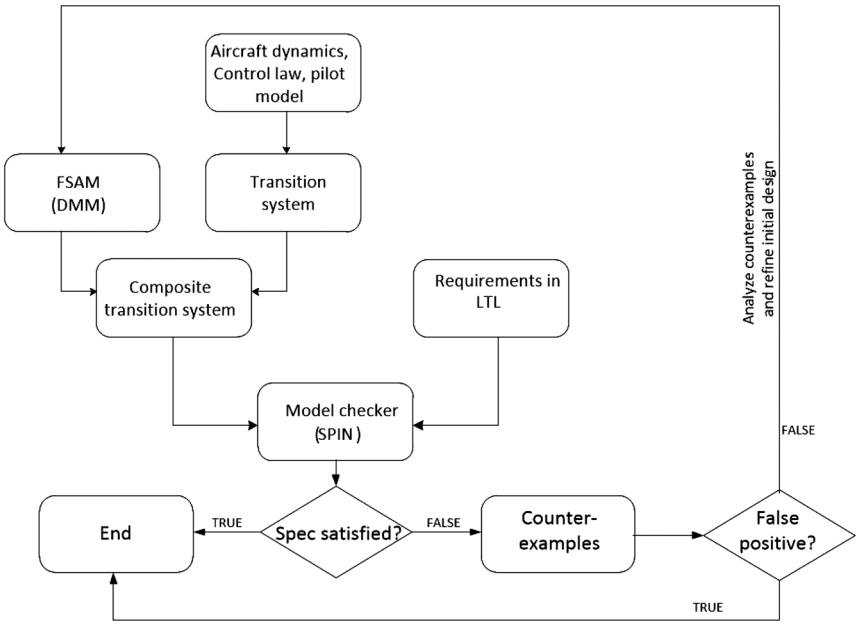

Fig. 5 Model checking process.

Using the preceding model checking approach, requirements Φ_1 and Φ_2 were violated with the baseline DMM shown in Fig. 2. In other words, the underlying FSAM logic could not prevent premature rotations and tail strikes.

V. Refinement of FSAM

As discussed earlier, the model checker revealed that requirements Φ_1 and Φ_2 were violated in $\mathcal{C}$. Three causes were identified: 1) specific pilot behaviors could result in the violation, 2) the EA controller could be poorly designed and/or inadequate to deal with the off-nominal conditions, and/or 3) the switching logic (FSAM DMM) might be incorrect/incomplete. According to the system dynamics in Eq. (1), Φ_1 could be violated if the pilot rotates the nose of the aircraft in the $V_{\text{mcg}} \leq V < V_1$ airspeed range. This is because protection against premature rotation while in the $V_{\text{mcg}} \leq V < V_1$ airspeed range was not available in $\mathcal{A}$. Φ_2 could be violated if the pilot chose to delay the rotation until after achieving V_{lof} speed was reached due to an omission of tail strike protection in $\mathcal{A}$ outside the airspeed range $V_R \leq V < V_{\text{lof}}$. After analyzing the preceding counterexamples, appropriate changes to the FSAM's DMM $\mathcal{A}$ were made. These changes are highlighted in Fig. 6 and were also carried into our archival FSAM [8] DMM specification. The updated DMM protects from tail strikes via new transitions, indicated by the dashed lines. Transition $(s_3 \rightarrow s_9)$ prevents a premature rotation initiated before V_1 and the transition $(s_{10} \rightarrow s_{11})$ prevents a tail strike by activating the tail strike protection EA controller if the aircraft is still on the ground after the V_{lof} airspeed. Model checking was repeated with the updated DMM, verifying that both requirements Φ_1 and Φ_2 were now satisfied.

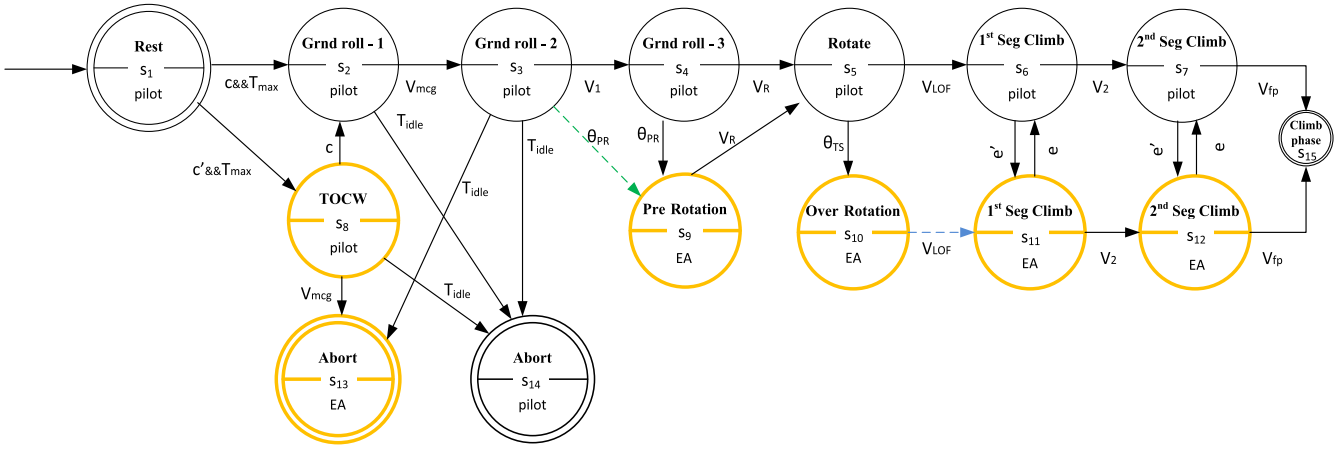

Fig. 6 Revised FSAM DMM.

