Flight Safety Assessment and Management

by

Sweewarman Balachandran

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2016

Doctoral Committee:

Associate Professor Ella M. Atkins, Chair
Professor Dennis S. Bernstein
Professor Ilya V. Kolmanovsky
Assistant Professor Necmiye Ozay
David E. Smith, NASA Ames Research Center

2016

A C K N O W L E D G M E N T S

Grad school and this dissertation research have both been an incredible learning experience. I'm thankful to my Advisor Dr. Ella Atkins for giving me this amazing opportunity. Her guidance was key to making this journey smooth and an enjoyable experience. Ella's courses and research discussions have made me appreciate the importance of software architectures, algorithms, data-structures and abstractions in addition to the fundamental mathematics required to build real-world robotic systems. I enjoyed working in her lab on hardware and software for the rovers, the table-sats and the UAVs.

I'd also like to thank the members of my dissertation committee. Professor Ozay has helped me tremendously with the Verification chapters in this thesis. Dr. David Smith provided valuable insights with respect to the MDP formulations. Discussions with Professors Kolmanovsky and Bernstein helped integrating components of the bigger project my research was a part of.

I'm grateful to have met several wonderful people during my stay in Ann Arbor. I enjoyed discussing research, history and politics with my office-mates Kevin, Pedro, Cosme and Hossein. Pedro's enthusiasm, attention to detail and work ethic have always inspired me. Jhanani was a very supportive friend during her short stay at the University of Michigan. I wish them and the other amazing friends I've met at the University of Michigan the very best in their endeavors.

This work was funded by the National Aeronautics and Space Administration under Cooperative Agreement NNX12AM5. I once again thank Dr. David Smith and Dr. Christine Belcastro for their support and feedback.

I'd like to thank my parents for their love and support to pursue graduate education. Without their support I wouldn't have made it this far. I'm also grateful to uncle Balanayagam, relatives in Toronto, and uncle Saba and family for their support and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

FBW Fly-By-Wire

LOC Loss of Control

FMS Flight Management System

EAFMS Envelope Aware Flight Management System

FSAM Flight Safety Assessment and Management

TCAS Traffic Collision Avoidance System

GPWS Ground Proximity Warning System

ROPS Runway Overrun Protection System

MDP Markov Decision Process

CMDP Constrained Markov Decision Process

POMDP Partially Observable Markov Decision Process

LTL Linear Temporal Logic

DMM Deterministic Moore Machine

DFSA Deterministic Finite State Automaton

RTO Rejected Takeoff

AEO All Engines Operational

OEI One Engine Inoperative

NOMENCLATURE

Chapter 2

x	Longitudinal position on runway
y	Lateral position on runway/cross track error
h	Altitude
S	Finite set of states
Σ	Input alphabets
Λ	Output alphabets
$\mathcal{T}$	Transitions
$\mathcal{G}$	Output function
$\mathcal{A}_{lg}$	Longitudinal takeoff DMM
$\mathcal{A}_{lt}$	Lateral takeoff DMM
V	True airspeed
V_1	Takeoff decision speed
V_R	Takeoff rotation speed
V_{lof}	Liftoff speed
α, β, γ	Angle of attack, Side slip angle, Flight path angle
ϕ, θ, ψ	Roll, Pitch and Yaw angles
C_L, C_D	Lift, Drag coefficients
ρ, μ, S_{ref}	Atmospheric density, Friction coefficient, Planform area
T, W	Thrust, Weight
u, v, w	Velocities in the body frame
p, q, r	Angular rates
$\delta_a, \delta_e, \delta_r, \delta_t$	Aileron, elevator, rudder and throttle inputs
$\overline{p}$	Pilot control
$\overline{ap}$	Envelope Aware autopilot control
$\bar{\mathcal{V}}$	Airspeed partition
$\bar{\mathcal{Y}}$	Cross track error partition
$\bar{\Psi}$	Heading partition
$\bar{\Gamma}$	Lateral acceleration partition
$\mathcal{P}$	Takeoff abort flag

$\mathcal{R}$	Takeoff risk

Chapter 3

$\mathcal{S}$	Set of MDP States
$\mathcal{A}$	Set of MDP Actions
$\mathcal{T}$	Transition probabilities
$\mathcal{R}$	Reward function
$s \in \mathcal{S}$	Discrete MDP state
$a \in \mathcal{A}$	Discrete MDP action
λ	Discount factor
π	Policy
$\mathcal{V}$	Value/Utility
Q	State-action value/utility
F	MDP state feature/sub-feature
$\bar{Q}$	Airspeed-Position partition
$\bar{P}$	Pitch-Altitude partition
$\bar{L}$	Heading-Lateral position partition
$\bar{M}$	Control mode
$\bar{S}$	Mode select switch
h_{eng}	Engine status
TOGL	Toggle
NOOP	No operation
ρ	Occupational measure

Chapter 4

$\bar{V}$	Airspeed feature
$\bar{A}$	Angle of attack/sideslip feature
$\bar{\Theta}$	Dynamic pitch feature
$\bar{\Phi}$	Dynamic roll feature
$\bar{v}, \bar{\alpha}, \bar{\theta}, \bar{\phi}$	Airspeed, angle of attack/sidelip, dynamic pitch and dynamic roll state
$\bar{H}, \bar{T}$	Vertical speed and thrust features
$\bar{h}, \bar{t}$	Vertical speed and thrust state
$\bar{F}$	Flight plan feature
$\bar{f}$	Flight plan state
$\bar{I}$	Icing severity feature
$\bar{i}$	Icing severity state

$\bar{M}, \bar{S}$	Mode, mode selector features

Chapter 6

x, v, h	Longitudinal position, airspeed, altitude
q, θ, γ	Pitch angular rate, pitch attitude, flight path angle
T, W, ρ	Thrust, weight, atmospheric density
μ, g, S_{ref}	Rolling friction coefficient, acceleration due to gravity, planform area
C_{L_g}, C_{D_g}	Coefficient of lift and drag with ground effects
$\bar{q}, \mathring{q}$	Discrete state denoting a cell, discrete state denoting a facet of a cell
$\mathcal{H}, \bar{T}$	Observation map, abstraction function

ABSTRACT

Flight Safety Assessment and Management

by

Sweewarman Balachandran

Chair: Ella M. Atkins

This dissertation develops a Flight Safety Assessment and Management (FSAM) system to mitigate aircraft loss of control risk. FSAM enables switching between the pilot/nominal autopilot system and a complex flight control system that can potentially recover from high risk situations but can be hard to certify. FSAM monitors flight conditions for high risk situations and selects the appropriate control authority to prevent or recover from loss of control. The pilot/nominal autopilot system is overridden only when necessary to avoid loss of control. FSAM development is pursued using two approaches. First, finite state machines are manually prescribed to manage control mode switching. Constructing finite state machines for FSAM requires careful consideration of possible exception events, but provides a computationally-tractable and verifiable means of realizing FSAM. The second approach poses FSAM as an uncertain reasoning based decision theoretic problem using Markov Decision Processes (MDP), offering a less tedious knowledge engineering process at the cost of computational overhead. Traditional and constrained MDP formulations are presented. Sparse sampling approaches are also explored to obtain suboptimal solutions to FSAM MDPs. MDPs for takeoff and icing-related loss of control events are developed and evaluated. Finally, this dissertation applies verification techniques to ensure that finite state machine or MDP policies satisfy system requirements. Counterexamples obtained from verification techniques aid in FSAM refinement. Real world aviation accidents are used as case studies to evaluate FSAM formulations. This thesis contributes decision making and verification frameworks to realize flight safety assessment

and management capabilities. Novel flight envelopes and state abstractions are prescribed to aid decision making.

CHAPTER 1

Introduction

Advancements in digital computers have revolutionized the flight deck in commercial aircraft. Fly-By-Wire (FBW) technology has facilitated the use of advanced flight control, guidance and navigation algorithms which reliably maintain a trimmed flight condition and autonomously follow a desired flight plan. Triple redundancy further ensures the flight software can reliably fly the airplane in the event of any single component failure. Though these improvements have dramatically reduced the number of aviation accidents over time [1], Loss of Control (LOC) remains a primary contributing factor to commercial and general aviation accidents. Over 4000 fatalities have been attributed to LOC in the past decade alone [7, 8]. Fig 1.1 illustrates the accident rate per million departures of commercial transport category aircraft. According to the past two decades' statistics [1], an average rate of 0.3 accidents per million departures and 20 million departures per year result in 6 fatal commercial transport aircraft accidents per year. This overall crash figure is likely to worsen with the increase in the size of aircraft fleets and number of flight hours. Table 1.1 summarizes recent fatal commercial aviation accidents that occurred due to LOC.

LOC is a condition where an unusual attitude, rate of change of attitude, aerodynamic

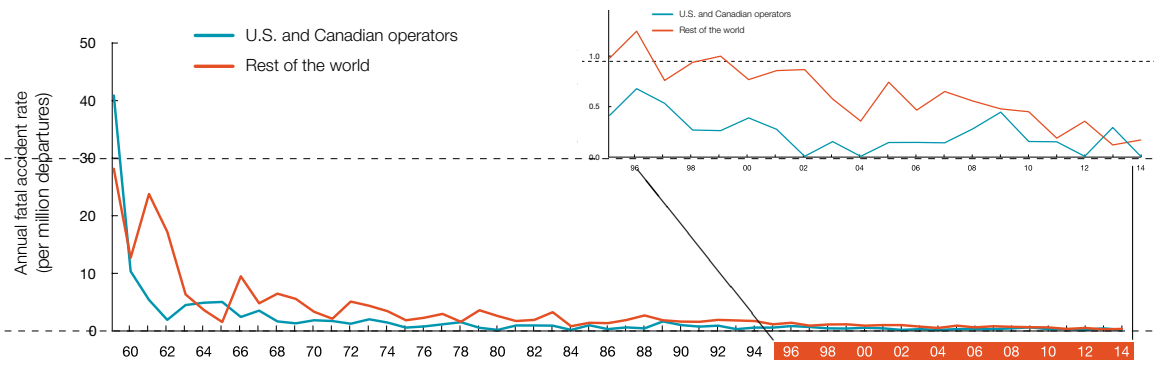

Figure 1.1: Statistical summary of commercial aircraft accident rates, 1959-2014 [1]

state or aircraft position violates normal operating constraints causing deviations outside the *normal flight envelope* [9]. LOC often results from a chain of events initiated by adverse environmental conditions and on-board 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.

Aircraft manufacturers use different strategies to deal with specific LOC situations. For example, flight envelope protection systems in Airbus commercial transport aircraft ensure safe flight envelopes cannot be violated by the flight crew [10]. The flight control computer overrides pilot commands that could result in flight envelope constraint violation. Boeing commercial transport aircraft use flight envelope limiting systems to alert the flight crew as envelope limits are reached or violated. However, the flight crew can override the envelope limits by using excessive force on the flight controls [11]. Systems such as Ground Proximity Warning System (GPWS) [12] and Runway Overrun Protection System (ROPS) [13] provide specific advisories to the flight crew to avoid certain LOC situations. Though these systems have significantly reduced aviation accident rates over the past decade, they disengage or even misinform the crew under off-nominal conditions such as sensor system failures, actuator failures and airframe icing. During such off-nominal conditions, flight crews must rely on past experience, air traffic controllers, and ground crews to make safety critical decisions, and an inappropriate decision can jeopardize safety of flight.

The current automation features available on-board airliners are not sufficiently "intelligent" to compute the risk level associated with a given flight condition. They do not understand and adapt to changes in the environment that can affect the aircraft dynamics. They also cannot assess the situational awareness level and intentions of the flight crew. More generally, current on-board automation features do not effectively utilize all available information to make decisions that can prevent or at least recover from LOC situations.

Consider the crash of Asiana Airlines Flight 214 [14]. The Boeing 777 was being manually flown by the flight crew during final approach into San Francisco International Airport (KSFO). The aircraft violated the safe operating envelope constraints with respect to both the airspeed and flight path angle. An autonomous flight envelope protection system would have recognized the risk associated with such flight conditions, overridden the flight crew and triggered a go-around flight mode thus mitigating the LOC risk. Similarly, for the accident cases summarized in Table 1.1 plus numerous others, LOC risk information could have enabled autonomy or a better informed crew to avert catastrophe.

Airline	Flight No.	Aircraft Type	Location	Date	Cause
Colgan Air	3407	DHC-8	Clarence Center,NY	02/01/09	Poor airspeed management, stall
Air France	447	A330	Atlantic Ocean	06/01/09	Blocked pitot probes, stall
Asiana Airlines	214	B777	San Francisco,CA	07/06/13	Poor approach speed management, stall
Malaysian Airlines	370	B777	Indian Ocean	03/08/14	Unknown
Air Asia	8501	A320	Java Sea	12/28/14	Rudder limiter failure, inappropriate pilot inputs, stall
Germanwings	9525	A320	Alpes-de-Haute-Provence, France	03/24/15	Suicide by pilot

Table 1.1: Fatal commercial aviation accidents from 2009-2015

1.1 Related Work

This section describes work related to loss of control prevention. Loss of control prevention architectures and sub-systems applying to specific LOC situations are discussed.

Belcastro et al [9] proposed the concept of Aircraft Integrated Resilient Safety Assurance and Fail-safe Enhancement (AIRSAFE). AIRSAFE takes a holistic approach to prevent LOC. The core features of AIRSAFE are vehicle health management, vehicle safety management, resilient control and flight crew interfaces. It proposes using online modeling and databases to predict impact of upset conditions on airplane dynamics, health assessment to continuously assess the state of aircraft, power plant and avionics health, support the flight crew by assisting with recoveries using resilient guidance and control module overrides. It emphasizes the use of efficient interfaces to improve situational awareness of the flight crew and enable efficient control mode transfer between the crew and resilient control laws. This thesis proposes the Envelope-Aware Flight Management System (EAFMS) which is conceptually similar to the ideals of AIRSAFE. This thesis focuses on the Flight Safety Assessment and Management (FSAM) module in EAFMS, a specific instantiation of the Vehicle Safety Management concept proposed in AIRSAFE.

Hovakimyan et al. [15–17] introduced the Integrated Reconfigurable Control for Vehicle Resilience (iReCoVeR). iReCoVeR, also conceptually similar to AIRSAFE and EAFMS,

integrates resilient flight control, flight envelope protection, LOC prediction, fault detection, and flight envelope determination. It focuses on providing L1 adaptive control augmentation to the nominal baseline controller. In contrast to the EAFMS, the iReCoVeR doesn't assist the flight crew with flight planning capabilities and lacks the decision-making capabilities to automatically switch to a resilient controller in cases more general than envelope protection.

The sandbox control architecture was introduced by Bak et al. [18] and proposes using an unverifiable or potentially difficult to verify control law in a sandbox environment. Control is transferred to a nominal/verifiable control law if the sandboxed control law behaves inappropriately. Control transfer is determined by a decision-making module that relies on manually defined rules based on reachability analysis of hybrid systems. The sandbox control architecture incorporates the control authority switching feature described by AIRSAFE. However, since the sandbox architecture was not intended for aerospace LOC applications, it doesn't facilitate the use of modules such as system identification and flight envelope estimation.

Borst et al. [19] introduced the aircraft Safety Augmentation System (SafAS). SafAS is an automated pilot support system that prevents aircraft from veering off course into hazards such as terrain, severe weather, and restricted airspace. One of SafASs objectives is to efficiently integrate aural, visual and haptic warnings issued to the flight crew before taking full control of the aircraft. Control transfer to a fail-safe controller is governed by user defined rules. Several tests with pilots in the loop were conducted to evaluate the performance and pilot acceptance of the SafAS system.

Gingras et al. [20] developed the Icing Contamination Envelope Protection (ICEPro) system. ICEPro helps identify degradations in airplane performance and flying qualities resulting from ice contamination by measuring aircraft state and control inputs, estimating aerodynamic parameters in real time and combining knowledge from databases. ICEPro also provides pilots with envelope limiting cues.

The Smart Icing System (SIS) was developed by Bragg et al. [21]. SIS senses the presence of icing, characterizes its effect on airplane performance, automatically activates the ice protection system and adapts the flight control system in response to the degraded flight envelopes due to icing. SIS also notifies the flight crew about the implications of a degraded flight envelope. The SIS was meant as an augmentation to existing icing related safety procedures. IcePro and SIS focus only on providing cues to the pilot and do not override the flight crew to mitigate risk.

Calise et al. [22, 23] proposed a non-linear adaptive control architecture that utilized neural networks and is capable of online learning of the network parameters and actuator

reallocation strategies. The proposed control scheme blends aerodynamic and propulsion actuation limits for safe operation. Napolitano et al. [24] proposed a flight control architecture that would be resilient to actuator and sensor failures. Resiliency is achieved by integrating sensor and actuator failure detection, identification and accommodation with the help of neural networks. Intelligent control schemes like those summarized above are suitable to achieve resiliency under LOC situations involving failed actuators or sensors. The prompt activation of these schemes can in turn be accomplished by FSAM.

Wensley et al. [25] proposed Software Implemented Fault Tolerance (SIFT), a highly reliable computer system for safety critical aircraft control applications. Unlike the above architectures and systems, SIFT reasons about critical failures at the computing layer and achieves fault tolerance by isolating the fault then reconfigures the system by replicating critical control tasks on different processing units.

The Flight Envelope Protection system on Airbus aircraft [10] and Flight Envelope Limiting systems on Boeing aircraft [11] are examples of subsystems designed to prevent LOC due to unusual attitudes. Flight Envelope Protection overrides the flight crew to prevent flight envelope violations while the Flight Envelope Limiting system makes it harder for the crew to violate envelope constraints. However, these systems are automatically disengaged under off-nominal conditions.

The Runway Overrun Prevention System (ROPS) was introduced by Airbus to warn flight crews about degraded landing conditions during final approach [13]. During approach, ROPS informs the flight crew via an intuitive interface whether to execute a go around or not. It also warns and assists the flight crew after touch down with the necessary actions to reduce the risk of a runway overrun.

The Automatic Ground Collision Avoidance System (Auto-GCAS) developed by the US Air Force Research Laboratories prevents Controlled Flight Into Terrain (CFIT) situations [26]. Using prior maps, a trajectory prediction routine and a collision avoidance routine the Auto-GCAS system overrides the pilot with a recovery autopilot to prevent imminent CFIT.

1.2 Problem Statement

Recent advancements in computation power, algorithms and sensors have made possible autonomous control, guidance and planning functions that can together mitigate LOC risk. However, except for the Airbus flight envelope protection and Auto-GCAS systems, LOC prevention systems devleoped rely on the flight crew to make task level and guidance decisions to mitigate LOC risk. The ability of a human pilot to comprehend available informa-

tion and make appropriate and timely decisions to mitigate risk is severely degraded under high stress conditions. Furthermore, the proprietary nature of systems designed to mitigate risk, lack of adequate training to understand system algorithms, and the limited human capacity to process high bandwidth data from the relevant sensor suite challenge flight crews in recognizing the need to configure (as needed) and promptly activate emerging LOC prevention modules.

The main focus of this dissertation is to formulate a decision-making system that continuously analyzes risk with respect to LOC and manages the transfer of control authority between the pilot/nominal automation and a sophisticated flight control system that can potentially prevent/recover from LOC situations but can be hard to certify. In this work, LOC is presumed to occur when either the aircraft violates flight envelope constraints or when the aircraft impacts off-runway terrain. Pilot/nominal automation override is performed only when necessary to avoid LOC. Though systems such as the Airbus flight envelope protection system and the Auto-GCAS system override the flight crew to mitigate imminent LOC risk, they apply only to specific loss of control situations and are not generalizable or applicable when multiple LOC risk factors are present.

The Envelope Aware Flight Management System (EAFMS) architecture (Figure 1.2) has been proposed to augment the conventional flight management system with system identification, envelope estimation, adaptive flight planning, adaptive control and control authority monitoring and management modules that collectively mitigate LOC risk. More generally, the EAFMS provides the intelligence required to make safe decisions in high risk LOC situations. The system identification module identifies changes to aircraft dynamics. Envelope Estimation uses information about the updated aircraft dynamics to estimate changes in aircraft flight envelopes. Information about updated aircraft dynamics and flight envelopes are used by the adaptive flight planner to construct [optimal] trajectories that satisfy degraded constraints. Adaptive control allows the new flight plan to be followed without violating degraded constraints. The decision-making module known as Flight Safety Assessment and Management (FSAM) is responsible for assessing the current risk conditions and selecting the appropriate control authority that will ensure LOC risk is mitigated.