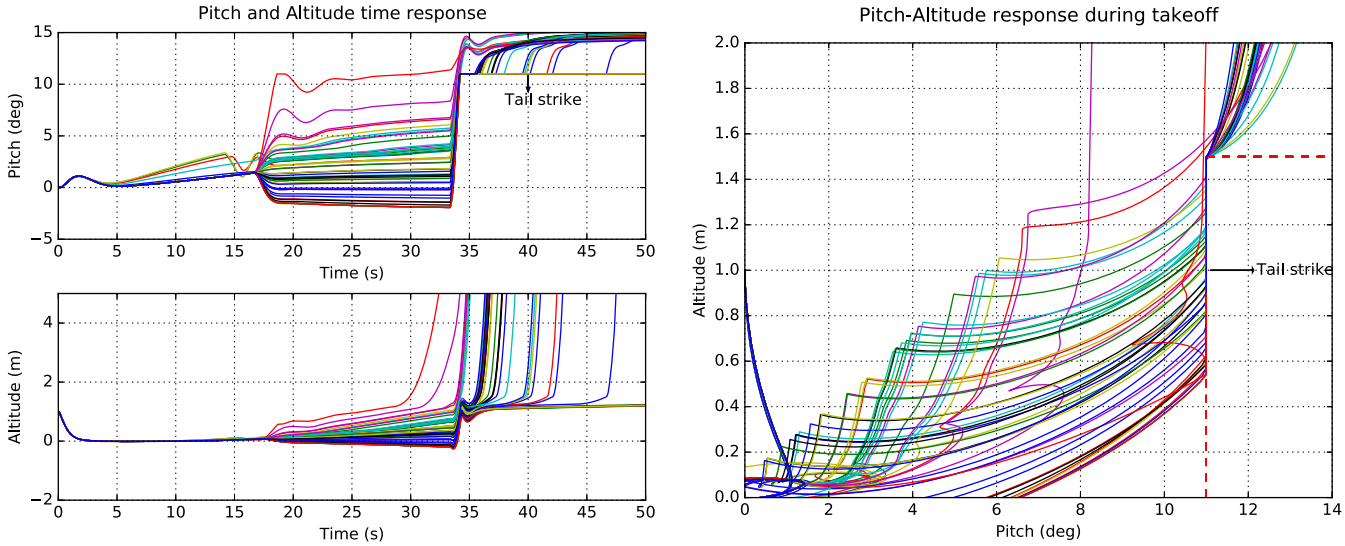

Fig. 7 Monte Carlo simulations of takeoff phase with original FSAM DMM.

A Monte Carlo analysis was performed using the full nonlinear equations of motion describing takeoff dynamics (see [8] for a description of the full nonlinear dynamics). The pilot and EA controller inputs were chosen as described in Eqs. (2) and (3). Switching between control authorities was determined by the FSAM DMM. The EA controller [Eq. (3)] was deterministic. For the pilot model [Eq. (2)], parameters characterizing pilots' behavior (θ_{ref}, V_r, K_p, K_d, τ) were randomly sampled from a uniform distribution within the 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, for each Monte Carlo trial. The domain of the parameters are chosen such that the different pilot behaviors of interest can be simulated. For example, sampling V_r from low airspeed ranges forces the pilot to initiate pitch rotation prematurely. Similarly, by sampling K_p, θ_{ref} from high-gain and high-pitch angle ranges, respectively, tail-strike events are observed. Each Monte Carlo trial is initialized at $x = 0$, $v = 0$, $h = 0$, $\theta = 0$, and $q = 0$. The numerical values of all parameters used in the Monte Carlo simulation are listed in Table A3. Figure 7 illustrates the aircraft responses after several Monte Carlo trials with the original uncorrected FSAM DMM. Figure 7 shows many instances of tail strikes (i.e., $\theta \geq \theta_{\text{tail}}$ and $h < h_{\text{lof}}$) even though the original DMM was formulated to prevent such scenarios. The Monte Carlo trials with the corrected DMM exhibited no high-risk rotation or tail-strike events (see Fig. 8).

VI. Validation of FSAM and FAR

The Monte Carlo simulations discussed in the preceding section confirm that the refinements made in response to the counterexamples obtained from model checking prevent the occurrence of tail strikes. The model checking approach formally guarantees that the FSAM logic is correct with respect to the specified requirements, takeoff dynamics, and pilot and EA controller models. However, verification of FSAM with respect to FAR requirements may be insufficient to ensure safety across the spectrum of real-world missions. This leads to additional questions: Are the right requirements being enforced, and is the takeoff logic complete with respect to LOC prevention? These questions are addressed with a scenario aimed to provoke careful thought about generalized requirements.

Consider a scenario in which a general aviation (GA) aircraft executes a soft-field takeoff (e.g., from a grass strip after a recent rain).** The goal of a soft-field takeoff is to minimize the load on the nose gear and become airborne as soon as possible. Soft-field takeoff operating procedures require maintaining a nose-up attitude during the initial ground roll. This enables the airplane to become airborne. The airplane then accelerates while in ground effect until the required climb speed is achieved. In this scenario, requirement Φ_1 may not help the pilot establish a safe takeoff, particularly if FSAM and the EA controller were not analyzed with consideration of the soft-field takeoff. The transition $s_3 \rightarrow s_9$ according to the revised DMM in Fig. 6 would then potentially prevent the pilot from maintaining acceptable loads on the nose gear. This may result in the nose

**GA is covered in a separate FAR section, but tail strike is still an issue.

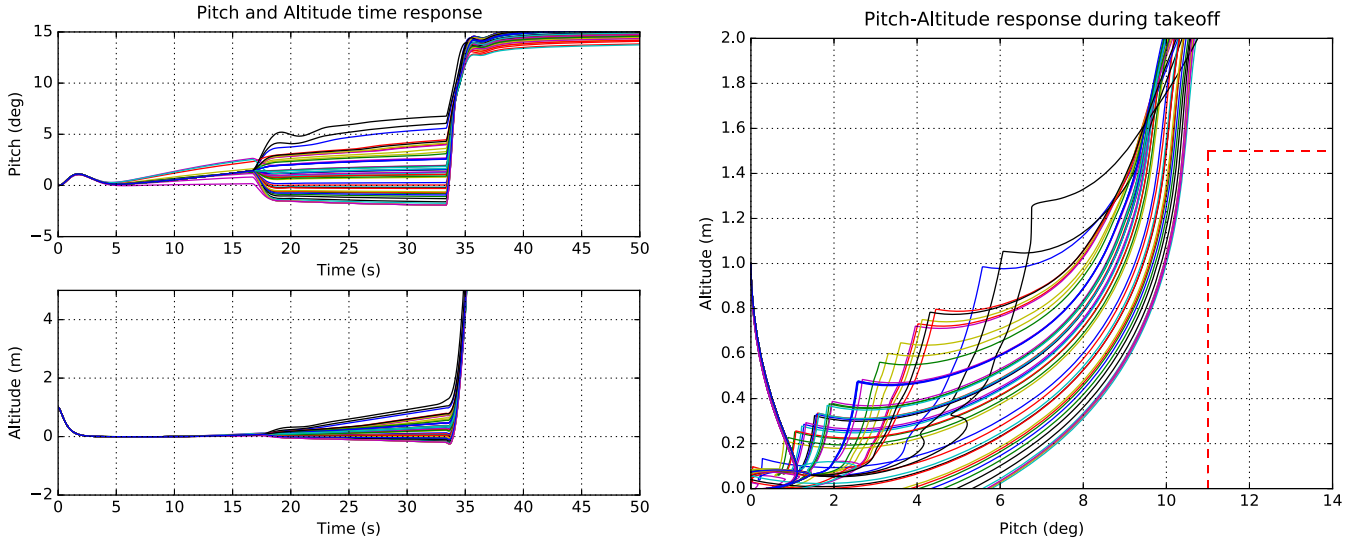

Fig. 8 Monte Carlo simulations of the takeoff phase with revised FSAM DMM.

gear digging into the soft field, increasing the rolling friction and potentially leading to runway excursion or even tipover in an extreme case. With a short as well as soft field, failing to efficiently become airborne may also lead to poor climb performance or runway excursion.

The FSAM DMM has been revised (see Fig. 6) to ensure satisfaction of FAR requirement Φ_1, but it may not be valid with respect to a GA aircraft performing a soft-field takeoff. Typically, a conflict identified during validation can be addressed by modifying the initial requirements to accommodate the conflicting operational needs or, if possible, by modifying the design of a system to address the conflict. Thus, it is essential to modify the requirement Φ_1 according to runway type and also modify the design of $\mathcal{A}$ by adding states and transitions that account for soft-field takeoffs. An FSAM DMM for a soft-field takeoff would allow early rotation, while preventing a tail strike. It would also prevent excessive nose-down control inputs to minimize load on the nose wheel. It is worth noting that FAR requirement Φ_1 is not complete with respect to different takeoff strategies, such as the cited soft-field takeoff example. This example illustrates the importance of applying each requirement in exactly those contexts in which it is actually required. As this example illustrates, the FSAM verification process must always take into account pertinent operational requirements in addition to baseline FAR.

VII. Discussion

This paper proposed a model checking framework to verify and (manually) refine, when necessary, the design of an FSAM system against safety requirements. Pilot behavior was encoded using an uncertain transfer function model, and an envelope-aware controller was used for the autopilot mode. Simplified equations for takeoff dynamics were used to construct an overapproximation on which model checking was performed. The simplified dynamics presented in this paper adequately captures events such as premature rotation, tail strikes, and runway overruns. Also, this model leverages the underlying structural properties such as monotonicity and linearity required to construct the discrete transition system.