This dissertation introduces the Flight Safety Assessment and Management (FSAM) system. FSAM is a "watchdog" designed to constantly monitor flight conditions for anomalies. If the aircraft encounters high risk flight conditions, FSAM warns the flight crew. If the flight crew fails to appropriately and rapidly respond, FSAM will override the current control authority with an Envelope Aware (EA) controller designed to handle the identified off-nominal condition. When LOC risk is mitigated, control is transferred back to the flight

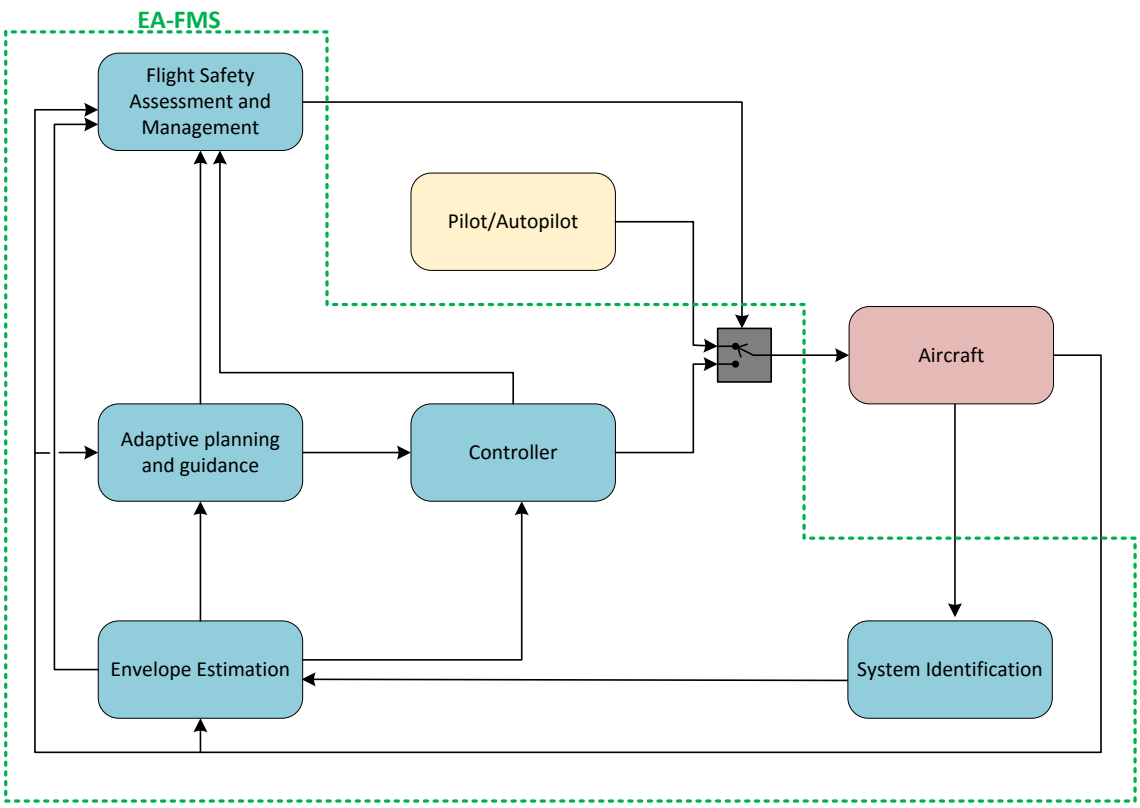

Figure 1.2: Envelope-Aware Flight Management System Architecture

crew.

1.3 Research Objectives

Flight Safety Assessment and Management is developed to be applicable to manned and unmanned aircraft. Two main design strategies for FSAM are investigated. First, control authority switching strategies are manually engineered using deterministic finite state machines. These deterministic graphical models are intuitive, easy to implement, and straightforward to verify thus certify using existing practices. Second, an uncertain reasoning approach is used to generate control authority switching rules. A Markov Decision Process (MDP) balances LOC risk with a penalty on FSAM intervention to assure FSAM overrides only as necessary. Finally, verification methods are developed to ensure FSAM interventions do not violate critical safety requirements.

1.4 Contributions

This dissertation offers a variety of contributions. The following subsections categorize contributions with respect to methods or algorithms and FSAM-related applications or case studies.

1.4.1 Methods and Algorithms

- This work presents a method to generate envelopes distinguishing safe and unsafe states with respect to translational, rotational and directional dynamics for an aircraft during takeoff. Knowledge of takeoff envelopes aids the construction of FSAM finite state machines and simplifies the state representation for FSAM MDPs.
- A holistic MDP formulation that can address several LOC contributing factors is provided in this work. The MDP formulation is used to generate policies used by FSAM to mitigate LOC risk during takeoff.
- An MDP formulation to address icing-related loss of control situations is developed in this work. The icing MDP formulation defines state features that utilize information from the envelope estimation and flight planning modules thus considering the changes to aircraft envelopes and the capabilities of the flight planner during decision-making.
- A sparse sampling approach is proposed to enable online or real-time decision-making for the MDP FSAM formulation. Trade-offs associated with parameter choices in the sparse sampling algorithm are also evaluated.
- A model checking approach to formally verify a takeoff FSAM system using Linear Temporal Logic (LTL) is presented in this work. Takeoff equations of motion are simplified to exploit existing reachability analysis algorithms and build discrete transition models that can be verified using off the shelf model checking tools.
- A method to check requirement violation using a cross-entropy Monte-Carlo search is established in this work and applied to FSAM.

1.4.2 Applications

- This work presents several real world aviation accident case studies that illustrate the applications and benefits of the FSAM finite state machines and MDP policies for the

takeoff phase of flight. FSAM formulations are developed to prevent lateral runway excursion, longitudinal runway overrun, and tail strike. Black box data available in NTSB accident dockets is utilized to show when and how FSAM with EAFMS could have intervened to prevent a series of real LOC incidents and accidents.

- FSAM is integrated with Envelope Estimation and Flight Planning capabilities of the EAFMS to address an in-flight rudder jam situation. FSAM in this case must react quickly to prevent an initial upset from which even an adaptive controller might not be able to recover.

1.5 Innovations

- This work introduces the Flight Safety Assessment and Management system. FSAM is an LOC "watchdog" that issues override decisions as needed, generalizing previous work to develop case-specific LOC prevention modules that function independently.
- Automation aids to warn the flight crew regarding specific takeoff loss of control situations are currently available. Application of automation override to avoid loss of control on takeoff is novel.
- The abstraction of takeoff envelopes to compactly represent states for the FSAM MDP formulation is necessary for efficiency and is new.
- The general MDP formulation presented in this work incorporates features that enable reasoning over aircraft health, flight crew and environment characteristics in addition to aircraft dynamics. So far, aircraft safety systems described in the literature have limited their focus to features related to aircraft dynamics.
- The Constrained Markov Decision Process (CMDP) formulation presented in this work eliminates the need to manually tune reward function weight parameters to obtain MDP policies that reliably meet system constraints as well as minimizing EAFMS intervention.
- In this work, a finite state machine defining FSAM switching logic is composed with a discrete transition system representing aircraft dynamics under different control laws. This composition yields an over-approximation of reachable states and is novel.

1.6 Dissertation Outline

Chapter 1 concludes below with a list of publications. Chapter 2 introduces the flight safety assessment and management system for LOC risk mitigation. Developing a system to comprehensively mitigate LOC risk factors that might occur from takeoff through landing is computationally intractable. Thus, the problem of flight safety assessment and management is decomposed based on flight phases. Chapter 2 then develops an FSAM formulation for takeoff. Control authority switching decisions are manually engineered and encoded using deterministic graphical models or state machines.

Chapter 3 presents a decision-theoretic formulation of FSAM. An MDP is used to construct LOC mitigation strategies automatically. Emphasis is placed on developing MDP formulations for FSAM. This chapter explores the use of state-space abstractions to mitigate the computational overhead associated with large scale MDP formulations. An MDP formulation that can handle constraints and reduce parameter tuning overhead is presented.

Chapter 4 develops an MDP formulation to address loss of control scenarios related to in-flight icing. The MDP states are parameterized by key envelope parameters such as stall speed, maximum airspeed and maximum/minimum attitude angles. The MDP state representation also captures the risk associated with a given flight plan with respect to ice accumulation.

Chapter 5 uses a sparse sampling algorithm as an online MDP solver to mitigate risk due to loss of control. Results illustrating the tradeoffs associated with parameter choices for the sparse sampling algorithm are also presented.

Chapter 6 and 7 consider the verification of the FSAM system. Chapter 6 presents a framework to verify the finite state machine formulations against FAR requirements expressed in Linear Temporal Logic. Chapter 7 explores methods to quantitatively analyze the robustness of the FSAM finite state machines and MDP policies to critical safety requirements. Specifically, a cross-entropy based method is used to determine the robustness of the FSAM formulation.

Chapter 8 provides conclusions and discusses future work. Fig 1.3 graphically illustrates the organization of this dissertation.

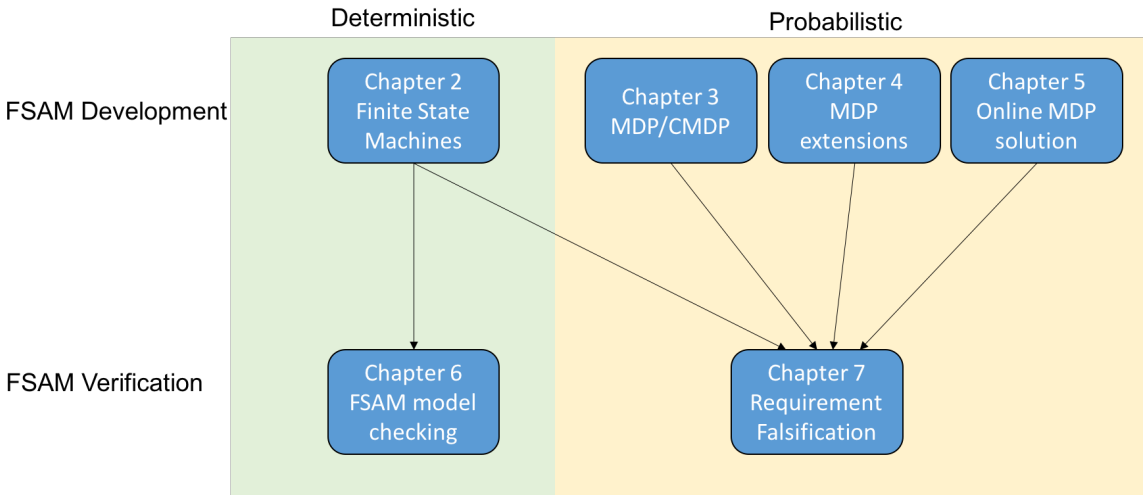

Figure 1.3: Thesis organization

1.7 Publications

Conference

- Balachandran, S. and Atkins, E. M. "Flight Safety Assessment and Management during Takeoff" AIAA Infotech@Aerospace conference, Boston, MA, 2013. DOI: 10.2514/6.2013-4805
- 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, National Harbor, MD, 2014. DOI: 10.2514/6.2014-0785
- Balachandran, S. and Atkins, E. M. "A Constrained Markov Decision Process Framework for Flight Safety Assessment and Management" AIAA Infotech@Aerospace conference, Kissimmee, FL, 2015. DOI: 10.2514/6.2015-0115 (won Intelligent Systems Best Student Paper Award)
- Balachandran, S. and Atkins, E. M. "Flight Safety Assessment and Management to Prevent Loss of Control Due to In-Flight Icing", AIAA Guidance, Navigation and Control Conference, San Diego, CA, 2016. DOI: 10.2514/6.2016-0094
- Balachandran, S. and Atkins, E. M. "An Autonomous Override System to Prevent Airborne Loss of Control", in proceedings of Innovative Applications in Artificial Intelligence (IAAI), Phoenix, AZ, 2016

Journal

- 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 (2015), pp. 599-615. DOI: 10.2514/1.I010350
- Balachandran, S., Ozay, N. and Atkins, E. M. "Verification Guided Refinement of a Flight Safety Assessment and Management System for Takeoff", Journal of Aerospace Information Systems (Accepted for publication).
- Balachandran, S. and Atkins, E. M. "A Markov Decision Process Framework for Flight Safety Assessment and Management", Journal of Guidance, Control and Dynamics, 2016 (Revision submitted).
- Donato, P., Balachandran S., McDonough, K., Atkins, E. M. and Kolmanovsky, I. Envelope-Aware Flight Management for Loss of Control given Rudder Jam, Journal of Guidance, Control and Dynamics, 2016 (Accepted for publication).

CHAPTER 2

Flight Safety Assessment and Management using Manually Constructed Finite State Machines

2.1 Introduction

Flight Safety Assessment and Management (FSAM) is the decision-making component of the Envelope-Aware Flight Management System (EAFMS). FSAM is designed to constantly monitor flight conditions for anomalies and to assess risks associated with the current flight conditions. FSAM warns the flight crew when risk is present, and if the flight crew does not respond with appropriate control actions in time to assure recovery, FSAM overrides with EAFMS until LOC risk is mitigated. FSAM is effectively a "watchdog" system with LOC avoidance override capabilities such as flight envelope protection [10, 11] in a more general context.

This chapter focuses on formulating FSAM using graphical models. Specifically, a hierarchical formulation is specified by a timed automaton framework. The logic governing the control authority switching imposed by FSAM is represented as a Deterministic Moore Machine (DMM). Timed automata and DMMs are finite state machine formulations defined by states, actions and transitions triggered by timers/events. Thus, these tools form a suitable framework to model a control mode override strategy for FSAM.

Developing an FSAM system to address all possible LOC events across the entire flight regime can become cumbersome given that one has to reason over possible input sequences

Figure 2.1: Flight Safety Assessment and Management

that could result in high risk conditions. To maintain tractability of the formulation and to enhance readability, a decomposition based on the phases of flight is considered. Deterministic logic models applicable to a suite of LOC precursor scenarios are developed. The phases of flight ultimately to be considered in logic models include takeoff, climb, cruise, loiter, approach, and landing. These will be further divided to manage complexity of each machine. We seek readability of the state machine logic not just for software validation purposes but also to enhance flight crew understanding of the underlying system functionality.

The rest of this chapter is organized as follows. Section 2.2 discusses the relevant tools that will be used in this chapter to construct FSAM. Section 2.3 illustrates the hierarchical decomposition of FSAM. This chapter focuses on the FSAM formulation for takeoff and hence, Section 2.4 surveys literature related to takeoff safety. Section 2.5 provides a discussion of takeoff and factors that contribute to LOC during takeoff. Section 2.6 discusses the development of flight envelopes to efficiently and intuitively identify risk during takeoff. Section 2.7 presents the FSAM formulation used to avoid LOC during takeoff. Section 2.8 presents case studies illustrating the application of FSAM to a real world accident scenario while Section 2.9 discusses results and their implications. Section 2.10 presents conclusions.

2.2 Background

A Deterministic Finite State Automaton (DFSA) or simply Finite State Automaton or Finite State Machine is an abstract mathematical model of computation. The machine consists of a finite number of states. It can only exist in one state at any given instant of time. The DFSA starts from an initial state and receives as input, a sequence or string of input symbols taken from a specified alphabet. These strings can have different interpretations based on the application of the DFSA. Each alphabet symbol indicates a unique transition from the current state to the next state, defining the discrete dynamics of the system. Each input string is either accepted or rejected. The set of all accepted input strings defines the language of the DFSA.

2.2.1 Timed Automaton

The timed automaton [27] is an extension of the DFSA for which transitions between states are governed by timers along with alphabet symbols. These timers serve as guards on the transitions to ensure system timing constraints are satisfied. Timed automata can be repre-

sented with a timed transition table [27]. A time transition table is a tuple $(\Sigma, S, S_0, \bar{C}, \bar{E})$ where

- Σ is a finite alphabet.
- S is a finite set of states.
- $S_0 \subseteq S$ is a set of initial states.
- $\bar{C}$ is a finite set of clocks.
- $\bar{E} \subseteq S \times S \times \Sigma \times 2^{\bar{C}} \times \Phi(\bar{C})$ gives the set of transitions. An edge $< s, s', a, \lambda, \delta >$ represents a transition from state s to state s' on input symbol a. The set $\lambda \subseteq \bar{C}$ gives the clocks to be reset with the transition, and $\delta \in \Phi(\bar{C})$ is a clock constraint over $\bar{C}$.

2.2.2 Deterministic Moore Machines

Deterministic Moore Machines (DMM) [28–30] are also extensions of the DFSA where each state is associated with an output or control action. DMMs are formally defined by the tuple $(\mathcal{S}, \mathcal{S}_0, \Sigma, \Lambda, \mathcal{T}, \mathcal{G})$, where

- $\mathcal{S}$ represents a discrete set of states.
- $\mathcal{S}_0 \subset \mathcal{S}$ represents an initial state.
- Σ is a finite set of input alphabet.
- Λ is a finite set of output alphabet.
- $\mathcal{T} \subseteq \mathcal{S} \times \Sigma \times \mathcal{S}$ represents the set of state transitions.
- $\mathcal{G} : \mathcal{S} \times \Lambda$ is the output function mapping each state to a unique output character (control action).

2.3 Hierarchical Decomposition of Flight Safety Assessment and Management

The motivation to hierarchically decompose FSAM is twofold. First, a decomposition aims to break down FSAMs decision-making process according to the phase of flight, taking into account the various changes in flight plan that could occur during a typical flight leg.

This helps to reduce the complexity of the logic modules. Second, it introduces FSAMs architecture intuitively to both the Pilot and Engineering community. Fig 2.2 shows a timed automaton that depicts the progression of flight phases from takeoff through landing. Each state represents a particular phase of flight. Marked in grey are the nominal phases of flight. A nominal progression from takeoff to landing does not deviate from the original flight plan. Marked in yellow are the off-nominal states. The off-nominal states arise due to various reasons such as bad weather, air traffic conflicts and emergencies on board. The symbols used by the automaton are described in Table 2.1. A timer (t) is initiated at the beginning of the takeoff phase to keep track of the elapsed time of flight. From takeoff to landing, the aircraft is in transit from one phase to another. In each state, the automaton receives an alphabet symbol that reminds it about the current phase of flight (i.e M_{climb} , M_{cruise} , $M_{descent}$, etc...). This can be viewed as a reflexive transition which is implicit in each state of the automaton and thus it is not explicitly marked in the state diagram in Fig 2.2.

Each state in Fig 2.2 has its own subset of deterministic finite state machines with goals specific to the corresponding phase of flight. A transition from the current state to the next state (i.e. the next phase of flight) occurs when the aircraft arrives at the next phase of flight before/at the estimated time of arrival (e.g: M_{climb} , $t \leq ETA_1$). The ETAs at each phase are calculated prior to departure as per the original flight plan then updated en-route according to changes in flight conditions (e.g. wind, detours due to weather, air traffic). An off-nominal condition is flagged by the symbol O. Few examples of off-nominal conditions are: failing to reach the next phase before/at the estimated time of arrival (i.e slow with respect to original flight plan, e.g: $M_{takeoff}$, $t > ETA_1$), veering off course with respect to the original flight plan and discrepancies in fuel available vs fuel required for the remainder of the flight as per regulations.