The process of constructing the discrete transition system that describes the reachable facets for each controller (P and EA) separately, and then merging them according to the switching strategy imposed by FSAM, promotes understanding of how each controller affects the nominal system. This enables a comprehensive analysis of counterexamples obtained from the model checker, which in turn facilitates identifying necessary changes to underlying DMM logic. It also leads to incremental changes in the design. Note that other anomalous or exceptional conditions (wind, loading, performance, system failure, etc.) must also be considered in the requirements and DMM for a comprehensive takeoff DMM capability. Achieving truly complete knowledge of behaviors remains a challenge for system designers, as well as both automation and human crews.

In this work, the discrete states in the transition systems abstractly represented facets of the cells in the state-space partition. Algorithm 1 explicitly enumerates all cells in the state-space partition and checks for transitions between facets of a given cell. This can become tedious, especially if there are a large number of cells in the state-space partition. However, it is possible to only check the transitions between facets of cells that are reachable from a given initial cell. It is also possible to consider an abstraction that directly represents the cells instead of the cell facets. In this case, it is sufficient to check the transitions between cells instead of facets. This in turn would speed up construction of discrete transition system ($\mathcal{B}$). However, this type of abstract representation yielded many counterexamples during model checking, due to nondeterminism induced in the abstract model, which turned out to be false positives.

It is also possible to automate the FSAM logic refinement process using tools such as Counterexample-Guided Abstraction Refinement (see Refs. [48,49]). However, automating refinement risks a final DMM result that is not physically intuitive or readable. For a manually constructed DMM, the DMM design team needs to also verify that modifications are consistent with user interface needs.

The full aircraft takeoff dynamics model described in [8,9] is a higher-order nonlinear model that combines a traditional aircraft dynamics model with the landing gear (oleo strut and wheel) dynamics and facilitates modeling the aircraft's response to differential braking inputs and nose wheel steering inputs during takeoff. In principle, it is possible to consider more complex nonlinear dynamics within the proposed framework and use methods described in [7,25–27,46,47,50] within Algorithm 1.

VIII. Conclusions

This paper contributes a model checking framework that enables formal verification of manually constructed DMM formulations and applies this method to the takeoff FSAM system. The switched systems is verified via three main steps: 1) select an abstract representation of the underlying state space, 2) construct a discrete transition system that overapproximates the reachable states under the various control authorities, and 3) compose the discrete transition system and the switching logic represented as a DMM. This verification procedure is applied to an FSAM DMM for takeoff based on FAR part 25 safety requirements. Simplifying assumptions enable leveraging existing algorithms to perform

reachability analysis and model checking. Model checking results were also cross validated with a Monte Carlo analysis using full nonlinear dynamics to eliminate false positives. This paper has also illustrated that model checking can be used to guide/help a system engineer to refine the system design in addition to proving correctness of the system.

For a comprehensive verification of FSAM, one needs to consider different scenarios, such as rejected takeoffs, engine failure scenarios, crosswind conditions, and more. In such cases, the simplified models described in this paper must be replaced by models that can adequately capture the behaviors of interest for verification. Abstractions should also consider other state variables such as heading and longitudinal and cross-track position to capture safe versus unsafe states. It is important to recognize that FSAM only activates when safety is verifiable, and so unhandled cases will result in a need for appropriate crew response. Selecting the right set of requirements plays a crucial role in validating the system. To facilitate verification of FSAM against complex scenarios, work is underway to develop a statistical model checking framework that makes use of Monte Carlo simulations to establish probabilistic guarantees on requirement satisfaction. The use of formal methods reduces the need to run extensive flight tests to study the behavior of the overall system. However, a number of factors, such as pilot interfaces and acceptance, must still be considered. Although work remains, the deterministic models presented in this paper are verifiable and thus ultimately certifiable using current regulation practices.

Appendix A: Longitudinal Deterministic Moore Machine [8]

The longitudinal FSAM DMM governs the decision-making process with respect to the longitudinal dynamics of the aircraft during takeoff. It is designed to prevent events that could severely impact ground roll performance of the aircraft, including inappropriate crew inputs such as improper rejected takeoff and improper rotations. Because the V speeds are vital for takeoff decision making and are intuitive for pilots, a state formulation for the DMM that captures the critical V-speed thresholds is employed.

The longitudinal (lg) Moore machine $\mathcal{A}_{\text{lg}}$ is represented as the tuple $(\mathcal{S}_{\text{lg}}, \mathcal{S}_{\text{lg}\,0}, \Sigma_{\text{lg}}, \Lambda_{\text{lg}}, \mathcal{T}_{\text{lg}}, \mathcal{G}_{\text{lg}})$, in which

$$\mathcal{S}_{\text{lg}} = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}\} \tag{A1}$$

$$\mathcal{S}_{\text{lg}\,0} = \{s_1\} \tag{A2}$$

$$\Sigma_{\text{lg}} = \{V_{\text{mcg}}, V_1, V_R, V_{\text{lof}}, V_2, V_{\text{fp}}, T_{\text{idle}}, T_{\text{max}}, c, c', e, e', f, \theta, \bar{\theta}\} \tag{A3}$$

$$\Lambda_{\text{lg}} = \{P, \text{EA}\} \tag{A4}$$

$$\mathcal{G}_{\text{lg}} = \begin{cases} P & \text{if } s_i \in \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_{14}\} \\ \text{EA} & \text{otherwise} \end{cases} \tag{A5}$$

Transitions $\mathcal{T}_{\text{lg}}$ are illustrated as edges in a directed graph (Fig. 2). The definition of each alphabet symbol in the set Σ_{lg} is provided in Table A1. In this work, it is assumed that the engines, instruments, and control surfaces function nominally and there is no need for a rejected takeoff. Also, the aircraft is appropriately configured for takeoff and does not exit the flight envelopes after becoming airborne. Consequently, the DMM execution does not encounter the T_{idle}, c', f, and e' input symbols. The relation between the DMM alphabet symbols used in this work and the atomic propositions required to construct the composite transition system discussed in Sec. IV.C is provided in Table A2.

Table A1 Input alphabet symbols for the takeoff Moore machine

Alphabet Σ	Description
V_{mcg}	Minimum controllable ground speed with one engine inoperative
V_1	Takeoff decision speed (go/no-go speed)
V_R	Rotation speed
V_{lof}	Liftoff speed
V_2	Takeoff safety speed
V_{fp}	Minimum flap retraction speed
T_{max}	Takeoff thrust setting
T_{idle}	Idle thrust setting
c	Aircraft configured for takeoff
c'	Improper takeoff configuration
e	Envelope protection deactivated
e'	Envelope protection activated
f	Inadequate acceleration performance
θ_{PR}	Premature rotation
θ_{TS}	Maximum allowable pitch attitude reached during rotation

Table A2 Mapping between DMM alphabet symbols and atomic propositions

Alphabet	Propositions
V_{mcg}	π_{v2}
V_1	$\pi_{v3} \vee \pi_{v4}$
V_R	$\pi_{v5} \vee \pi_{v6}$
V_{lof}	π_{v7}
V_2	π_{v8}
θ_{PR}	$\pi_{\theta 2} \wedge \pi_{H1}$
θ_{TS}	$\pi_{\theta 5} \wedge \pi_{H1}$

Appendix B: Reachability Analysis

Algorithm A1 describes the *isReach*( ) function used by Algorithm 1. It takes as inputs two facets q and q′ of a cell and returns true, if under the current control authority p_k there exists a trajectory starting from q and ending in q′, while remaining within the cell containing the two facets. The main idea in Algorithm A1 is to exploit the fact that the airspeed and altitude [in Eq. (1)] monotonically increase with time (in the region of interest) and the pitch dynamics is piecewise affine, therefore, it is enough to propagate the extreme points of the facet. Algorithm A1 propagates the initial condition $\bar{X}_0 = [x_0, v_0, h_0, \theta_0, q_0]$ obtained from each vertex of facet q, until the ensuing trajectory leaves the cell, to determine if facet q′ is reachable from q. (To be more precise, one can propagate δ expansions of facets because a time-sampled trajectory might not intersect the facet, but will be within some δ neighborhood of it, in which δ can be inferred from the sampling time and the Lipschitz constant of the dynamics [36].) Because this work considers discrete-time semantics of LTL, state propagation is performed using the discrete-time version of the system dynamics in Eq. (1). In Algorithm A1, $\bar{f}$ denotes the discrete-time equivalent of f. Each vertex of facet q provides initial conditions for airspeed v_0, pitch θ_0, and altitude h_0. The longitudinal position x_0 is initialized at zero because the requirements considered in this paper do not impose restrictions on x. If $v < V_r$, the pitch rate q_0 is initialized at zero. This is because the pitch remains constant until rotation, and therefore the pitch rate is zero. However, for $v \geq V_r$, $q_0 \in \{q_{\min}, q_{\max}\}$. Here, $q_{\min}$ and $q_{\max}$ denote the minimum and maximum attainable pitch rate during takeoff. $\mathcal{V}(\text{q}) \subset \mathbb{R}^5$ denotes the set of initial conditions for a facet q. Note that, because the pilot model described by Eq. (2) consists of the parameters $\theta_{\text{ref}}, V_r, K_p, K_d$, and τ, for which the values are assumed to lie within bounded intervals, each initial condition for $p_k = P$ is propagated for all possible extreme values of the parameters (i.e, $K_p \in \{K_{p_{\min}}, K_{p_{\max}}\}$, $K_d \in \{K_{d_{\min}}, K_{d_{\max}}\}, \theta_{\text{ref}} \in \{\theta_{\text{ref}_{\min}}, \theta_{\text{ref}_{\max}}\}, V_r \in \{V_{r_{\min}}, V_{r_{\max}}\}$, and $\tau \in \{\tau_{\min}, \tau_{\max}\}$). For $p_k = \text{EA}$, the controller parameters are known exactly and hence each initial condition is propagated only once. $\mathcal{K}(p_k, X)$ denotes the set of controller parameters for the given control authority p_k that are used to construct the control law. U_η denotes the controller input constructed according to Eq. (2) (when $p_k = P$) or Eq. (3) (when $p_k = \text{EA}$) using the parameters $\eta = (\theta_{\text{ref}}, V_r, K_p, K_d, \tau)$. Note that Algorithm A1 requires the initialization of the delay term in the pilot control mode described by Eq. (2). This is achieved by reversing the dynamics and estimating upper and lower bounds on $\theta[n-m]$ for all $1 \leq i \leq m$.