On receiving the symbol O, the automaton transitions to the corresponding *alert* state. This is a warning state where the crew is made aware of the off-nominal conditions. If the flight crew doesn't take the appropriate actions to mitigate risk, control will be transferred to the envelope-aware control law to ensure LOC risk is mitigated. The symbol O' marks the return to the nominal flight conditions. Under certain circumstances, the flight crew may be required to follow an alternate flight plan. For example, the flight crew may be asked by air traffic controllers to climb to a different flight level to avoid other air traffic. Such changes are identified by comparing the original flight plan with the flight plan changes made to the navigation computers of the Flight Management System (FMS) en-route and flagged by the symbol C. It is worth noting that the state machine is able to distinguish between intentional (flagged by C) and unintentional (flagged by O) deviations from the

Symbols	Description
ETA_1	Estimated time of arrival at climb phase
ETA_2	Estimated time of arrival at cruise phase
ETA_3	Estimated time of arrival at descent phase
ETA_4	Estimated time of arrival at approach phase
C	Change in flight plan / emergency flight plan
M_{phase}	Current phase of flight
O	Flight plan fault (delay in flight leg/off course/fuel warnings ...)
O'	Return to nominal flight plan
RTO	Rejected takeoff
t	Elapsed time

Table 2.1: Symbols used in top-level timed automaton

nominal flight plan. The top level automaton in Fig 2.2 tracks elapsed time of flight, inertial position, and fuel consumption. The crew is warned if any discrepancies arise. This in turn helps avoid inadvertent excursions from the original flight plan due to factors such as lack of situational awareness, distraction in the cockpit, and mode confusion. This bookkeeping strategy is consistent with current practices where pilots use flight plan sheets to manage the flight plan en-route by manually recording elapsed time of flight versus estimated time of flight to keep track of the inertial position and fuel consumption. [1]

This chapter focuses on developing an FSAM formulation for the takeoff phase. For takeoff, LOC translates to a situation in which the aircraft veers off the side of the runway, overshoots the runway, or leaves the ground in a condition (e.g., insufficient speed/inappropriate rotation attitude) that introduces substantial risk in the subsequent departure climb. This work contributes a deterministic decision-making framework that addresses the above LOC factors in a holistic manner with an approach that can be certified using existing processes in DO-178B or DO-178C [31]. The DMM [28] for takeoff FSAM characterizes the evolution of aircraft states to support safe takeoff decisions with FSAM warning or override to avoid LOC risk. The DMMs are formulated based on analysis of aviation accident surveys, accident/incident reports, flight data obtained from the NTSB accident database, aircraft operating manuals, pilot handbooks, checklist procedures and flight control laws from the literature [10, 11, 32–34]. This chapter introduces envelopes for the takeoff phase that enable the identification of safe and unsafe states. These envelopes guide the design of the DMMs.

[1] On modern commercial airliners, this bookkeeping is done by the on-board flight management systems. However, general aviation pilots still use navigation logs to record progress of each flight plan.

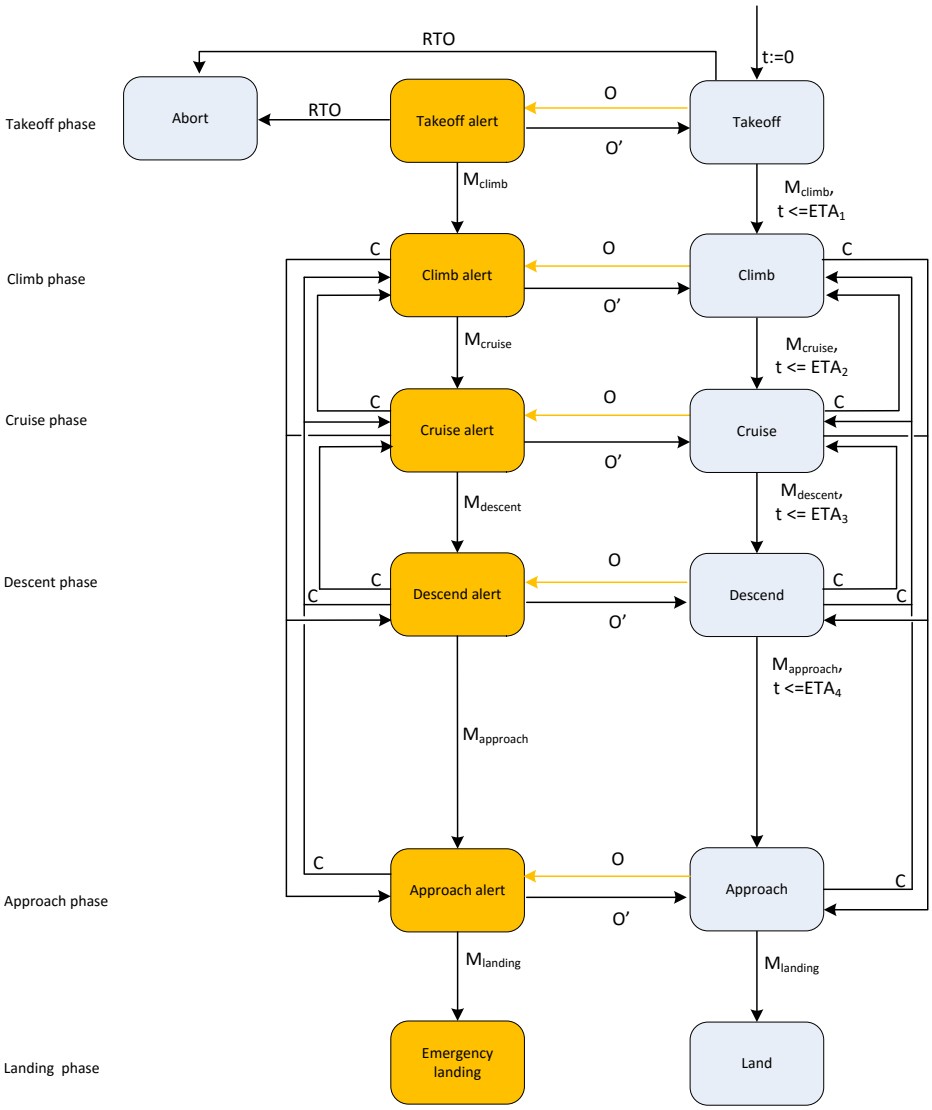

Figure 2.2: Top level timed automaton

2.4 Related Work: Takeoff Safety

Despite the safety-critical nature of takeoff, limited previous work has been devoted towards LOC risk mitigation specifically for takeoff. Srivatsan et al [35], Milligan et al [36] and Zammit-Mangion et al [37] proposed systems that constantly monitor takeoff ground roll performance parameters and detect anomalies by comparing the current performance with a precomputed nominal performance profile. Verspay et al [38] evaluated the merits of various types of Takeoff Performance Monitoring Systems (TOPM) as well as characteristics of a TOPM that improved pilot decision-making during takeoff. It was found that

a system with the ability to predict continued takeoff status and stopping performance has the potential to improve safety. Inagaki et al [39] investigated automating GO/NO-GO decisions using a situation adaptive autonomy framework [40]. These publications focus only on aiding the flight crew in making safe GO/NO-GO decisions during takeoff and do not consider other LOC risks such as loss of directional control or inappropriate rotation. Furthermore, these publications only provide cues to the flight crew and rely on the flight crew to make risk mitigating decisions.

2.5 Takeoff

The causal factors for takeoff-related accidents were first examined to aid the construction of an FSAM system for the takeoff phase of flight. Ninety-seven Rejected Takeoff (RTO) runway overrun accidents and incidents have been reported from 1960-2000 resulting in more than 400 fatalities [1,7,8]. Takeoff accident causal factors are summarized in Fig 2.4.

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 [34]. In a commercial transport aircraft, a typical takeoff ground roll lasts 20 - 35 seconds. The Federal Aviation Regulations (FARs) define several airspeed checkpoints called V-speeds [34, 41] to guide the flight crew in making appropriate decisions during takeoff. The most important V-speed is V_1, the decision speed by which the flight crew must decide to continue or reject a takeoff, i.e., make a GO/NO-GO decision with sufficient remaining runway to safely reject the takeoff. The flight crew may need to reject a takeoff due to several factors such as engine failure(s), tire burst(s) and runway incursion. A rejected takeoff initiated after V_1 will leave insufficient runway length to stop safely. Rotation initiated before the appropriate V-speed can result in an early departure stall [32]. Figure 2.3 graphically represents takeoff V-speeds. A listing of V-speeds is provided in Section 2.7 (Table 2.2).

The goal of FSAM is to identify LOC risk and assure its mitigation. To simplify the construction of FSAM, the following assumptions are made:

- The aircraft is cleared for takeoff and faces no risk due to obstacles or other aircraft.
- The aircraft flight envelopes are nominal and remain unchanged.
- There are no electro-mechanical or structural failures.
- All systems and sub-systems are functioning according to specification.

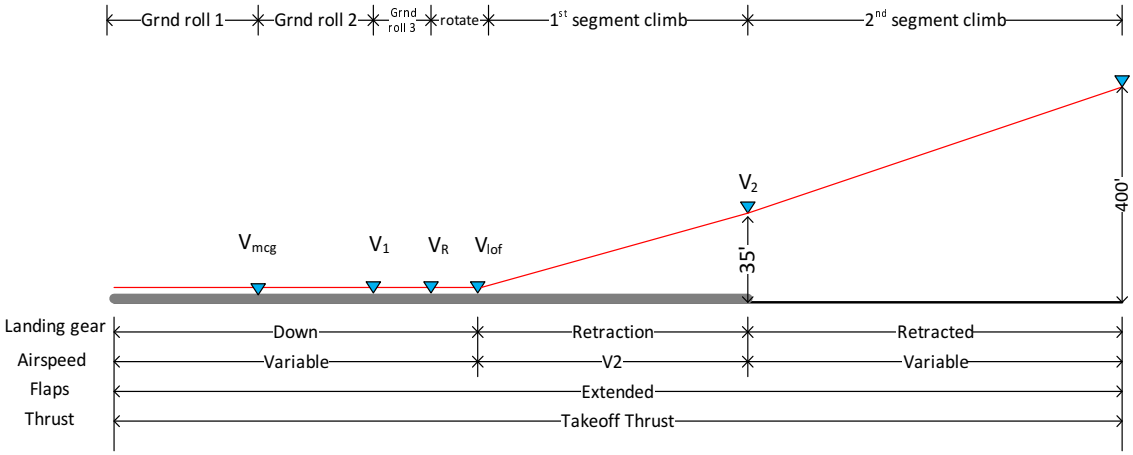

Figure 2.3: Takeoff phase of flight

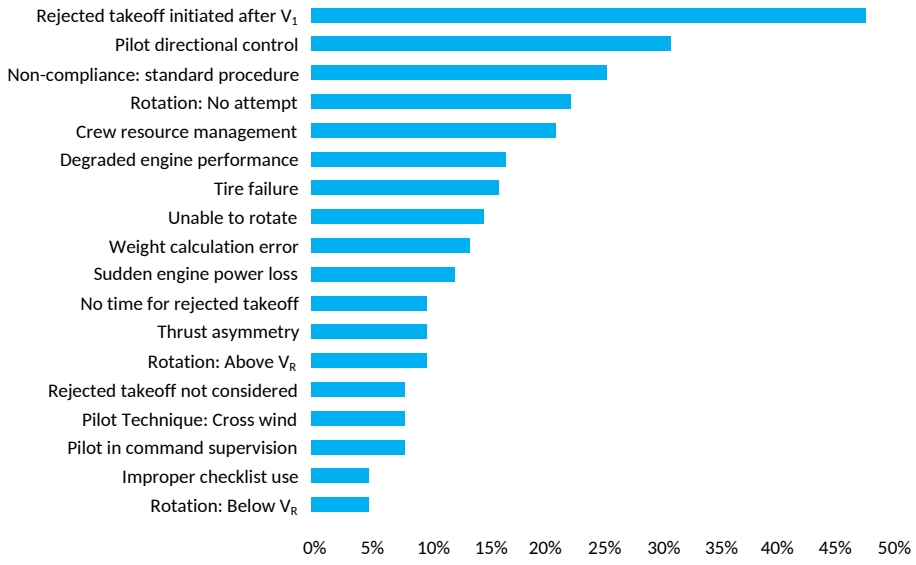

Figure 2.4: LOC contributing factors for takeoff [2]

- The EAFMS has sufficient situational awareness to recover for takeoff related LOC hazards.

These assumptions would be relaxed in a comprehensive takeoff FSAM beyond the scope of this work.

2.6 Takeoff Flight Envelopes

To achieve an effective FSAM capability, safe flight envelopes for takeoff must first be defined. For a given takeoff configuration (weight, thrust and flap/slat settings), safe versus unsafe partitions for translational, rotational and directional dynamics can be identified based on aircraft equations of motion for takeoff. Below, we present envelopes for the takeoff phase that are essential to prevent factors such as improper rejected takeoff decisions, degraded acceleration performance (due to reduced engine performance or other factors such as weight calculation errors), tail strikes, poor rotation procedures and directional control issues. Appendices A-C provide details on the non-linear aircraft takeoff dynamics model and simplifications performed to construct the translational, rotational and directional envelopes.

2.6.1 Translational dynamics

The maximum airspeed at which a rejected takeoff must be initiated to stop safely within the available runway space can be estimated as illustrated in Appendix C.1. The phase portrait in Fig 2.5 illustrates how airspeed (V) and longitudinal position (x) evolve after a rejected takeoff is initiated. The solid blue curve in Fig 2.5 defines the partition of the $V-x$ space for which a rejected takeoff will enable the aircraft to stop safely at or before the end of the runway.

Analogously, one can estimate the minimum airspeed beyond which a One Engine Inoperative (OEI) takeoff can be safely continued (see Fig 2.6). All trajectories to the left of this envelope will overshoot the runway before attaining airspeed V_2.

Fig 2.7 combines the constraints in Fig 2.5 and Fig 2.6 to partition safe regions in $V-x$ space with respect to RTO and OEI conditions. The intersection of the two curves represents V-speed V_1.

When operating with All Engines Operational (AEO), the aircraft must always stay in an envelope where at least one safe action can be executed. Fig 2.8 defines a minimum thrust boundary assuming AEO to avoid the zone that is unsafe with respect to RTO and

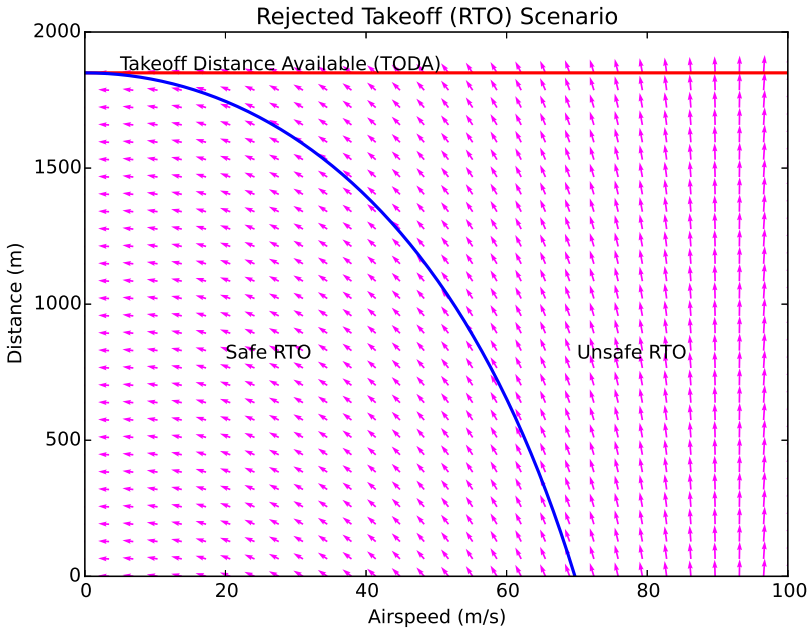

Figure 2.5: Rejected Takeoff (RTO) Envelope

OEI. If the aircraft deviates outside this minimum thrust boundary under the AEO condition before V_1, takeoff must be rejected.

2.6.2 Rotational dynamics

Fig 2.9 illustrates the constraint on pitch during rotation. Over-rotation leads to tail strikes, imposing a maximum pitch attitude constraint. The set of all initial conditions that can result in a tail strike scenario is estimated via a reachable set analysis [42] to identify safe versus unsafe operating regions (See Appendix C.1.1). For an aircraft with a tail strike constraint of 11° and other characteristics listed in Table A.1, Fig 2.10 illustrates the safe versus unsafe sets at different airspeeds. Note that the unsafe set becomes smaller at higher airspeeds. This is because the elevator effectiveness increases with airspeed and is effective in reducing the pitch rate quickly to prevent a tail strike.

2.6.3 Directional dynamics

Lateral runway excursions due to poor directional control can be managed by enforcing constraints on lateral motion. Fig 2.11 illustrates safety constraints on cross track position (y) and heading (ψ). Bounds $|\, y \,| \leq |\, y_1 \,|$ and $|\, \psi \,| \leq |\, \psi_1 \,|$ represent transitions to moderate risk states for FSAM, while either $|\, y \,| > |\, y_2 \,|$ or $|\, \psi \,| > |\, \psi_2 \,|$ represent unacceptable lateral

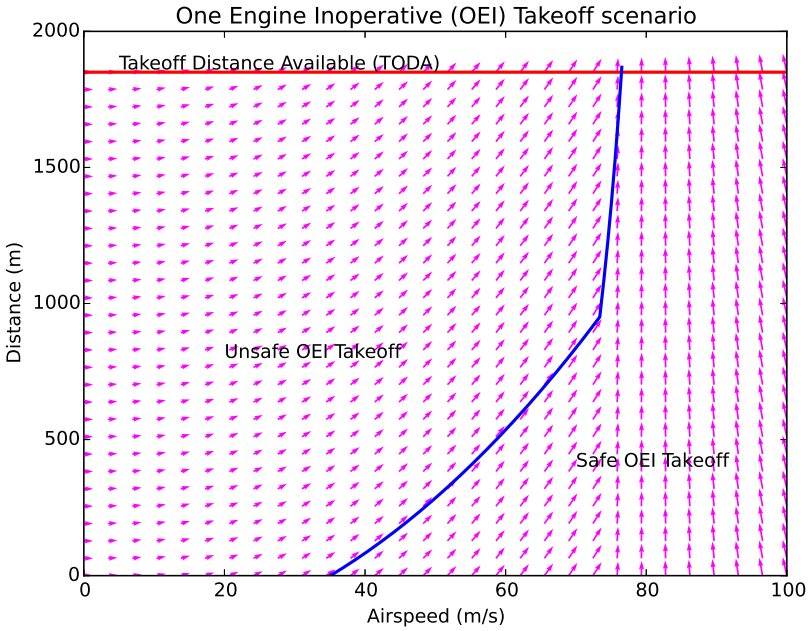

Figure 2.6: Takeoff with one engine inoperative

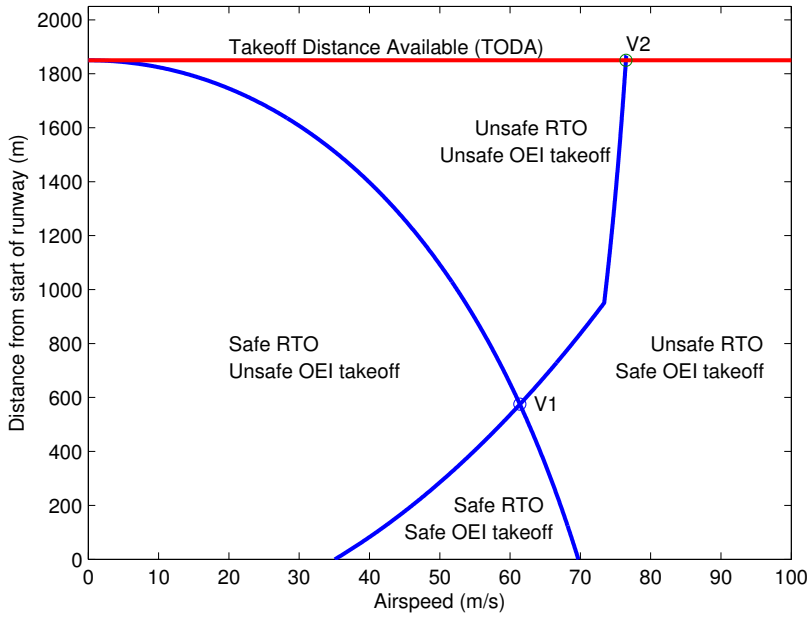

Figure 2.7: Safe and Unsafe regions of takeoff flight envelopes

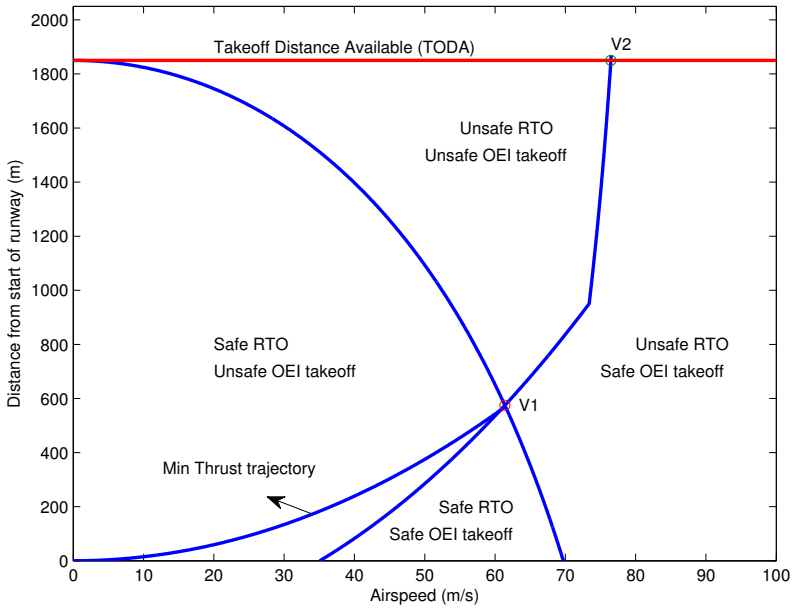

Figure 2.8: RTO, OEI and AEO envelopes

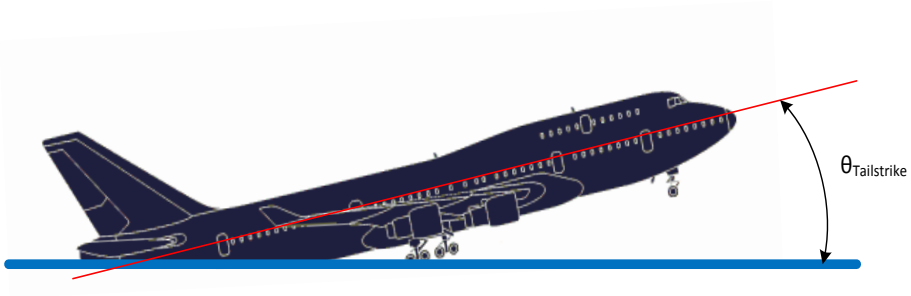

Figure 2.9: Tail strike constraints

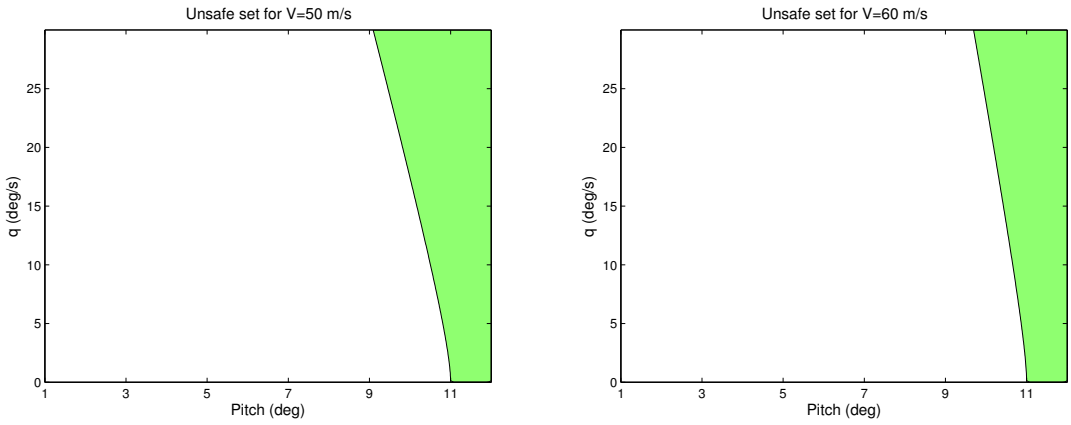

Figure 2.10: Unsafe sets at different airspeeds

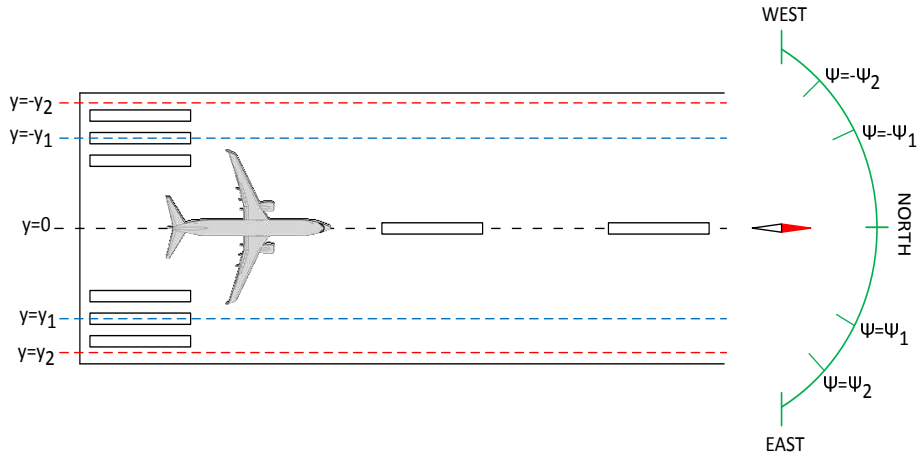

Figure 2.11: Lateral takeoff constraints to avoid runway excursion

traversal condition. Similar to the rotational dynamics, the set of all initial conditions that lead to constraint violation can be estimated to identify the safe versus unsafe operating regions. Fig 2.12 illustrates a critical surface below which, all initial conditions can result in violation of $|\, y \,| \leq |\, y_2 \,|$ where $y_2 = 25m$. Note that Fig 2.12 only illustrates the surface with respect to the constraint $y \geq -y_2$. A similar surface can be defined for $y \leq y_2$ (See Appendix C.1.2).

The following section provides DMM formulations that utilize the knowledge of safe versus unsafe states illustrated in Fig 2.8-2.12 to trigger transitions to the Envelope-Aware controller and mitigate risk.

2.7 DMM formulation of FSAM

The decision-making logic to prevent takeoff LOC is modeled as a Deterministic Moore Machine (DMM). DMMs are modular and composable [43, 44] and use of a deterministic specification for FSAM will facilitate its verification and certification using well established tools in model checking, a topic studied in Chapter 5. The DMM formulation for takeoff is decomposed into a longitudinal DMM and lateral DMM mirroring how control laws are typically designed for fixed wing aircraft. The two DMM formulations are specified below.

2.7.1 Longitudinal Deterministic Moore Machine

The longitudinal takeoff FSAM DMM identifies LOC risk with respect to the longitudinal aircraft state variables. Takeoff stages are correlated with V-speeds as shown in Fig 2.3. We represent the longitudinal Moore machine ($\mathcal{A}_{lg}$) by the tuple $(\mathcal{S}_{lg}, \mathcal{S}_{lg0}, \Sigma_{lg}, \Lambda_{lg}, \mathcal{T}_{lg}, \mathcal{G}_{lg})$

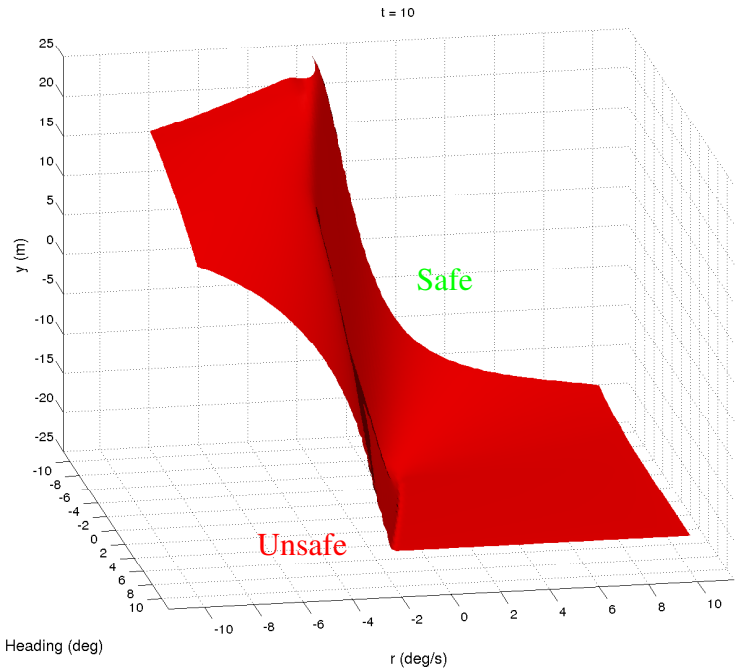

Figure 2.12: Surface representing the boundary of the unsafe set

where

$$\mathcal{S}_{lg} = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}\} \tag{2.1}$$

$$\mathcal{S}_{lg0} = \{s_1\} \tag{2.2}$$

$$\Sigma_{lg} = \{V_{mcg}, V_1, V_R, V_{lof}, V_2, V_{fp}, T_{idle}, T_{max}, c, c', e, e', f, \theta, \bar{\theta}\} \tag{2.3}$$

$$\Lambda_{lg} = \{P, EA\} \tag{2.4}$$

$$\mathcal{G}_{lg} = \begin{cases} P & if\ s_i \in \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_{14}\} \\ EA & otherwise \end{cases} \tag{2.5}$$

Transitions $\mathcal{T}_{lg}$ represent edges in a directed state transition graph (Fig 2.13). The definition of each alphabet symbol in the set Σ_{lg} is listed in Table 2.2. Table 2.3 provides descriptions of each state. A state $s \in \mathcal{S}_{lg}$ is defined by the triplet $[\bar{\mathcal{V}}, \mathcal{P}, \mathcal{R}]$. $\bar{\mathcal{V}}$ represents an airspeed range with values shown in Eqn (2.6)-(2.14). Here $\mathcal{P} \in \{0, 1\}$ is a flag set to true when continuing the takeoff is no longer safe because of inappropriate aircraft configuration. $\mathcal{R} \in \{\varepsilon, low, med, high\}$ represents the risk level associated with the current state, where ε denotes a zero risk state.

$$\bar{\mathcal{V}} \in \{\bar{v}_i\},\ i = 1,..,8 \tag{2.6}$$

$$\bar{v}_1 = \{V \in \mathbb{R} \mid V = 0\} \tag{2.7}$$

$$\bar{v}_2 = \{V \in \mathbb{R} \mid 0 < V \leq V_{mcg}\} \tag{2.8}$$

$$\bar{v}_3 = \{V \in \mathbb{R} \mid V_{mcg} < V \leq V_1\} \tag{2.9}$$

$$\bar{v}_4 = \{V \in \mathbb{R} \mid V_1 < V \leq V_R\} \tag{2.10}$$

$$\bar{v}_5 = \{V \in \mathbb{R} \mid V_R < V \leq V_{lof}\} \tag{2.11}$$

$$\bar{v}_6 = \{V \in \mathbb{R} \mid V_{lof} < V \leq V_2\} \tag{2.12}$$

$$\bar{v}_7 = \{V \in \mathbb{R} \mid V_2 < V \leq V_{fp}\} \tag{2.13}$$

$$\bar{v}_8 = \{V \in \mathbb{R} \mid V > V_{fp}\} \tag{2.14}$$

Each state $s \in \mathcal{S}$ is mapped to an output by function $\mathcal{G}_{lg}$. P represents the "pilot in control" command and EA represents the "envelope aware autopilot" command output. The output of each state is indicated on the lower half of each state depicted in Fig 2.13. This work assumes the envelope aware controller has sufficient situational awareness to recover from the LOC triggers/hazards.

As shown in Fig 2.13, the aircraft starts from an initial state of rest (s_1) at $(x,y) = (0,0)$. If the aircraft is configured for takeoff (flagged by c) and takeoff thrust is established (flagged by T_{max}), 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 2.13 represents the nominal V-speed sequence. The additional states represent off-nominal conditions with LOC risk. If the aircraft is inappropriately configured for takeoff ($c' \wedge T_{max}$), the DMM enters a configuration warning 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 ($V_{mcg} < V \leq V_1$), if the aircraft has inadequate acceleration (f), FSAM rejects the takeoff to prevent entry into the zone that is unsafe with respect to RTO and OEI (see Fig 2.8). 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 over-speed become active. Pushing the aircraft to the stall boundary during the climb (s_6,s_7) results in override with envelope aware (EA) control (s_{11},s_{12}) analogous to stall or envelope protection modules found on existing Airbus aircraft. FSAM reverts control to the flight crew after the aircraft is stabilized on climbout.

The DMM models presented here are a sub-component of the FSAM system covering

Table 2.2: 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
d	Crossing 1st directional threshold
d'	Crossing 2nd directional threshold
e	Envelope protection de-activated
e'	Envelope protection activated
f	Inadequate acceleration performance
o'	Stall
θ	Positive pitch attitude
$\bar{\theta}$	Maximum allowable pitch attitude reached during rotation
θ'	Safe rotation attitude

all phases of flight. Consequently, after takeoff, FSAM switches to a climb DMM that is beyond the scope of this work.

2.7.2 Lateral Deterministic Moore Machine

The lateral FSAM DMM ensures directional control is sufficient to prevent crosstrack runway excursions. Directional control loss can result from high crosswinds or gusty winds, engine thrust asymmetry, and inappropriate rudder inputs.

The lateral DMM $\mathcal{A}_{lt}$ is represented by the tuple $(\mathcal{S}_{lt}, \mathcal{S}_{lt0}, \Sigma_{lt}, \Lambda_{lt}, \mathcal{T}_{lt}, \mathcal{G}_{lt})$, where:

$$S_{lt} = \{s'_1, s'_2, s'_3, s'_4, s'_5, s'_6, s'_7, s'_8, s'_9, s'_{10}, s'_{11}, s'_{12}, s'_{13}, s'_{14}, s'_{15}\} \tag{2.15}$$

$$S_{lt0} = \{s'_1\} \tag{2.16}$$

$$\Sigma_{lt} = \{V_{mcg}, V_1, V_R, V_{lof}, V_2, V_{fp}, T_{idle}, T_{max}, c, c', e, e', d', \bar{d}\} \tag{2.17}$$

$$\Lambda_{lt} = \{P, EA\} \tag{2.18}$$

$$\mathcal{G}_{lt} = \begin{cases} P & if\ s'_i \in \{s'_1, \ldots, s'_7\} \\ EA & otherwise \end{cases} \tag{2.19}$$

Table 2.3: Examples of state representations

$\mathcal{A}_{lg}$ States	Representation	$\mathcal{A}_{lt}$ States	Representation
s_1	$[\bar{v}_1, 0, \varepsilon]$	s'_1	$[\bar{v}_1, \bar{y}_1, \bar{\psi}_1, \bar{g}_1, \varepsilon]$
s_2	$[\bar{v}_2, 0, \varepsilon]$	s'_2	$[\bar{v}_2, \bar{y}_1, \bar{\psi}_1, \bar{g}_1, \varepsilon]$
s_3	$[\bar{v}_3, 0, \varepsilon]$	s'_3	$[\bar{v}_3, \bar{y}_1, \bar{\psi}_1, \bar{g}_1, \varepsilon]$
s_4	$[\bar{v}_4, 0, \varepsilon]$	s'_4	$[\bar{v}_4, \bar{y}_1, \bar{\psi}_1, \bar{g}_1, \varepsilon]$
s_5	$[\bar{v}_5, 0, \varepsilon]$	s'_5	$[\bar{v}_5, \bar{y}_1, \bar{\psi}_1, \bar{g}_1, \varepsilon]$
s_6	$[\bar{v}_6, 0, \varepsilon]$	s'_6	$[\bar{v}_6, \bar{y}_1, \bar{\psi}_1, \bar{g}_1, \varepsilon]$
s_7	$[\bar{v}_7, 0, \varepsilon]$	s'_7	$[\bar{v}_7, \bar{y}_1, \bar{\psi}_1, \bar{g}_1, \varepsilon]$
s_8	$[\bar{v}_2, 0, med]$	s'_8	$[\bar{v}_2, \bar{y}_2, \bar{\psi}_2, \bar{g}_2, med]$
s_9	$[\bar{v}_4, 0, low]$	s'_9	$[\bar{v}_3, \bar{y}_2, \bar{\psi}_2, \bar{g}_2, med]$
s_{10}	$[\bar{v}_5, 0, low]$	s'_{10}	$[\bar{v}_4, \bar{y}_2, \bar{\psi}_2, \bar{g}_2, med]$
s_{11}	$[\bar{v}_6, 0, low]$	s'_{11}	$[\bar{v}_5, \bar{y}_2, \bar{\psi}_1, \bar{g}_2, med]$
s_{12}	$[\bar{v}_7, 0, low]$	s'_{12}	$[\bar{v}_6, \bar{y}_2, \bar{\psi}_2, \bar{g}_2, med]$
s_{13}	$[\bar{v}_3, 1, med]$	s'_{13}	$[\bar{v}_7, \bar{y}_2, \bar{\psi}_2, \bar{g}_2, med]$
s_{14}	$[\bar{v}_3, 1, \varepsilon]$	s'_{14}	$[\bar{v}_{2,3}, \bar{y}_3, \bar{\psi}_3, \bar{g}_1, med]$
s_{15}	$[\bar{v}_8, 0, \varepsilon]$	s'_{15}	$[\bar{v}_8, \bar{y}_1, \bar{\psi}_1, \bar{g}_1, \varepsilon]$

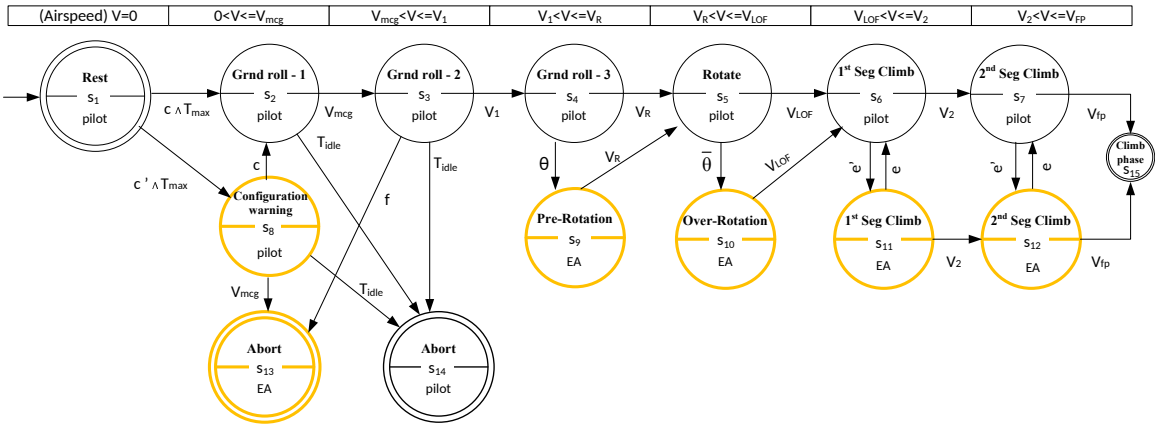

Figure 2.13: DMM for longitudinal takeoff dynamics (see Table 2.2)

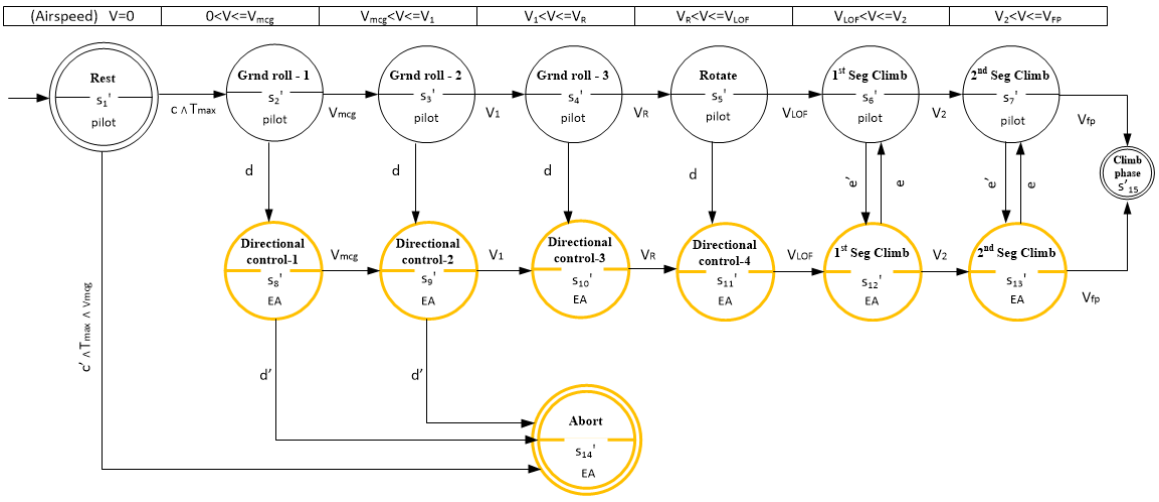

Figure 2.14: DMM for lateral-directional takeoff dynamics (see Table 2.2)

Transitions $\mathcal{T}_{lt}$ are shown as edges in the Fig 2.14 DMM graph. Each state s' is defined as the quintuple $[\bar{\mathcal{V}}, \bar{\mathcal{Y}}, \bar{\Psi}, \bar{\Upsilon}, \mathcal{R}]$. $\bar{\mathcal{V}}$, and $\mathcal{R}$ are defined as in DMM $\mathcal{A}_{lg}$. $\bar{\mathcal{Y}}$ represents discretized cross track errors with y_1 and y_2 defined as in Fig 2.11.

$$\bar{\mathcal{Y}} \in \{ \bar{y}_i \},\ i = 1,2,3 \tag{2.20}$$

$$\bar{y}_1 = \{ y \in \mathbb{R} \,|\, |y| \leq |y_1| \} \tag{2.21}$$

$$\bar{y}_2 = \{ y \in \mathbb{R} \,|\, |y_1| < |y| \leq |y_2| \} \tag{2.22}$$

$$\bar{y}_3 = \{ y \in \mathbb{R} \,|\, |y| > |y_2| \} \tag{2.23}$$

$\bar{\Psi}$ represents discrete inertial heading intervals with deviation constraints ψ_1, ψ_2 also shown in Fig 2.11:

$$\bar{\Psi} \in \{ \bar{\psi}_i \},\ i = 1,2,3 \tag{2.24}$$

$$\bar{\psi}_1 = \{ \psi \in [-\pi, \pi] \,|\, |\psi| < |\psi_1| \} \tag{2.25}$$

$$\bar{\psi}_2 = \{ \psi \in [-\pi, \pi] \,|\, |\psi_1| \leq |\psi| \leq |\psi_2| \} \tag{2.26}$$

$$\bar{\psi}_3 = \{ \psi \in [-\pi, \pi] \,|\, |\psi| > |\psi_2| \} \tag{2.27}$$

$\bar{\Upsilon}$ represents lateral acceleration given by

$$\bar{\Upsilon} \in \{ \bar{g}_i \},\ i = 1,2 \tag{2.28}$$

$$\bar{g}_1 = \{ \ddot{y} \in \mathbb{R} \,|\, |\ddot{y}| \leq |\ddot{y}_1| \} \tag{2.29}$$

$$\bar{g}_2 = \{ \ddot{y} \in \mathbb{R} \,|\, |\ddot{y}| > |\ddot{y}_1| \} \tag{2.30}$$

Directional control constraint violations often arise due to pilot induced oscillations (PIO) [33]. If one or more lateral constraint thresholds are violated (flagged by d), FSAM logic transfers control to the envelope-aware controller which then attempts to bring the aircraft within nominal (low risk) bounds. If the envelope-aware controller is not able to maintain the aircraft within the specified bounds (flagged by d'), then FSAM aborts the takeoff.