Algorithm A1 Function *isReach*( )

Function: *isReach*(q, p_k, q′)
for $\bar{X}_0$ **in** $\mathcal{V}$(q)
  $X = \bar{X}_0$
  for η **in** $\mathcal{K}(p_k, X)$
    $\bar{q}_i \coloneqq \bar{q}_j \coloneqq \mathcal{F}^{-1}(\text{q})$
    where as $\bar{q}_{i_-} = \bar{q}_i$
      $X' \coloneqq \underline{f}(X, U_\eta)$
      $\bar{q}_j \coloneqq \bar{T}(X')$
      $X' \coloneqq X$
      if $\bar{q}_j \neq \bar{q}_i$
        if $\text{q}' \in \mathcal{F}(\bar{q}_i) \cap \mathcal{F}(\bar{q}_j)$
        return true
  return false

Appendix C: Numerical Values

Parameter numerical values used in this paper are given in Table A3. The aircraft physical parameters, such as m, S_{ref}, and $\bar{c}$, were obtained from [51]. The maximum thrust value was obtained from [52]. C_{L_g}, C_{D_g}, μ were chosen based on [40].

Table A3 Numerical parameters

Parameters	Values
$m, S_{\text{ref}}, I_{yy}$	45,420 kg, 122.4 m^2, 0.3172e7 kg · m^2
$\rho, \bar{c}$	1.225 kg · m^{-3}, 4.19 m
$\alpha_0, \gamma_0, T_{\max}$	0°, 8°, 300 kN
C_{L_g}, C_{D_g}, μ	1.2, 0.05, 0.1
$C_{m_q}, C_{m_{ue}}$	−44.43, −1.785
A_3, B_3	$(C_{m_q}, C_{m_{ue}}) \times (1/2I_{yy})\rho V_{\text{mcg}}^2 S_{\text{ref}} \bar{c}$
A_4, B_4	$(C_{m_q}, C_{m_{ue}}) \times (1/2I_{yy})\rho V_R^2 S_{\text{ref}} \bar{c}$
$\bar{K}_1, \bar{K}_3, \bar{K}_5, \bar{K}_7$	−3
$\bar{K}_2, \bar{K}_4, \bar{K}_6, \bar{K}_8$	0.1
$\theta_{\text{PR}}, \theta_{\text{TS}}, \theta_{\text{ng}}, \theta_{\text{tail}}, h_{\text{TS}}, h_{\text{lof}}$	3°, 9°, 3°, 10°, 0.9 m, 2.5 m
$(\theta_{\text{ref}_{\min}}, \theta_{\text{ref}_{\max}}), (V_{r_{\min}}, V_{r_{\max}})$	(7°, 13°), (50, 70) ms^{-1}
$(K_{p_{\min}}, K_{p_{\max}}), (\tau_{\min}, \tau_{\max})$	(−5, −1), (0, 0.1) s
$V_{\text{mcg}}, V_1, V_R, V_{\text{lof}}, V_2, V_{\text{fp}}$	10, 47, 55, 66, 67.5, 80 ms^{-1}
$\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$	−2°, 3°, 5°, 8°, 9°, 15°
h_1, h_2, h_3, h_4	−5, 1, 2.5, 15 m
No. of Monte Carlo trials	1000

C_{m_q}, $C_{m_{\dot{\alpha}}}$ were chosen from [53]. The remaining parameters were chosen to reflect a typical twin-jet transport aircraft such as B737 or A320.

Acknowledgement

This work was supported in part by NASA under Cooperative Agreement NNX12AM54A.