The overall FSAM DMM for takeoff is defined by the parallel composition (i.e., concurrent execution) of both $\mathcal{A}_{lg}$ and $\mathcal{A}_{lt}$. Although the two machines ($\mathcal{A}_{lg}$ and $\mathcal{A}_{lt}$) have a similar structure, they may not follow analogous transition sequences. For example, in case of an imminent tail strike during rotation, $\mathcal{A}_{lg}$ transitions from $s_5 \xrightarrow{\bar{\theta}} s_{10} \xrightarrow{V_{lof}} s_6$ and $\mathcal{A}_{lt}$ transitions from $s'_5 \xrightarrow{V_{lof}} s'_6$, i.e, FSAM transfers longitudinal control to the EA controller while retaining directional control with the pilot. This notion of decoupling the longitudinal and directional control authorities, though convenient from a system design perspective, may or may not be welcomed or easily understood by flight crews. Analyzing the benefits of a coupled versus decoupled FSAM formulation would require human subject evaluations beyond the scope of this work.

2.8 Case Study

In this section we present a case study to illustrate and evaluate use of FSAM for takeoff. The case study is based on accident data obtained from the flight data recorders.

2.8.1 Loss of Directional Control in Continental Airlines Flight 1404

The behavior and effectiveness of FSAM's takeoff DMM were first analyzed using a case study based on Continental Airlines FL1404 accident [33]. Due to severe crosswinds during takeoff, the Boeing 737 veered off the side of the runway after the pilot failed to maintain directional control. Fig 2.16 illustrates relevant parameters extracted from the Flight Data Recorder (FDR). After 10 seconds, the aircraft veers away from runway heading (heading transitions from 0^0 to -30^0) when the crosswinds exceeds 40 knots. The NTSB determined the probable cause of the accident as "*The captain's cessation of rudder input, which was needed to maintain directional control of the airplane, about four seconds before the excursion, when the airplane encountered strong gusty crosswind that exceeded the captain's training and experience.*" This is reflected in Fig 2.16, showing that the pilot relaxes the rudder pedals following a large rudder input at roughly 5 seconds.

To study the behavior of FSAM DMM in response to scenarios similar to FL 1404, a

Figure 2.15: Flight crew and FSAM functions during takeoff

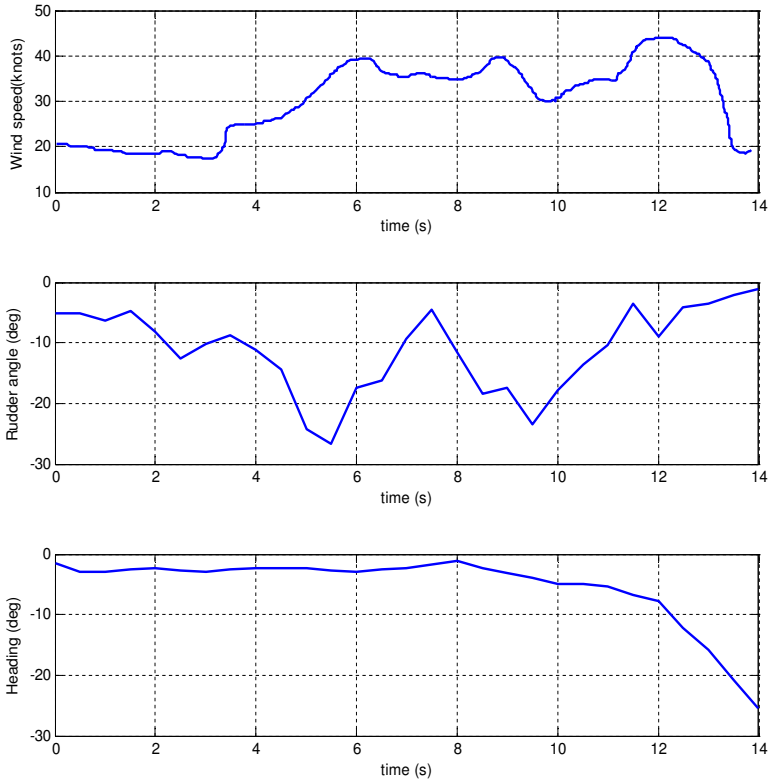

Figure 2.16: Accident data from flight data recorder

lateral runway excursion was simulated (Fig 2.17). Details about the physical models and the controller design can be found in Appendices A-C. The results of the simulation are shown in Fig 2.18. These plots illustrate the dynamics of an aircraft augmented with the FSAM DMM taking off in a severe crosswind. The FSAM DMM transfers lateral control of the aircraft from the pilot to the EA-controller when the aircraft exits the inner threshold with respect to heading ($|\psi| > |\psi_1|$) (see Fig 2.18). The Envelope-Aware controller is able to steer the aircraft back within the inner thresholds. After the aircraft is stabilized on the initial departure climb, lateral control is transferred back to the pilot. To enable a sensitivity analysis, we also chose different thresholds and crosswind magnitudes. In each scenario, FSAM consistently rejected the takeoff whenever possible [45].

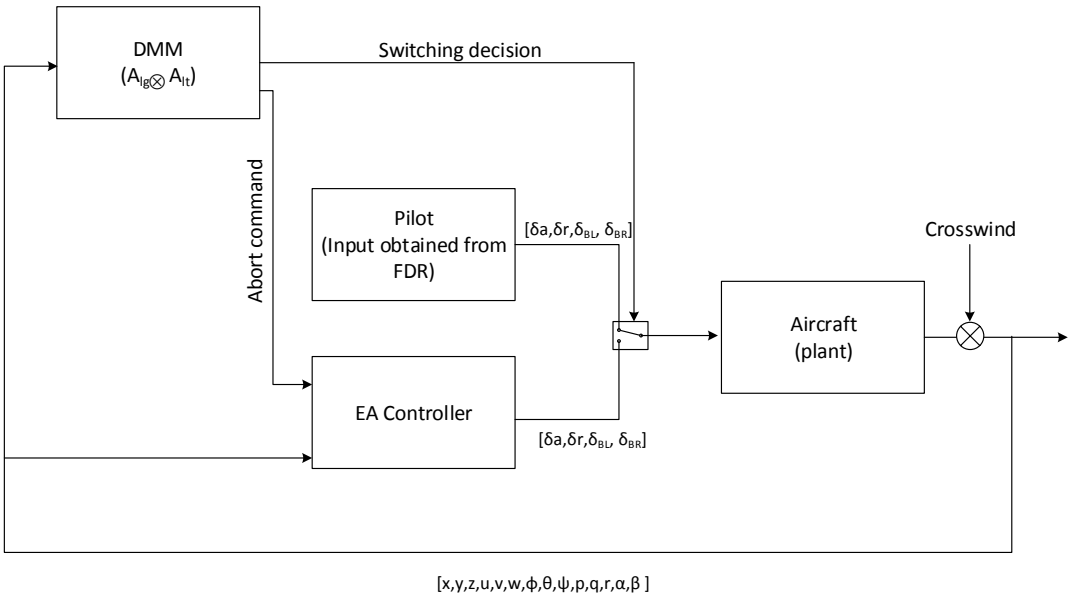

Figure 2.17: Simulation setup

2.9 Discussion

The takeoff FSAM DMMs were able to avoid LOC for the presented case study scenario but are not yet complete. Although a DMM will only be capable of executing the LOC mitigation sequences for which it has been designed, it would be possible to construct a DMM database that identifies and reacts to a broad suite of known risk factors, e.g see Fig 2.4. Ultimately, if FSAM encounters a scenario it hasn't been designed to handle, FSAM must recognize this or at least ensure the crew remains in charge to handle the situation, a capability requiring further research. Verifying that FSAM never initiates an override in scenarios for which it was not designed will be the key to safety certification.

The DMMs illustrated in this chapter were manually constructed. This process is not scalable to all known risks over all phases of flight. Furthermore, the full suite of FSAM DMMs need to be collectively verified and validated to assure no unexpected interactions between DMMs will cause inappropriate FSAM response.

2.10 Conclusion

This chapter has presented a Flight Safety Assessment and Management (FSAM) capability that identifies and mitigates risks associated with Loss of Control (LOC). FSAM initially warns the crew of imminent LOC risks. It overrides to an alternate recovery controller if

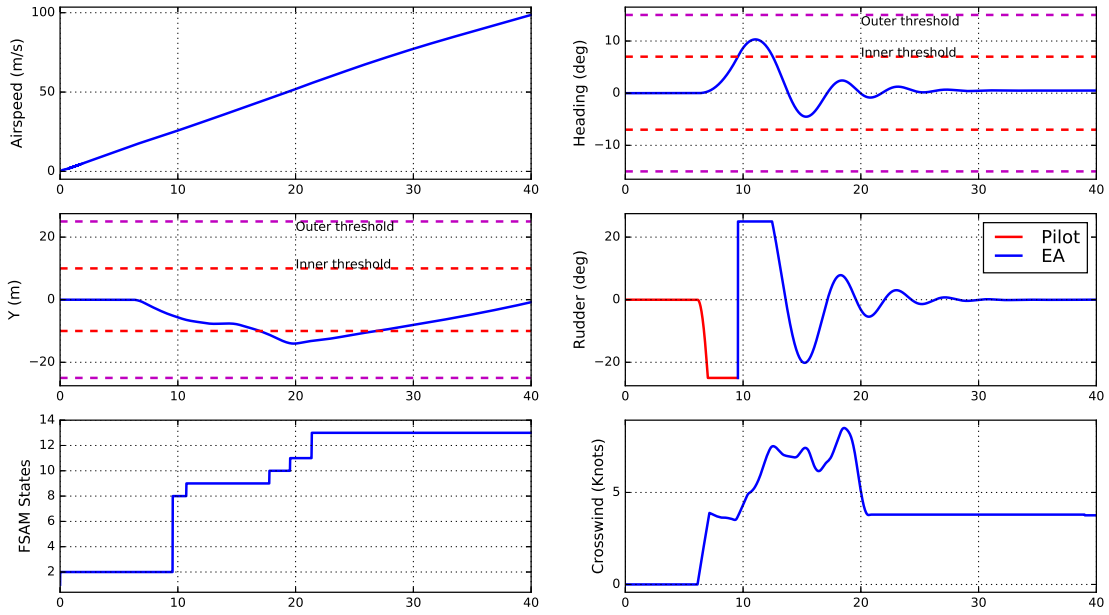

Figure 2.18: Continued takeoff scenarios (case study 1)

the crew fail to mitigate the risk. FSAM is formulated as a Deterministic Moore Machine (DMM) and applied to the takeoff phase of flight. The FSAM DMM machines are evaluated on case studies motivated by real world aviation accidents and incidents. Results show that a capable FSAM implementation can potentially avert LOC. This chapter contributes a novel formulation to ensure takeoff flight safety using deterministic Moore machines. The nominal sequence of states in the DMMs have a one to one correspondence with the typical V-speed decision sequence on which a pilot is trained. Furthermore, envelopes have been developed for the takeoff phase that simplify identification of safe and unsafe regions with respect to translational, rotational and directional takeoff dynamics. The takeoff phase in commercial aircraft is the only phase which remains manually flown. The work presented in this chapter can play a significant role in ensuring manual control is safe.

For a more comprehensive safety management system, hazards associated with conditions such as instrument failures, actuator failures, structural problems and other traffic must also be recognized and handled by FSAM. FSAM DMMs must also be developed for the other phases of flight. Each DMM will ultimately require verification prior to certification and must be integrated into an informative crew interface display for manned transport applications. The decision-making system described in this chapter can ultimately be extended to provide a comprehensive and verified means of avoiding LOC, the leading cause of aviation accidents today. Further work to accommodate additional FSAM cases and verification is described in subsequent chapters.

CHAPTER 3

A Decision Theoretic Formulation for Flight Safety Assessment and Management

3.1 Introduction

This chapter explores the use of decision-theoretic planning to allow override actions to be optimized using 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. Decision-theoretic techniques have been used for the development and enhancement of the Traffic Collision Avoidance System (TCAS). Kochenderfer et al. [46,47], Temizer et al. [48] and Winder et al. [49] have used the Markov Decision Process (MDP) and Partially Observable Markov Decision Process (POMDP) 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 as was described above in Chapter 2. Methods such as hybrid automata and reachability analysis can further guide the designer in defining appropriate switching strategies/rules [50]. Finite state machines can also be synthesized from Linear Temporal Logic (LTL) specifications [51]. However, manually specifying a state machine can be inefficient when the machine needs to address a broad class of scenarios. Use of a planner [52–54] 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 chapter therefore uses an MDP to generate a look-up 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 to address 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 sub-level MDPs where each sub-level MDP is responsible for preventing LOC for a specific phase of flight or specific suite of elevated risk factors. This chapter 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 also promotes better understanding of the resulting policy. This MDP formulation is extended with constraints to ensure unsafe states are unreachable. Note that this chapter 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 [55–60] and are not the focus of this dissertation.

The rest of this chapter is organized as follows. Section 3.2 reviews the MDP while Section 3.3 proposes a comprehensive FSAM MDP formulation. This full formulation is computationally intractable, so Section 3.4 simplifies the original MDP formulation to address a suite of takeoff LOC risk factors. Section 3.6 discusses the constrained MDP framework. Section 3.8 applies the takeoff FSAM MDP formulation to a real-world aviation incident. Sections 3.9 and 3.10 provide a discussion and conclusion, respectively.

3.2 Background

An observable MDP [61, 62] 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. $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ represents the transition probabilities associated with transitions from a given state to another state when executing an action. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ represents a reward function that assigns a finite real value to each state-action pair. Actions $a_n \in \mathcal{A}$ for each state $s_n \in \mathcal{S}$ at each decision epoch are chosen such that they maximize the expected cumulative discounted reward function of the form

$$\mathcal{V}(s_0) = \mathbb{E}\Big[\sum_{n=0}^{\infty} \gamma^n \mathcal{R}(s_n, a_n)\Big] \tag{3.1}$$

where $s_n \in \mathcal{S}$ is the current state and $a_n \in \mathcal{A}$ is the action selected at the current state. $\gamma \in (0, 1]$ is a discount factor chosen to emphasize short versus long-term rewards. A policy

Algorithm 3.1 Value Iteration

Assign $\mathcal{V}_o(s)$ arbitrarily, and $\mathcal{V}_n(s) := \mathcal{V}_o(s) + \frac{\epsilon(1-\gamma)}{\gamma}$, $\forall\ s \in \mathcal{S}$
while $\|\mathcal{V}_n - \mathcal{V}_o\| < \frac{\epsilon(1-\gamma)}{\gamma}$ **do**
  for all $s \in \mathcal{S}$ **do**
    for all $a \in \mathcal{A}$ **do**
      $\mathcal{V}^a(s) := \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}(s,a,s')\mathcal{V}_n(s')$
    end for
    for all $s \in \mathcal{S}$ **do**
      $\mathcal{V}_o(s) := V_n(s)$
      $\mathcal{V}_n(s) := \max_{a \in \mathcal{A}} \left\{ \mathcal{V}^a(s) \right\}$
    end for
  end for
end while

is defined as the mapping π where $\pi : \mathcal{S} \rightarrow \mathcal{A}$. The optimal policy π^* is given by:

$$\pi^*(s) = \arg\max_a \left\{ \mathcal{Q}(s,a) \right\} \tag{3.2}$$

$$\mathcal{Q}(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{T}(s,a,s')\mathcal{V}^*(s') \tag{3.3}$$

$\mathcal{V}^*(s)$ is the optimal value of state s and can be obtained using algorithms such as value iteration, policy iteration or linear programming [62].

MDPs can be used to solve a wide range of sequential decision making problems (see [61–64] for different applications of MDPs). Encoding a sequential decision making problem as an MDP requires specification of a suitable state $\mathcal{S}$ and action representation $\mathcal{A}$, transition probabilities $\mathcal{T}$ that describe the influence of the actions on the states and reward function $\mathcal{R}$ that captures the objective of the decision problem. The conventional algorithms that perform value or policy Iteration explicitly enumerate the states to find the optimal policy. Consequently, these algorithms incur the curse of dimensionality and hence do not scale well for problems with large state dimensions. Alternately, approximate methods such as state aggregation [63], value function approximation [65], policy search methods [66, 67], differential dynamic programming [68] and sparse sampling [69] have been successfully used to solve problems involving continuous state-spaces or large state dimensions.

3.3 Markov Decision Process Formulation for FSAM

FSAM MDP state must capture all information necessary to make risk-optimal warning and override decisions. This section describes a full set of state features that might enable an FSAM MDP to make appropriate decisions regarding control authority. As will become apparent, this comprehensive state model is large, motivating subsequent abstractions and simplifications.

3.3.1 State features

State features relevant to LOC risk assessment and decision making can be broadly classified into 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). A description of each feature is provided below. Each state $s \in \mathcal{S}$ of the FSAM MDP formulation is represented by their composition:

$$s = [F_1, F_2, F_3, F_4] \tag{3.4}$$

3.3.1.1 Aircraft dynamics and control

F_1 represents the evolution of the continuous dynamics of the aircraft and is viewed as the composition of the following sub-features:

$$\begin{aligned} F_1 &= [F_{11}, F_{12}, F_{13}, F_{14}] \\ F_{11} &= [u, v, w, p, q, r, \phi, \theta, \psi, x, y, z] \\ F_{12} &= [\delta_e, \delta_a, \delta_r, \delta_t] \\ F_{13} &= [c_g, c_f, c_p, f_{sys}] \\ F_{14} &= [\bar{M}, \bar{S}] \end{aligned} \tag{3.5}$$

Here F_{11} describes traditional aircraft *physical* state [70]. u, v, w describe aircraft velocity, p, q, r are the body axis angular rates, ϕ, θ, ψ represent Euler angle attitude, and x, y, z denote 3-D position. F_{12} describes fixed-wing control inputs elevator (δ_e), aileron (δ_a), rudder (δ_r), and throttle (δ_t). F_{13} describes the configuration of the aircraft in terms of flaps (c_f), spoilers (c_p) and landing gear (c_g). $f_{sys} = (f_{s_1}, \ldots, f_{s_n})$ are flags that denote the on/off state of various flight systems with potential to influence LOC risk, e.g., pitot or wing heat. F_{14} specifies the control mode status. In this work, FSAM defines exactly two control modes $\bar{M}$, one representing the nominal pilot-autopilot system and another representing

the envelope-aware controller that either an FSAM override or the pilot can activate via the mode select switch $\bar{S}$. F_{11} and F_{12} represent continuous-valued variables, F_{13} and F_{14} take discrete values. All F_1 parameters are observable from onboard sensors.

Monitoring the aircraft's continuous dynamic states (F_{11}) enables prevention of aerodynamic, structural and performance constraint violations. Violation of these constraints can lead to catastrophic events. Monitoring continuous control inputs (F_{12}) helps identify conditions such as control surface saturation, cross-control, or inadequate thrust. Monitoring the configuration (F_{13}) of the aircraft with respect to the landing gear, flaps, spoilers and various subsystems helps ensure that the aircraft is configured properly for a given phase of flight.

3.3.1.2 Aircraft health

F_2 describes aircraft and subsystem health status:

$$F_2 = [h_{eng}, h_{act}, h_{sys}] \tag{3.6}$$

$h_{eng} = (h_{e_1}, \ldots, h_{e_n})$ are flags that denote engine operational state (nominal/inoperative), $h_{act} = (h_{a_1}, \ldots, h_{a_n})$ denote control surface actuator status (nominal/jammed/free-floating), and $h_{sys} = (h_{s_1}, \ldots, h_{s_n})$ denote the status (nominal/failed) of onboard support systems such as cabin pressurization, heating, fuel pumps, power systems and anti-icing. All F_2 features are discrete and observable from sensor and health monitoring subsystems.

Aircraft performance limits are closely coupled with the aircraft's health status. External factors such as icing can also degrade airplane performance. System failures can result in deactivation of certain features in a fly-by-wire system, such as the flight director or envelope protection systems, which can lead to increased LOC risk particularly in conjunction with inappropriate crew inputs.

3.3.1.3 Operator

Flight crew state can substantially increase the likelihood of inappropriate crew inputs, a critical factor in an FSAM override decision. The following pilot state abstraction F_3 is proposed:

$$F_3 = [F_{31}, F_{32}, F_{33}] \tag{3.7}$$

$F_{31} \subset \{CP, FO\}$ indicates who is present in the cockpit, with Captain (CP) and First Officer (FO) represented as an example. $F_{32} = (h_{CP}, h_{FO})$ where $h_{CP}, h_{FO} \in \{nominal, unconscious,$

$fatigued\}$ compactly classify crew health. $F_{33} \in \{nominal, abnormal\}$ classifies cockpit activity. All F_3 attributes are discrete. While this work does not claim progress in translating sensor observations to "human state estimates", research has shown that such observers are feasible [71].

3.3.1.4 Environment

Both flight controller and crew performance can be influenced by the environment, most critically atmospheric conditions F_4:

$$F_4 = [f_{winds}, f_{visibility}, f_{temp}, f_{precip}] \tag{3.8}$$

where $f_{winds} \in \mathbb{R}^3$ represents the wind vector, $f_{visibility} \in \mathbb{R}$ is visibility, $f_{temp} \in \mathbb{R}$ is surrounding air temperature, and $f_{precip} \in \{none, rain, snow, hail, tstorm\}$ denotes precipitation in various forms. $f_{wind}, f_{visibility}, f_{temp}$ are continuous while f_{precip} is discrete in this formulation. Wind, visibility and temperature can be estimated from onboard sensors supplemented by forecasts and station reports. Precipitation and thunderstorms can be sensed via weather radar and storm scope along with meteorological reports and forecasts accessed via a datalink.

3.3.2 Action

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: NOOP (No Operation) and TOGL (Toggle). Any time FSAM selects NOOP, the current control mode is likely to remain engaged. If the current control mode indicates nominal pilot/autopilot authority and FSAM selects the TOGL action, FSAM activates the envelope-aware controller. If the current control mode 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 request control back from the envelope-aware control via the mode select switch $\bar{S}$.

3.3.3 Transition probabilities

State transition dynamics can be modeled as a Markov chain with consideration for each FSAM MDP action (NOOP or TOGL). Because transition dynamics fundamentally evolve as a function of current control mode $\bar{M}$, one transition probability matrix is defined for each control mode $\bar{M}$. Switching between the two probability tables for $\bar{M} = P$ and $\bar{M} = EA$ then occurs for each state in which the MDP selects the TOGL action. Let $\bar{\mathcal{T}}_M, M \in \{P, EA\}$ denote the Markov chain transition matrix under control mode M. Given the state abstractions proposed in this work (described in the following sections), the underlying continuous time process remains in a state $s_k \in \mathcal{S}$ for some duration called the *sojourn time* $\sigma(s_k)$. σ is modeled as an exponential distribution with parameter $\beta(s_k)$. Estimated values of $\beta(s_k)$ and $\bar{\mathcal{T}}_M(s_j|s_i),\ s_i, s_j \in \mathcal{S}$ can be computed from flight/simulation data.

$$\beta(s_k) = \mathbb{E}\left[\frac{1}{\sigma(s_k)}\right] \tag{3.9}$$

$$\bar{\mathcal{T}}_M(s_j|s_i) = \frac{N(s_i, s_j)}{\sum\limits_n N(s_i, s_n)}, \text{ where } n = 1, \ldots, |\mathcal{S}| \tag{3.10}$$

Here $N(s_i, s_j)$ represents the total number of transitions from state s_i to s_j. The above Markov chain has state-dependent sojourn times. A discrete-time Markov chain can be transformed into an equivalent Markov chain $\mathcal{T}_M$ whose sojourn time distributions are identical for all states through a *uniformization* process [62] in which the uniformized Markov chain $\mathcal{T}_M$ is described by:

$$\mathcal{T}_M(s_j|s_i) = \begin{cases} 1 - \frac{1}{c}(1 - \bar{\mathcal{T}}_M(s_i|s_i))\beta(s_i) & \text{if } s_i = s_j \\ \frac{1}{c}(\bar{\mathcal{T}}_M(s_j|s_i)\beta(s_i)) & \text{otherwise} \end{cases} \tag{3.11}$$

where $c = \max\limits_{s_i \in \mathcal{S}}\{1 - \bar{\mathcal{T}}_M(s_j|s_i)\beta(s_i)\}$ is the sojourn time distribution parameter for the new uniformized Markov chain.

An alternate method of evaluating $\mathcal{T}_M$ is to consider each state as the composition of its individual state features. Consequently, the required probability distribution can be expressed in terms of the individual state features as follows:

$$\mathcal{T}_M(s_{n+1}|s_n) = \mathcal{T}_M(F_1^{n+1}, F_2^{n+1}, F_3^{n+1}, F_4^{n+1}|F_1^n, F_2^n, F_3^n, F_4^n) \tag{3.12}$$

Assuming that the features at future time steps $(n+1)$ depend only on the current features

(n) and also exploiting conditional independence relations among the state features, Eqn (3.12) can be simplified as follows:

$$\begin{aligned}\mathcal{T}_M(s_{n+1}|s_n) =& \mathcal{T}_1(F_1^{n+1}|F_1^n, F_2^n, F_3^n, F_4^n)\times \\ & \mathcal{T}_2(F_2^{n+1}|F_1^n, F_2^n, F_3^n, F_4^n)\times \\ & \mathcal{T}_3(F_3^{n+1}|F_1^n, F_2^n, F_3^n, F_4^n)\times \\ & \mathcal{T}_4(F_4^{n+1}|F_4^n)\end{aligned} \tag{3.13}$$

$\mathcal{T}_1$ represents transition dynamics of the traditional aircraft states along with discrete flight states, while $\mathcal{T}_2, \mathcal{T}_3$ and $\mathcal{T}_4$ represent transition probabilities associated with aircraft health, pilot state, and environment characteristics respectively. Environment state F_4 is independent of the other state features. The terms in Eqn (3.13) can be further simplified via conditional independence between state sub-features. It can be ensured that the distributions $\mathcal{T}_1, \ldots, \mathcal{T}_4$ have uniform sojourn times using Eqn (3.11).

3.3.4 Reward formulation

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:

$$\mathcal{R}(s,a) = \sum_{i=0}^{n} \eta_i \mathcal{R}_i \tag{3.14}$$

The $\mathcal{R}_i$'s penalize unsafe states and unnecessary *toggle* actions while η_i's 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 depending on the availability of recovery margins. Weighting parameters may also be learned from accident flight data and investigation board recommendations.

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 below for *takeoff*. Abstract state features are based on flight envelopes and their

translation to a suitable reward or cost function.

3.4 FSAM MDP for Takeoff

3.4.1 State Formulation

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 while 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 $\bar{M}$, mode select switch status $\bar{S}$, throttle control input $\bar{T}$ and engine health status $\bar{E}$. The state for a takeoff MDP formulation is represented as:

$$\begin{aligned} & s \in \mathcal{S} \\ & s = [V, \theta, \psi, x, y, z, \bar{T}, \bar{M}, \bar{S}, \bar{E}] \end{aligned} \tag{3.15}$$

The above state-space 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 discrete state features.

Aircraft takeoff envelopes were analyzed with respect to translational, rotational and lateral dynamics in Chapter 2. In a nominal takeoff, the aircraft accelerates to liftoff speed from rest, lifts off and further accelerates to speed V_2 before reaching the obstacle height at the end of the runway. In case of an 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 in Chapter 2 and is summarized below.

The phase portrait in Fig 3.1 illustrates the evolution of the V-x dynamics under a rejected takeoff scenario. In Fig 3.1, rejecting the takeoff at an airspeed-position state below the blue curve leads to trajectories that decelerate and stop within the available runway length. This would correspond to a safe rejected takeoff. Rejecting the takeoff at a state above the blue 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 3.2). If an engine failure occurs at a point below the blue curve, the airplane has sufficient airspeed to accelerate, lift off and attain V_2 before reaching the obstacle height

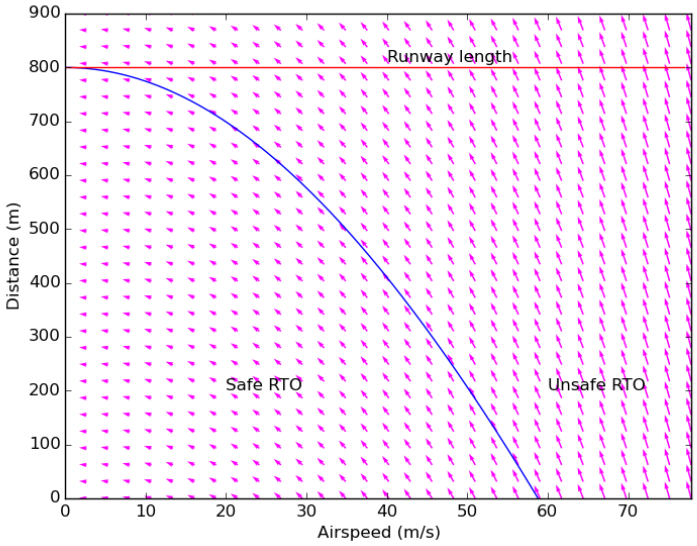

Figure 3.1: Rejected takeoff envelope

at the end of the runway. However, if an engine failure occurs at a point above the blue curve, the airplane has insufficient airspeed to accelerate to V_2 before the runway overrun. Combining the curves in figures 3.1 and 3.2 yields four distinct regions shown in Fig 3.3. Clearly a region exists 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 3.4.

Each curve in Fig 3.4 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 R_{max} denote the length of the runway. With these parameters, the $V - x$ state space is aggregated into 17 abstract states as shown in Fig 3.5. Note that states 15 and 16 in Fig 3.5 (a) represent runway overrun scenarios where the aircraft has crossed the available takeoff distance with inappropriate airspeed to either takeoff 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 $\theta \geq \theta^*, z \leq h^*$ denote the condition at which a tail strike occurs where θ^* is the tail strike pitch attitude when the aircraft is below altitude h^*. Let $\theta_1 = 0.2\theta^*$ and $\theta_2 = 0.8\theta^*$. With these parameters,

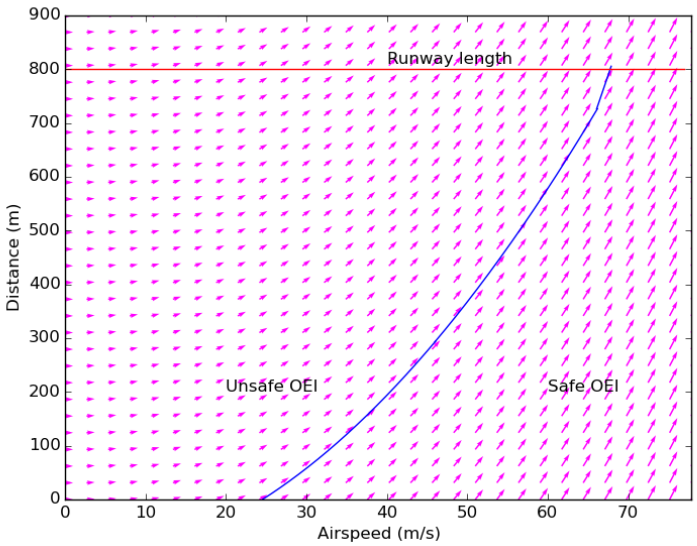

Figure 3.2: One-engine inoperative envelope

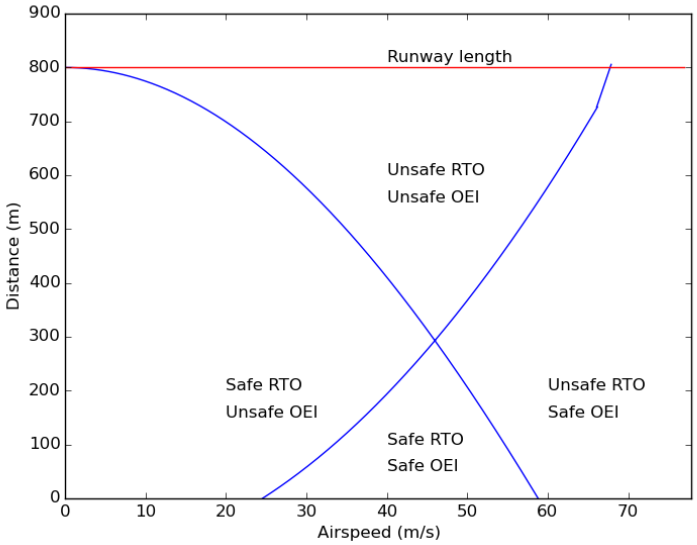

Figure 3.3: RTO and OEI envelopes - safe vs unsafe zones

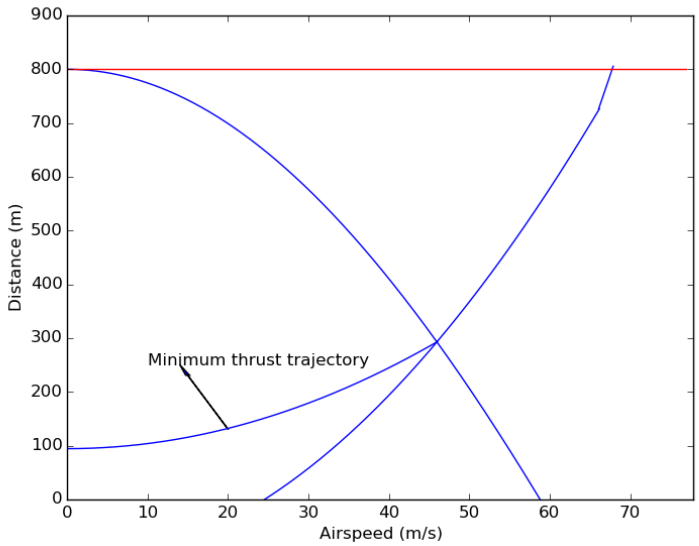

Figure 3.4: RTO and OEI envelopes - safe vs unsafe zones

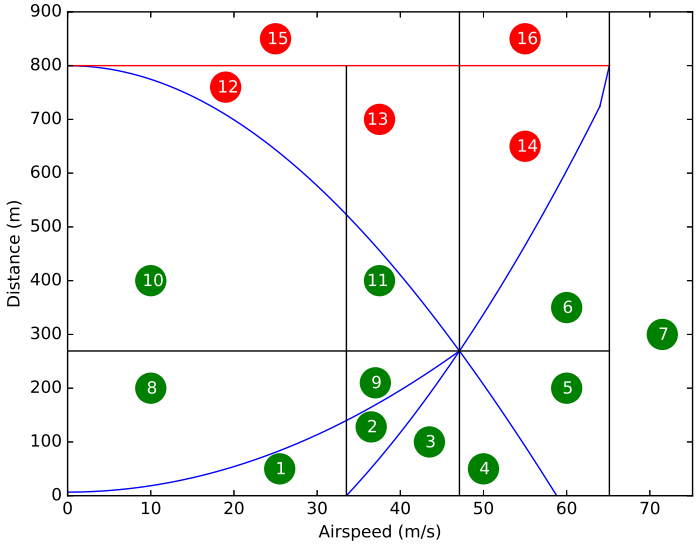

Figure 3.5: Partitions of V-X space

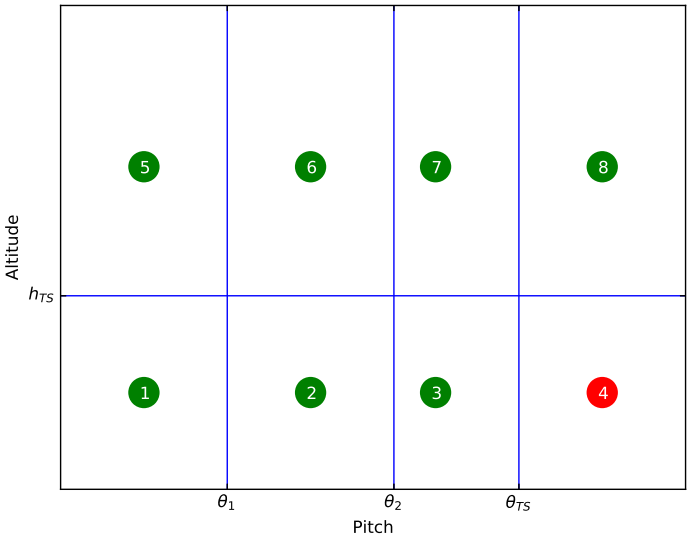

Figure 3.6: Partitions of $\theta - H$ space

pitch-altitude space is aggregated as shown in Fig 3.6.

To abstract lateral-directional states, 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 $\psi_1 = \psi_0 + 4^\circ$ and $\psi_2 = \psi_0 + 10^\circ$. With these parameters, partitions of the lateral displacement and yaw space are obtained as shown in Fig 3.7.

Thrust control inputs for takeoff are discretized as $\bar{T} \in \{T_{idle}, T_{max}\}$. $\bar{M} \in \{P, EA\}$ denotes the available control authorities where P denotes the Pilot and EA denotes the Envelope-Aware controller. The engine health status takes values $\bar{E} \in \{E_{AEO}, E_{OEI}, E_{AEI}\}$ where E_{AEO} represents "all engines operational", E_{OEI} represents "one engine inoperative" and E_{AEI} represents "all engines inoperative".

With the compact state features described above, the initial state formulation in Eqn

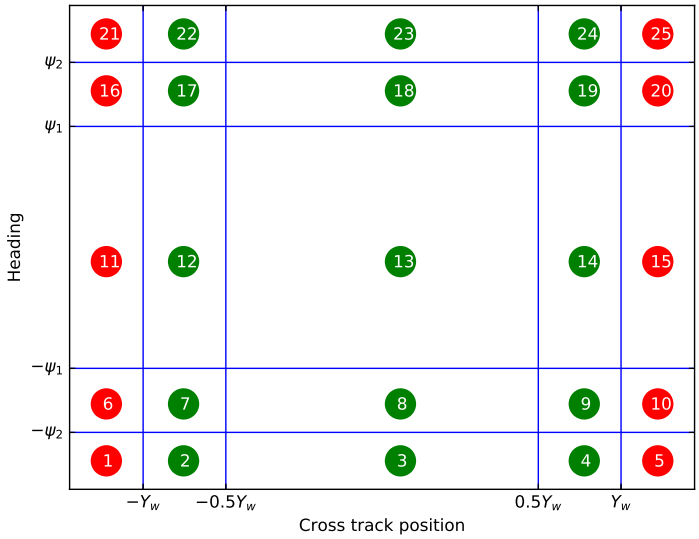

Figure 3.7: Partitions of $Y - \psi$ space

3.15 is transformed into:

$$
\begin{aligned}
& s \in \mathcal{S} \\
& s = [\bar{Q}, \bar{P}, \bar{L}, \bar{M}, \bar{\delta}_t, h_{eng}] \\
& \bar{Q} \in \{q_1, q_2, \ldots, q_{16}\} \\
& \bar{P} \in \{p_1, p_2, \ldots, p_8\} \\
& \bar{L} \in \{l_1, l_2, \ldots, l_{25}\} \\
& \bar{M} \in \{P, EA\} \\
& \bar{S} \in \{P, EA\} \\
& \bar{\delta}_t \in \{T_{idle}, T_{max}\} \\
& h_{eng} \in \{E_{AEO}, E_{OEI}, E_{AEI}\}
\end{aligned}
\tag{3.16}
$$

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.