References

[1] Belcastro, C. M., and Foster, J. V., "Aircraft Loss of Control Accident Analysis," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2010-8004, 2010.
[2] Borst, C., Grootendorst, F. H., Brouwer, D. I. K., Bedoya, C., Mulder, M., and van Paassen, M. M., "Design and Evaluation of a Safety Augmentation System for Aircraft," *Journal of Aircraft*, Vol. 51, No. 1, 2013, pp. 12–22. doi:10.2514/1.C031500
[3] 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.
[4] 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, 2009.
[5] Inagaki, T., "Situation-Adaptive Autonomy: Dynamic Trading of Authority Between Human and Automation," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 44, SAGE Publ., Thousand Oaks, CA, 2000, pp. 13–16. doi:10.1177/154193120004401304
[6] 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
[7] 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
[8] Balachandran, S., and Atkins, E. M., "Flight Safety Assessment and Management for Takeoff Using Deterministic Moore Machines," *Journal of Aerospace Information Systems*, Vol. 12, No. 9, Nov. 2015, pp. 599–615. doi:10.2514/1.I010350
[9] 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.
[10] Balachandran, S., and Atkins, E. M., "Flight Safety Assessment and Management During Takeoff," *AIAA Infotech@Aerospace Conference*, AIAA Paper 2013-4805, 2013.
[11] Sokolowski, J. A., Banks, C. M., and Petty, M. D., *Principles of Modeling and Simulation: A Multidisciplinary Approach*, Wiley, Hoboken, NJ, 2008, pp. 121–147, Chap. 6.
[12] Jacklin, S. A., Lowry, M. R., Schumann, J. M., Gupta, P., Bosworth, J. T., Zavala, E., Kelly, J., Hayhurst, K. J., Belcastro, C. M., and Belcastro, C. M., "Verification, Validation, and Certification Challenges for Adaptive Flight-Critical Control System Software," *AIAA Guidance, Navigation and Control Conference and Exhibit*, AIAA Paper 2004-5258, 2004.
[13] Baier, C., and Katoen, J. P., *Principles of Model Checking*, Vol. 26202649, MIT Press, Cambridge, MA, 2008, pp. 19–82, Chap. 2.
[14] Bjørner, N., Browne, A., Chang, E., Colón, M., Kapur, A., Manna, Z., Sipma, H. B., and Uribe, T. E., "STeP: Deductive-Algorithmic Verification of Reactive and Real-Time Systems," *Computer Aided Verification*, Springer–Verlag, New York, 1996, pp. 415–418.
[15] Bochot, T., Virelizier, P., Waeselynck, H., and Wiels, V., "Model Checking Flight Control Systems: The Airbus Experience," *31st International Conference on Software Engineering — Companion Volume, 2009, ICSE-Companion 2009*, IEEE Publ., Piscataway, NJ, May 2009, pp. 18–27. doi:10.1109/ICSE-COMPANION.2009.5070960
[16] Tribble, A. C., and Miller, S. P., "Safety Analysis of Software Intensive Systems," *IEEE Aerospace and Electronic Systems*, Vol. 19, No. 10, 2004, pp. 21–26. doi:10.1109/MAES.2004.1365014
[17] Tribble, A. C., and Miller, S. P., "Software Safety Analysis of a Flight Management System Vertical Management Function—A Status Report," *Proceedings of the 22nd Digital Avionics Systems Conference*, Vol. 1, IEEE Publ., Piscataway, NJ, Oct. 2003, pp. 1.B.1–1.1-9. doi:10.1109/DASC.2003.1245805
[18] Joshi, A., Heimdahl, M. P. E., Miller, S. P., and Whalen, M. W., "Model-Based Safety Analysis," NASA CR- 2006-213953, May 2006, http://shemesh.larc.nasa.gov/fm/papers/Joshi-CR-2006-213953-Model-Based-SA.pdf [retrieved 10 Sept. 2014].
[19] Bak, S., Manamcheri, K., Mitra, S., and Caccamo, M., "Sandboxing Controllers for Cyber-Physical Systems," *IEEE/ACM International Conference on Cyber-Physical Systems (ICCPS)*, IEEE Publ., Piscataway, NJ, 2011, pp. 3–12.
[20] Lygeros, J., and Lynch, N., "On the Formal Verification of the TCAS Conflict Resolution Algorithms," *Proceedings of the 36th IEEE Conference on Decision and Control*, Vol. 2, IEEE Publ., Piscataway, NJ, 1997, pp. 1829–1834.
[21] Degani, A., and Heymann, M., "Formal Verification of Human-Automation Interaction," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Vol. 44, No. 1, 2002, pp. 28–43. doi:10.1518/0018720024494838
[22] Bolton, M. L., Siminiceanu, R. I., and Bass, E. J., "A Systematic Approach to Model Checking Human–Automation Interaction Using Task Analytic Models," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 41, No. 5, 2011, pp. 961–976. doi:10.1109/TSMCA.2011.2109709
[23] Alur, R., "The Theory of Timed Automata," *Theoretical Computer Science*, Vol. 126, No. 2, 1999, pp. 183–235. doi:10.1016/0304-3975(94)90010-8
[24] Tabuada, P., *Verification and Control of Hybrid Systems: A Symbolic Approach*, Springer–Verlag, New York, 2009, pp. 23–50.
[25] Girard, A., "Reachability of Uncertain Linear Systems Using Zonotopes," *Hybrid Systems: Computation and Control*, Springer–Verlag, New York, 2005, pp. 291–305.
[26] Habets, L., Collins, P. J., and Van Schuppen, J. H., "Reachability and Control Synthesis for Piecewise-Affine Hybrid Systems on Simplices," *IEEE Transactions on Automatic Control*, Vol. 51, No. 6, 2006, pp. 938–948. doi:10.1109/TAC.2006.876952
[27] Kloetzer, M., and Belta, C., "Reachability Analysis of Multi-Affine Systems," *Hybrid Systems: Computation and Control*, Springer–Verlag, New York, 2006, pp. 348–362.
[28] Prajna, S., and Jadbabaie, A., "Safety Verification of Hybrid Systems Using Barrier Certificates," *Hybrid Systems: Computation and Control*, Springer–Verlag, New York, 2004, pp. 477–492.
[29] Zuliani, P., Platzer, A., and Clarke, E. M., "Bayesian Statistical Model Checking with Application to Simulink/Stateflow Verification," *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control*, New York, NY, 2010, pp. 243–252.
[30] 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*, New York, NY, 2012, pp. 125–134.