3.4.2 Action Formulation

The goal of the takeoff MDP is to select the appropriate control mode with the TOGL action and mitigate imminent takeoff LOC risk. Thus, the actions of the takeoff MDP are

$$\bar{A} \in \{NOOP, TOGL\} \tag{3.17}$$

Eqn 3.17 can be extended to include warning actions to alert the flight crew as well. For simplicity, this work considers only the two actions shown above.

3.4.3 Reward formulation for Takeoff

In this work, an additive reward formulation is defined as in Eqn (3.14):

$$\mathcal{R}(s,a) = \eta_1 \mathcal{R}_1(\bar{Q}) + \eta_2 \mathcal{R}_2(\bar{P}) + \eta_3 \mathcal{R}_3(\bar{L}) + \eta_4 \mathcal{R}_4(\bar{M}, \bar{A}) \tag{3.18}$$

Here $\mathcal{R}_1(\bar{Q})$ penalizes unsafe states with respect to the translational dynamics (see Fig 3.5) and is given by:

$$\mathcal{R}_1(\bar{Q}) = \begin{cases} -1 \; if \; \bar{Q} \in \{q_{15}, q_{16}\} \\ 0 \;\; otherwise \end{cases} \tag{3.19}$$

$\mathcal{R}_2(\bar{P})$ penalizes unsafe states with respect to the rotational dynamics (see Fig 3.6):

$$\mathcal{R}_2(\bar{P}) = \begin{cases} -1 \; if \; \bar{P} \in \{p_4\} \\ 0 \;\; otherwise \end{cases} \tag{3.20}$$

$\mathcal{R}_3(\bar{L})$ penalizes unsafe states with respect to the lateral dynamics (see Fig 3.7):

$$\mathcal{R}_3(\bar{L}) = \begin{cases} -1 \; if \; \bar{L} \in \{l_1, l_5, l_6, l_{10}, l_{11}, l_{15}, l_{16}, l_{20}, l_{21}, l_{25}\} \\ 0 \;\; otherwise \end{cases} \tag{3.21}$$

$\mathcal{R}_4(\bar{M}, \bar{A})$ 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.

$$\mathcal{R}_4(\bar{M},\bar{A}) = \begin{cases} -1 \; if \; \bar{M} = P \wedge \bar{S} = P \wedge \bar{A} = TOGL \\ -o_1 \; if \; \bar{M} = EA \wedge \bar{S} = P \wedge \bar{A} = NOOP \\ -o_2 \; if \; \bar{M} = EA \wedge \bar{S} = EA \wedge \bar{A} = TOGL \\ 0 \; otherwise \end{cases} \tag{3.22}$$

where $0 \leq o_1 \leq 1$ and $0 \leq o_2 \leq 1$. η's in Eqn (3.18) are positive scalars which emphasis the relative importance of the individual reward terms. For this work, the reward function weights were manually tuned to ensure policies favored pilot control but did not allow the system to violate constraints. Statistics can assist in computing reward weights. For example, reference [2] 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 [72] can also be adapted to compute reward function parameters in future work.

3.4.4 Transition Probabilities

The transition probabilities are obtained using Monte Carlo simulations. Monte-Carlo simulations are performed using an aircraft dynamics model for takeoff (Appendix A), an envelope-aware controller (Appendix B.1) and a human pilot model (Appendix B.2). Different values of initial conditions, pilot model parameters and engine failure states are sampled from various distributions (see Appendix B.2) to simulate nominal and anomalous takeoff sequences. The transition probabilities between the MDP states described in Eqn (3.16) are estimated as described in Eqn (3.9)-(3.11). Let $\mathcal{T}_M$, $M \in \{P, EA\}$ denote the transition probabilities for mode M. Let $\mathcal{T}_{NOOP}$ denote the transition probability matrix for $a = NOOP$. Let $\mathcal{T}_{TOGL}$ denote the transition probability matrix for $a = TOGL$. The state features in Eqn (3.16) are permuted such that $\mathcal{T}_{NOOP}$ and $\mathcal{T}_{TOGL}$ can be viewed as block diagonal matrices of the form:

$$\mathcal{T}_{NOOP} = \begin{bmatrix} \mathcal{T}_{M=P} & 0 \\ 0 & \mathcal{T}_{M=EA} \end{bmatrix} \tag{3.23}$$

$$\mathcal{T}_{TOGL} = \begin{bmatrix} 0 & \mathcal{T}_{M=EA} \\ \mathcal{T}_{M=P} & 0 \end{bmatrix} \tag{3.24}$$

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

3.5 Takeoff MDP 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 76800 states in the above Takeoff MDP formulation. The resulting policy is stored as a look-up 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:

$$\mathcal{T}^i_\pi = \begin{cases} \mathcal{T}^i_{NOOP} \; if \; \pi(i) = NOOP \\ \mathcal{T}^i_{TOGL} \; if \; \pi(i) = TOGL \end{cases} \tag{3.25}$$

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 given by

$$\chi_n = \chi_0^T \mathcal{T}_\pi^n \tag{3.26}$$

The Markov chain representing the complete policy is also difficult to visualize, so segments of the policies as used to illustrate their properties. Fig 3.8 presents a policy segment that illustrates FSAM MDP policy response to an imminent runway excursion risk. For ease of illustration, only transitions in $\bar{M}, \bar{\delta}_T$ and $\bar{Q}$ are shown. Each node represents a discrete state (s) annotated with their corresponding features and optimal values $\mathcal{V}^*(s)$. 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 when 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}). Policy behavior can vary depending on the choice of weighting factors (η) in Eqn (3.18). For example, increasing the penalty on envelope-aware states (i.e $\bar{M} = EA$) in Eqn (3.22) can result in transfer of control back to the pilot immediately. The

following example illustrates the trade-off between increasing the cost of $NOOP$ versus the cost of $TOGL$.

Fig 3.9 presents an FSAM MDP policy segment showing response to a tail-strike risk. Only state transitions impacting risk level, specifically $\bar{M}, \bar{P}$ and $\bar{Q}$, are shown. FSAM overrides or toggles control authority to the envelope-aware controller when there is an imminent tail-strike risk (i.e. $\bar{P} = p_3$). The envelope-aware control law reduces pitch attitude to prevent tail-strike during rotation. However, due to a large penalty on the envelope-aware controller state (i.e. $\bar{M} = EA$), control is returned to the pilot immediately as would be the case in an automobile when anti-lock brakes or traction control systems temporarily engage. But according to the Monte-Carlo simulations based on the underlying models described in the Appendix, the probability of a tail-strike with the pilot in control is still non-zero as highlighted in yellow in Fig 3.9. The tail-strike risk can be eliminated by choosing a higher weighting factor for the R_2 term relative to R_4 in Eqn (3.18). This would encourage the policy to return control back to the pilot only after the aircraft is free from tail-strike risk.

The complete FSAM MDP policy manages elevated risks associated with runway excursions and overruns as well as potential tail-strikes by assuring inappropriate longitudinal and lateral control inputs are overridden in time to avoid LOC. The full policy must ultimately be verified to ensure 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.

3.6 Constrained MDPs for FSAM

The goal of this section is to construct a constrained MDP [73] policy that enables FSAM to make risk-optimal decisions in a given flight condition 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 (3.1) subjected to constraints of the form

$$\begin{aligned} \mathcal{T}(s_1^* | s_0) &\leq \bar{p}_1 \\ \mathcal{T}(s_2^* | s_0) &\leq \bar{p}_2 \\ &\vdots \\ \mathcal{T}(s_m^* | s_0) &\leq \bar{p}_m \end{aligned} \tag{3.27}$$

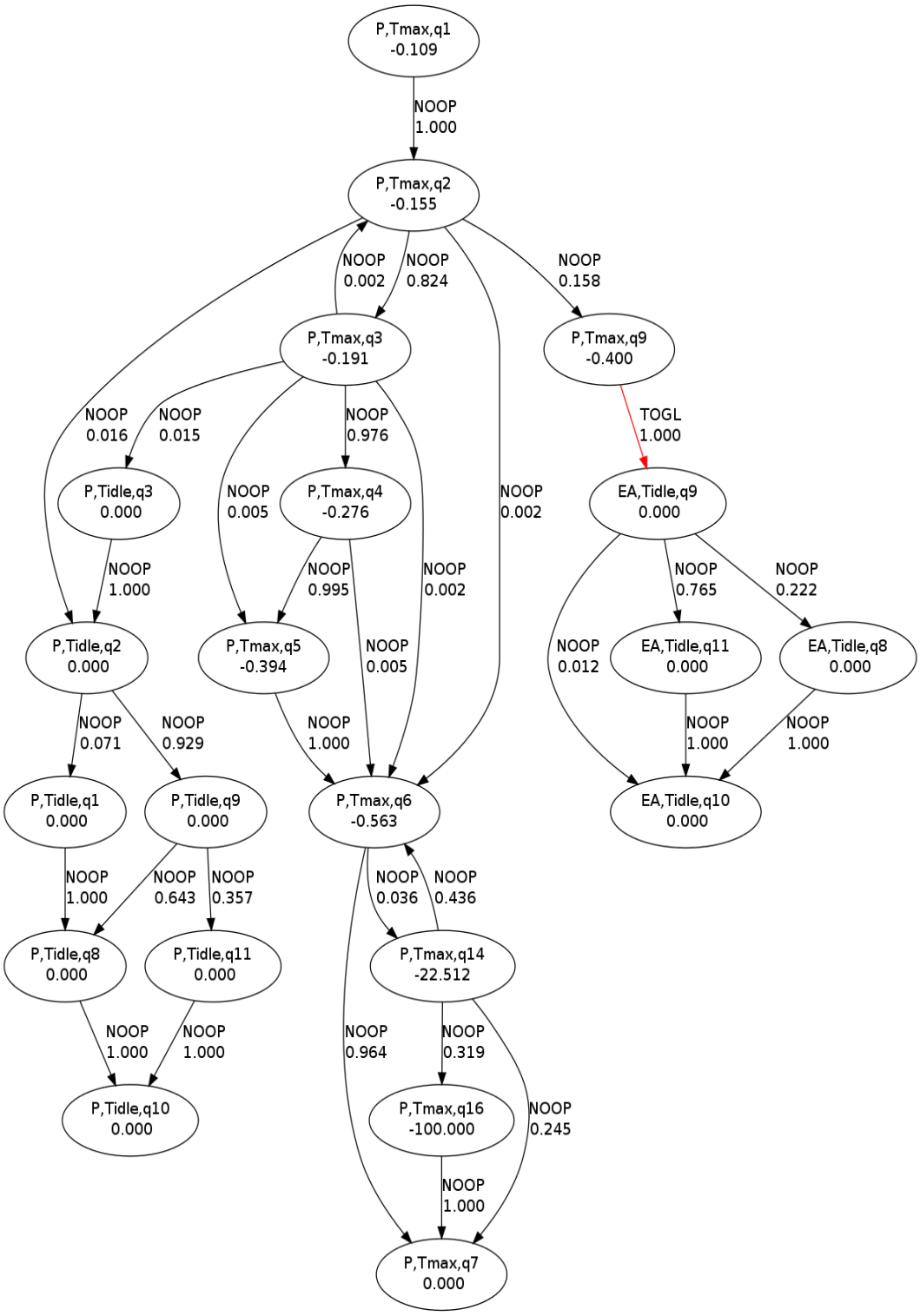

Figure 3.8: Runway excursion policy

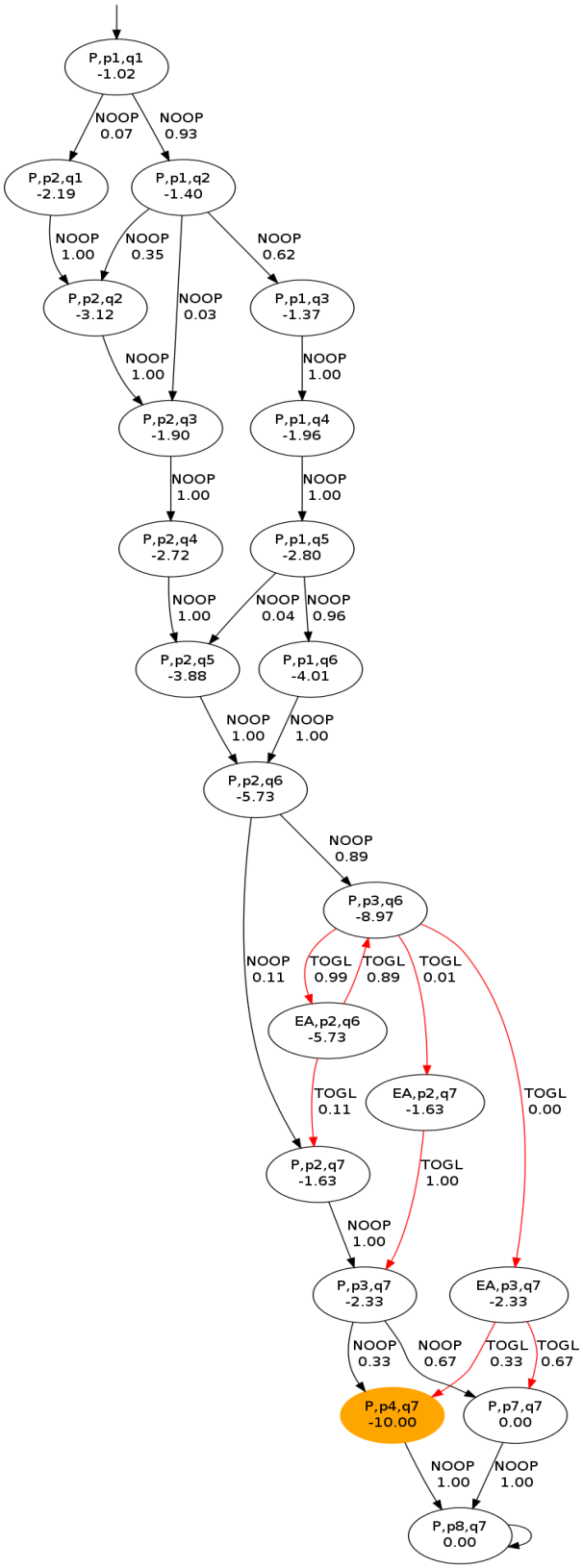

Figure 3.9: Tail-strike policy

Here $\mathcal{T}(s_i^*|s_0)$ is the conditional probability of entering state s_i^* from a given initial state s_0.

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

$$\mathcal{V}(s_0)_\pi = \mathbb{E}\Big[\sum_{n=0}^{\infty} \lambda^n \mathcal{R}(s_n, a_n)\Big]_{s_0} \tag{3.28}$$

For a Markov process, Eqn (3.28) can be expressed as

$$\begin{aligned}\mathcal{V}(s_0) &= \sum_{s_i \in s} \sum_{a_j \in \mathcal{A}} \sum_{n=0}^{\infty} \lambda^n \mathcal{T}(s_n = s_i, a_n = a_j | s_0) \mathcal{R}(s_n = s_i, a_n = a_j) \\ &= \sum_{s_i \in s} \sum_{a_j \in A} \rho(s_i, a_j)_{s_0}^{\pi} \mathcal{R}(s_n = s_i, a_n = a_j)\end{aligned} \tag{3.29}$$

Here $\rho(s_i, a_j)_{s_0}^{\pi}$ is defined as the occupational measure of the state-action pair (s_i, a_j).

$$\rho(s_i, a_j)_{s_0}^{\pi} := \sum_{n=0}^{\infty} \lambda^n \mathcal{T}(s_n = s_i, a_n = a_j | s_0) \tag{3.30}$$

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 Eqn (3.30) as follows

$$\begin{aligned}\sum_{a_j \in \mathcal{A}} \rho(s_i, a_j) &= \sum_{a_j \in \mathcal{A}} \sum_{n=0}^{\infty} \lambda^n \mathcal{T}(s_i, a_j | s_0) \\ &= \mathcal{T}(s_0) + \sum_{s_x \in \mathcal{S}} \sum_{a_y \in \mathcal{A}} \sum_{n=1}^{\infty} \lambda^{n-1} \mathcal{T}(s_x, a_y | s_0) \mathcal{T}(s_i | s_x, a_y) \\ &= \mathcal{T}(s_0) + \sum_{s_x \in \mathcal{S}} \sum_{a_y \in \mathcal{A}} \rho(s_x, a_y)_{s_0}^{\pi} \mathcal{T}(s_i | s_x, a_y)\end{aligned} \tag{3.31}$$

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

$$\sum_{a_j \in \mathcal{A}} \rho(s_i, a_j) - \sum_{s_x \in \mathcal{S}} \sum_{a_y \in \mathcal{A}} \rho(s_x, a_y)_{s_0}^{\pi} \mathcal{T}(s_i | s_x, a_y) = \mathcal{T}(s_0) \tag{3.32}$$

Eqns (3.29) and (3.32) can be expressed in their respective matrix forms as follows

$$\mathcal{V} = \mathcal{R}^T \rho \tag{3.33}$$

$$([I\,I\ldots I]-[\mathcal{T}_{a_1}^T\mathcal{T}_{a_2}^T\ldots\mathcal{T}_{a_n}^T])\rho=\xi \tag{3.34}$$

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 Eqn (3.33) and (3.34), the problem of maximizing the cumulative reward (Eqn (3.1)) is formulated as a linear program (LP) as follows

$$\max\mathcal{R}^T\rho \tag{3.35}$$

subject to the constraints

$$\begin{aligned}([I\,I\ldots I]-[\mathcal{T}_{a_1}^T\mathcal{T}_{a_2}^T\ldots\mathcal{T}_{a_n}^T])\rho&=\xi\\ \rho&\geq 0\end{aligned} \tag{3.36}$$

Note that the solution to Eqn (3.35) and (3.36) corresponds to the MDP without constraints (Eqn (3.1)). The additional constraints imposed by Eqn (3.27) are expressed as constraints on the occupational measures. For example, consider the constraint

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

The above constraint can be expressed as

$$\sum_{a_j\in\mathcal{A}}\mathcal{T}(s_i,a_j|s_0)\leq\bar{p}_i$$

$$\sum_{n=0}^{\infty}\lambda^n\sum_{a_j\in\mathcal{A}}\mathcal{T}(s_n=s_i,a_n=a_j|s_0)\leq\sum_{n=0}^{\infty}\lambda^n\bar{p}_i \tag{3.37}$$

$$\sum_{a_j\in\mathcal{A}}\rho(s_i,a_j)\leq\sum_{n=0}^{\infty}\lambda^n\bar{p}_i \tag{3.38}$$

$$\sum_{a_j\in\mathcal{A}}\rho(s_i,a_j)\leq\frac{1}{1-\lambda}\bar{p}_i$$

$$\bar{z}^T\rho\leq\frac{1}{1-\lambda}\bar{p}_i \tag{3.39}$$

Here $\bar{z}$ is a vector of zeros with ones in the positions corresponding to the occupational measures of state s_i. Eqn (3.35), (3.36) and (3.39) comprise the LP formulation for the constrained MDP or CMDP [73]. The optimal action for each state s_i is obtained from the

occupational measures as follows

$$\mathcal{T}(a_j|s_i) = \frac{\rho(s_i, a_j)^{\pi}_{s_0}}{\sum_{a_j} \rho(s_i, a_j)^{\pi}_{s_0}} \tag{3.40}$$

Note that Eqn (3.40) yields a probability distribution over the actions. In this work we select the action with the maximum probability.

3.7 CMDP for Takeoff

In this section we apply the above CMDP formulation to re-construct a resilient control override strategy that will enable us to obtain a policy that guarantees that the probability of entering a tail strike state remains below a selected threshold.