[31] Moore, E. F., "Gedanken-Experiments on Sequential Machines," *Automata Studies*, Princeton Univ. Press, Princeton, NJ, 1956, pp. 129–153.
[32] 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
[33] Manna, Z., and Pnueli, A., *Temporal Logic of Reactive and Concurrent Systems: Specifications*, Vol. 1, Springer–Verlag, Berlin, 1992, pp. 179–273.
[34] Liu, J., Ozay, N., Topcu, U., and Murray, R. M., "Synthesis of Reactive Switching Protocols from Temporal Logic Specifications," *IEEE Transactions on Automatic Control*, Vol. 58, No. 7, 2013, pp. 1771–1785.
[35] Kress-Gazit, H., Fainekos, G. E., and Pappas, G. J., "Where's Waldo? Sensor-Based Temporal Logic Motion Planning," *IEEE International Conference on Robotics and Automation*, IEEE Publ., Piscataway, NJ, 2007, pp. 3116–3121.
[36] Liu, J., and Ozay, N., "Abstraction, Discretization, and Robustness in Temporal Logic Control of Dynamical Systems," *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control*, New York, NY, 2014, pp. 293–302.
[37] "Pilot Guide to Takeoff Safety [Online Database]," Federal Aviation Administration, http://www.faa.gov/other_visit/aviation_industry/airline_operators/training/media/takeoff_safety.pdf [retrieved 01 Nov. 2012].
[38] Mamessier, S., Feigh, K., Pritchett, A., and Dickson, D., "Pilot Mental Models and Loss of Control," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2014-0609, 2014.
[39] McRuer, D. T., and Krendel, E. S., "Mathematical Models of Human Pilot Behavior," Distributed by National Technical Information Service (NTIS) AGARD-AG-188, Sprinfield, VA, 1974.
[40] Roskam, J., and Lan, C. T. E., *Airplane Aerodynamics and Performance*, DARcorporation, Lawrence, KS, 1997, pp. 435–507, Chap. 10.
[41] 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
[42] "Part 25—Airworthiness Standards: Transport Category Airplanes," Electronic Code of Federal Regulations e-CFR [online], Federal Aviation Administration, http://www.ecfr.gov/cgi-bin/text-idx?SID=5dc4d5058a9ad16943a2af08556801cd&tpl=/ecfrbrowse/Title14/14cfr25_main_02.tpl [retrieved 01 May 2015].
[43] Alur, R., Henzinger, T. A., Lafferriere, G., and Pappas, G. J., "Discrete Abstractions of Hybrid Systems," *Proceedings of the IEEE*, Vol. 88, No. 7, 2000, pp. 971–984. doi:10.1109/5.871304
[44] 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, June 2014, pp. 3273–3280. doi:10.1109/ACC.2014.6859428
[45] Lygeros, J., "On Reachability and Minimum Cost Optimal Control," *Automatica*, Vol. 40, No. 6, 2004, pp. 917–927. doi:10.1016/j.automatica.2004.01.012
[46] Asarin, E., Bournez, O., Dang, T., and Maler, O., "Approximate Reachability Analysis of Piecewise-Linear Dynamical Systems," *Hybrid Systems: Computation and Control*, Springer–Verlag, Berlin, 2000, pp. 20–31.
[47] Girard, A., and Martin, S., "Control Synthesis for Constrained Nonlinear Systems Using Hybridization and Robust Controllers on Simplices," *IEEE Transactions on Automatic Control*, Vol. 57, No. 4, 2012, pp. 1046–1051.
[48] Clarke, E., Grumberg, O., Jha, S., Lu, Y., and Veith, H., "Counterexample-Guided Abstraction Refinement for Symbolic Model Checking," *Journal of the ACM*, Vol. 50, No. 5, 2003, pp. 752–794. doi:10.1145/876638
[49] Clarke, E., Fehnker, A., Han, Z., Krogh, B., Ouaknine, J., Stursberg, O., and Theobald, M., "Abstraction and Counterexample-Guided Refinement in Model Checking of Hybrid Systems," *International Journal of Foundations of Computer Science*, Vol. 14, No. 4, 2003, pp. 583–604. doi:10.1142/S012905410300190X
[50] Asarin, E., Dang, T., and Girard, A., "Reachability Analysis of Nonlinear Systems Using Conservative Approximation," *Hybrid Systems: Computation and Control*, Springer–Verlag, Berlin, 2003, pp. 20–35.
[51] Rankin, J., "Bifurcation Analysis of Nonlinear Ground Handling of Aircraft," Ph.D. Dissertation, Univ. of Bristol, Bristol, England, U.K., 2010.
[52] "Tailstrike and Runway Overrun, Melbourne Airport, Victoria," Australian Transportation Safety Bureau Accident Report AO-2009-012, Australian Capital Territory, Australia, 2009, http://www.atsb.gov.au/publications/investigation_reports/2009/aair/ao-2009-012.aspx [retrieved 10 Sept. 2014].
[53] Grauer, J. A., and Morelli, E. A., "A Generic Nonlinear Aerodynamic Model for Aircraft," *Proceedings of AIAA Atmospheric Flight Mechanics Conference*, AIAA Paper 2014-0542, 2014.

G. P. Brat
Associate Editor