Without loss of generality, we impose the following probability constraint on the tail strike state $[P, p_4, q_7]$

$$\mathcal{T}([P, p_4, q_7]|[P, p_1, q_1]) = 0 \tag{3.41}$$

i.e. the probability of entering the tail strike state (p_4, q_7), starting from the initial state (p_1, q_1), with the pilot in control (P) is zero. Eqn (3.41) can be expressed as constraints on the occupational measures of state $[P, p_4, q_7]$ as illustrated in Eqn (3.39);

$$\bar{z}^T \rho = 0 \tag{3.42}$$

We can now solve the constrained MDP using the linear program described by Eqns (3.35)-(3.39). The resulting policy is shown in Fig 3.10. It can be seen that the new policy has no risk of tail strike (i.e. no (p_4, q_7) state). FSAM reliably overrides to prevent tail strike. Control is then transferred back to the pilot only after the aircraft no longer has the risk of a tail strike.

Fig 3.11 illustrates three takeoff scenarios with tail strike risk. We note here that the pilot is modeled as a human pilot transfer function [74] and is setup to apply excessive nose up elevator input during rotation to simulate a tail strike scenario. The red lines indicate the aircraft response to the excessive rotation command without FSAM augmentation. The constant pitch response (in red) at around 12 seconds indicates a tail strike. The blue lines indicate the response of the aircraft with the augmentation of the FSAM policy constructed in Section 3.5 using the unconstrained MDP (see Fig 3.9). Here, as illustrated previously (see Fig 3.9), FSAM overrides the pilot when it detects the excessive rotation input at around 10 seconds, but this MDP policy reverts control back to the pilot too soon. Subsequently, the continued application of the excessive nose up elevator input results in a

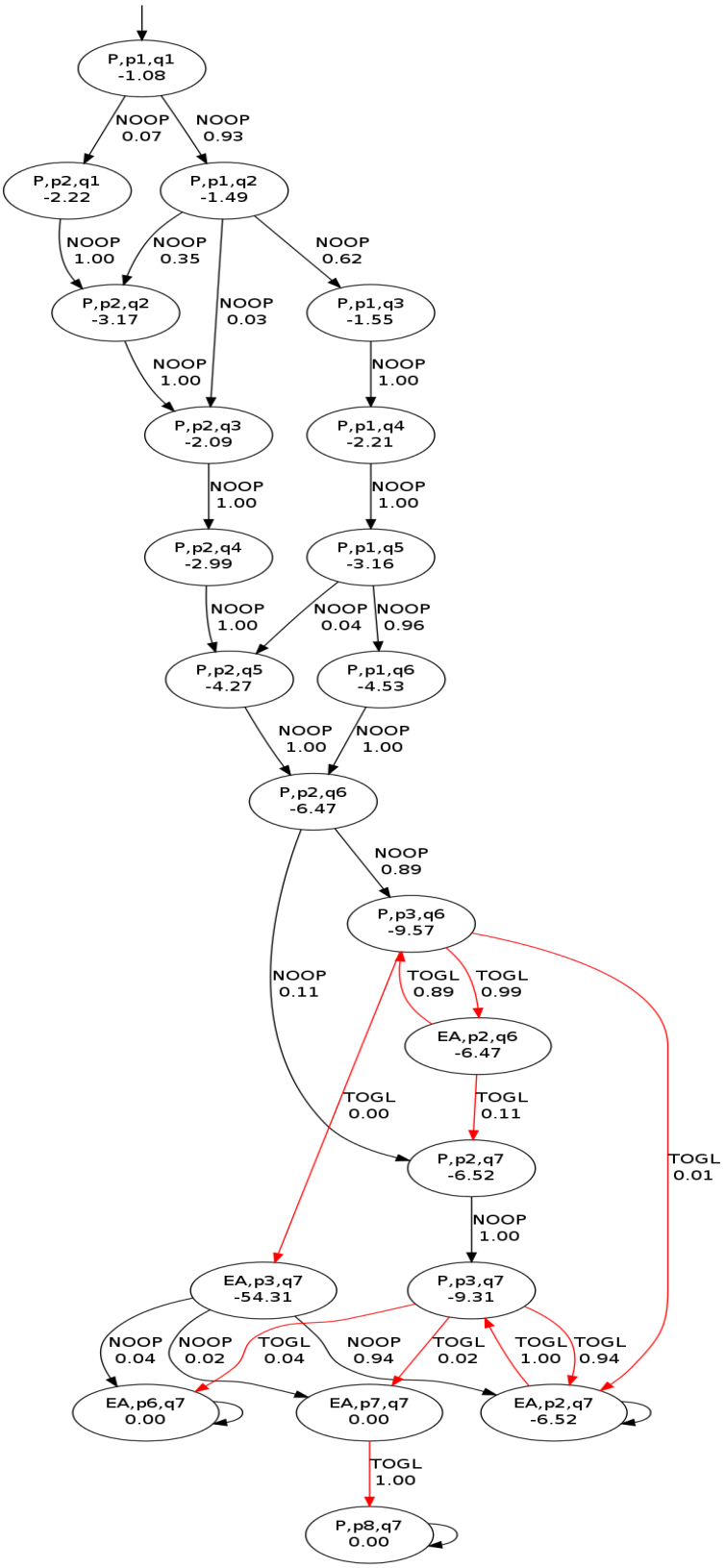

Figure 3.10: MDP with constraints

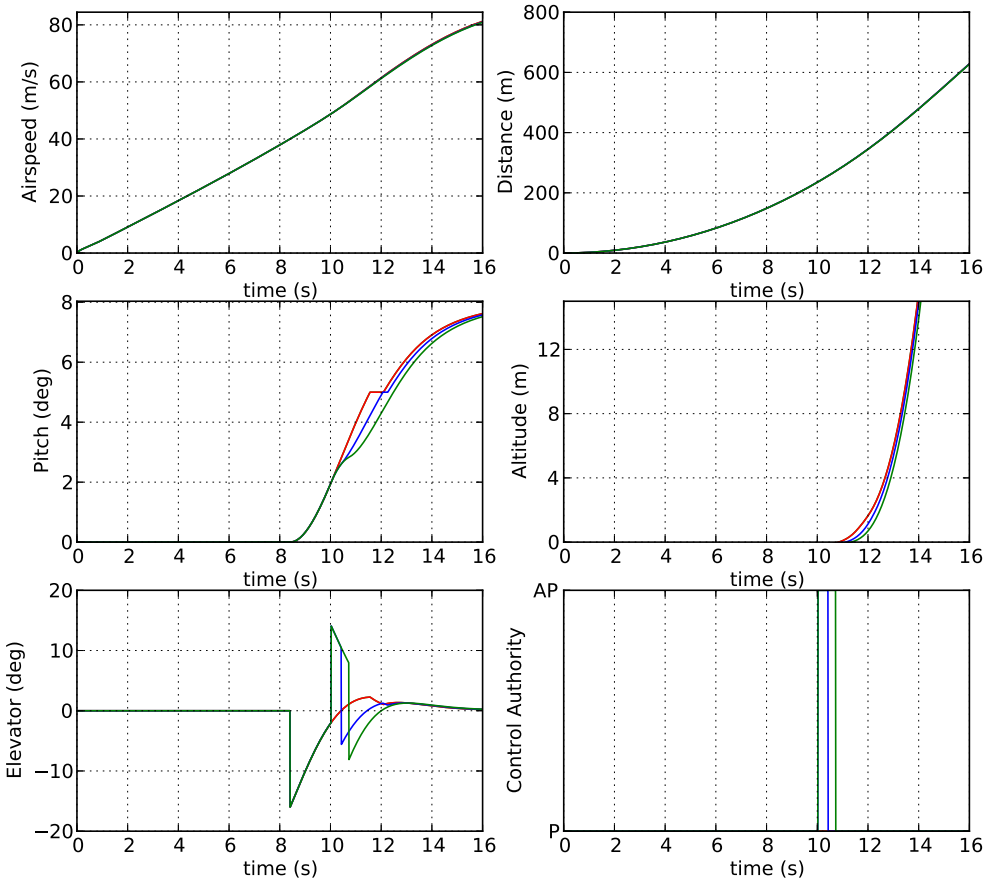

Figure 3.11: Tail strike scenarios with MDP and CMDP policies. No FSAM augmentation (red), FSAM with MDP policy (blue), FSAM with CMDP policy (green)

tail strike. The green lines indicate the aircraft's response to the MDP policy that was constructed using the CMDP approach described in Section 3.7. Here, FSAM reverts control to the pilot only after any imminent tail strike risk has been eliminated (see Fig 3.10).

3.8 Case Study

On 20th, March 2009 an Airbus A340 operated by Emirates Airlines, failed to takeoff safely from Melbourne Airport, Australia [75]. 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

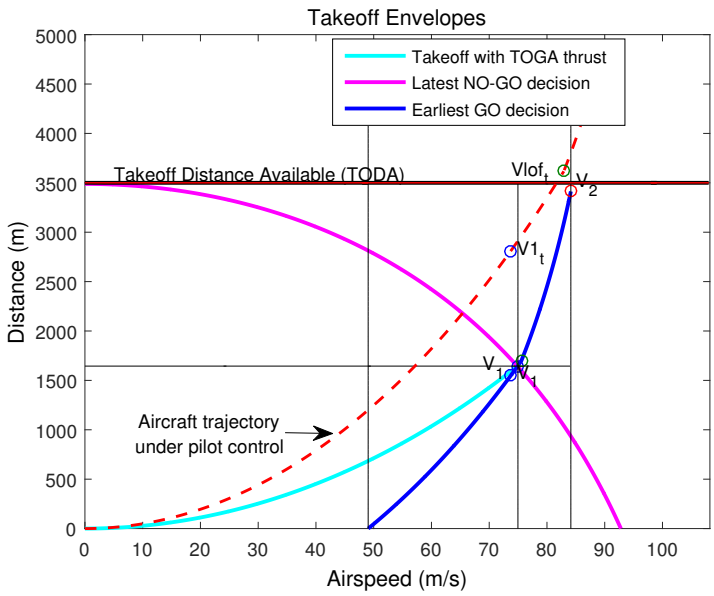

Figure 3.12: Trajectory of FL 407

takeoff roll and experienced a tail strike due to over-rotation. 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. Fig 3.12 illustrates the takeoff envelopes of the aircraft for the weight that was entered into the flight computer (262.9 tons). The dashed curve in Fig 3.12 indicates the actual aircraft trajectory (weighing 362.9 tons) from flight recorder data. Due to 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.

Let $\Delta W = W_{actual} - W_{input}$ represent the weight entry discrepancy. The scenario $\Delta W < 0$ poses no risk to takeoff safety since the thrust computed by the FMS would be more than the required takeoff thrust. Consequently, the aircraft can become airborne with a takeoff distance less than what is required. The scenario $0 \leq \Delta W \leq W_0$, where W_0 represents a weight difference for which the aircraft's acceleration still remains within the computed takeoff performance, also poses no risk to takeoff safety. However, for $\Delta W > W_0$, the aircraft's acceleration would yield trajectories that violated the required performance for a safe takeoff.

The above accident is used to illustrate the application of the policy developed using the MDP framework described in this work. Fig 3.13 illustrates the airspeed-position response with the MDP policy. As the accelerating aircraft enters $Q = q_9$, an unsafe region in the

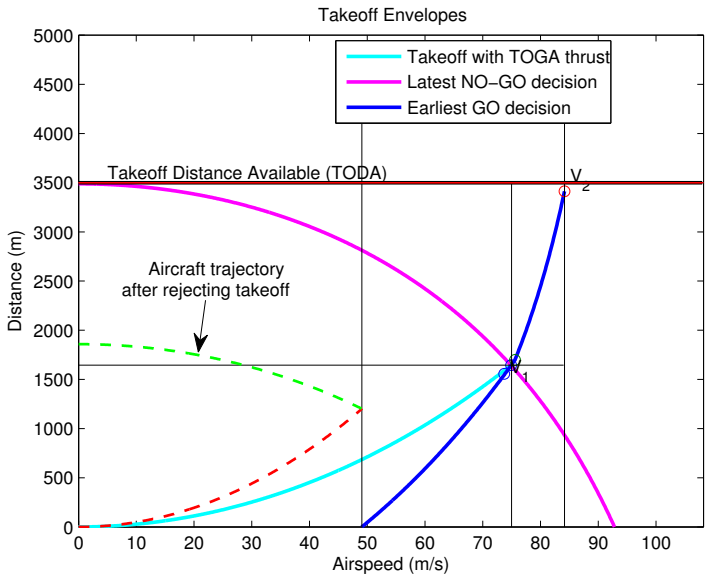

Figure 3.13: MDP policy applied to FL407

$V-x$ envelope, FSAM overrides the pilot with the envelope-aware controller which then rejects the takeoff. The aircraft then decelerates and stops safely in $Q = q_{17}$ well before the runway threshold. Fig 3.14 compares the aircraft states of Flight 407 modeled from data obtained from accident reports [75] and the simulated aircraft response to the takeoff MDP policy developed in this paper.

Note that the MDP policy used in this work depends on the partitions of the continuous states. The $V-x$ envelope is partitioned into 17 regions (see Fig 3.5). The computation of the envelope is discussed in Appendix C.1. For a given runway length and maximum available takeoff thrust, there is a minimum aircraft weight below which the $V-x$ envelope can no longer be partitioned into the 17 regions as indicated in Fig 3.5. The maximum takeoff weight that can be used for a given runway length is determined by the available thrust. Thus, to utilize the MDP formulation described in this work, the weight input to the flight control computer and the actual weight of the aircraft must remain within the policy's acceptable weight range.

3.9 Discussion

The above 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 as only one phase will be active at a time. Relaxing the assumptions on the MDP formulation

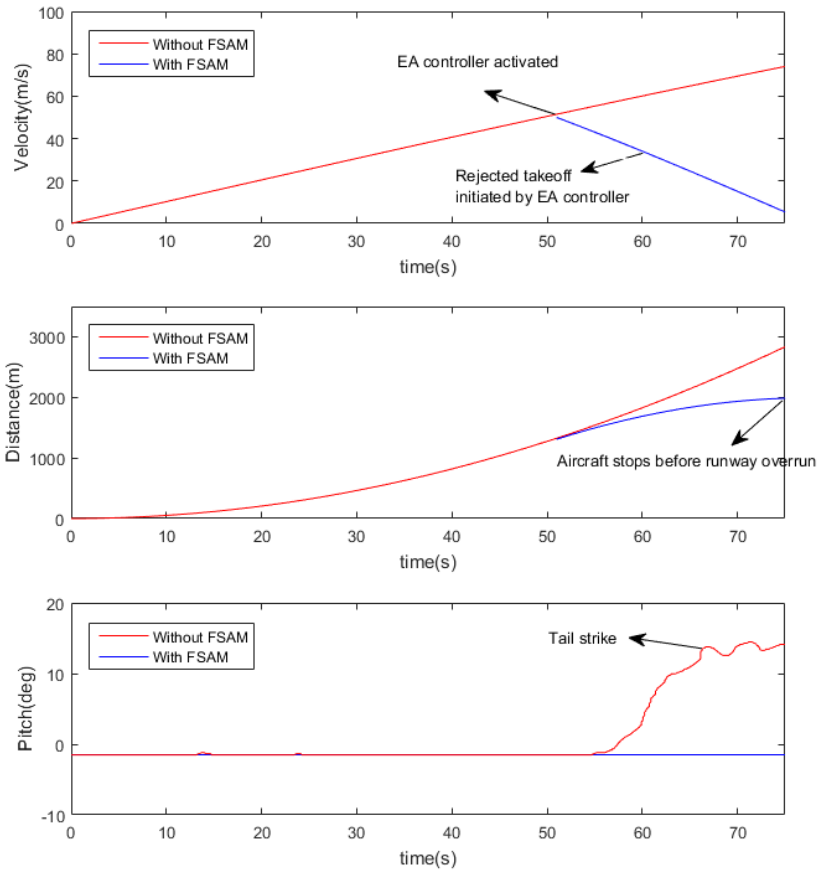

Figure 3.14: Comparison of flight trajectories of flight 407 versus simulated aircraft response with EA-FMS

will increase complexity substantially relative to the case study presented above. Conditional independence will be critical to exploit as will additional state-space abstractions.

This work exploited domain specific knowledge in the form of flight envelopes to construct partitions that eliminated the complexity associated with a typical grid based discretizations. It is further possible to evaluate policies (obtained using a partition scheme) and refine the partitions to construct less conservative policies [76, 77]. When no domain knowledge is available, methods proposed in [78, 79] can be used to learn a suitable partition that can alleviate the curse of dimensionality.

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 con-

trol 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, as flight data only represents one crew input case. Mining larger datasets, e.g., data from all of an airline's flights over a multi-year period, to determine statistically-significant state-space transition dynamics and probabilities.

The policies obtained from the MDP/CMDP formulations are stored in the form of look-up tables. Verifying large look-up table policies can be computationally-intensive. Use of model checking tools can facilitate verification of deterministic policies for large MDPs. Note that FSAM is only an overriding mechanism. Thus, if the available control authorities cannot mitigate a given LOC risk, the MDP policy can still 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 [63]. The availability of a simulation model for takeoff dynamics makes it possible to use reinforcement learning techniques such as Temporal Difference to solve the underlying MDP [63, 80]. The use of a Monte-Carlo tree search algorithm to solve the ideal MDP formulation in an online fashion is explored in following chapters.

3.10 Conclusion

This chapter 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 chapter 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 compu-

tationally tractable. A CMDP formulation eliminates the need to iteratively refine MDP policies by imposing probabilistic constraints on high risk states.

Chapter 2 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 also requires significant experience to ensure that the override directives are chosen appropriately. This chapter 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, an MDP formulation enables each policy to be optimized over uncertainties and generalized reward functions.

Despite the generality of the initial FSAM MDP formulation, this chapter makes several assumptions about pilot models, environment, and aircraft health in the takeoff case study. Extending the specific FSAM MDP models beyond these assumptions is essential to ensure 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.

CHAPTER 4

Application of FSAM to Icing Related Loss of Control

4.1 Introduction

Icing-related LOC situations are some of the most difficult cases to model and manage. Flight through atmospheric conditions conducive to icing can lead to accumulation of ice on the wings, tail surfaces and fuselage. Engine icing can also cause damage and even loss of thrust. Ice accumulation alters the shape of the airframe and disrupts airflow over the aircraft resulting in changes to its aerodynamic properties [81]. Consequently, ice accumulation increases weight and drag while decreasing lift. Wing icing also leads to a decrease in the stall angle of attack, while ice contamination of the horizontal stabilizers can result in tail plane stall [82]. Icing can also result in blockage of the pitot probes leading to erroneous airspeed measurements [83].

Several strategies can mitigate LOC risk due to icing. Prior to departure, de-icing fluids can be sprayed over the airframe to hinder ice accumulation. Aircraft anti-icing (e.g., wing heat) and de-icing (e.g., wing boot) systems can reduce or eliminate wing surface icing during flight. Current autopilot systems are equipped with warnings and envelope protection features that can help prevent stall [83, 84]. However, envelope protection logic is based on the nominal performance values such as the constant critical stall angle of attack applicable to clean surface conditions. With wing icing, the critical angle of attack decreases; currently no estimate of degraded stall angle of attack is provided to the pilot or autopilot. This renders the stall protection system ineffective for icing scenarios. Furthermore, the increase in drag due to icing requires the airplane to fly at higher airspeeds to produce the lift necessary for steady flight. Asymmetric ice accumulation can lead to upsets in roll control. Control surface effectiveness is also reduced due to icing. To prevent icing-related LOC, an ice protection system must first identify the changes in aerodynamics, performance and

control characteristics of the aircraft due to icing then ensure that both automation and crew warning systems incorporate these changes.

FSAM has previously been developed for high-risk LOC situations in which performance models are unchanged [85] or when a one-time performance reduction occurs [86]. This chapter focuses on developing an FSAM capability to ensure that an appropriate control authority is chosen to prevent or recover from icing, a potentially high-risk LOC scenario where performance can degrade progressively in flight. FSAM constantly monitors flight conditions to assess LOC risk, initially warning the flight crew when LOC risk exceeds a nominal threshold. If the crew does not respond with appropriate control actions in time to assure recovery, FSAM overrides with an Envelope Aware control law from EAFMS until the LOC risk is mitigated. FSAM is effectively a "watchdog" system providing LOC avoidance override for flight envelope protection in a general context. This chapter presents a Markov Decision Process (MDP) formulation that supports flight envelope protection during in-flight icing. Novel state features are based on a state-space abstraction that captures risk related to degradation of aircraft dynamics and controllability as a function of exposure to icing conditions.

The rest of the chapter is organized as follows. Section 4.2 provides a literature review. Section 4.3 formulates the FSAM MDP to address in-flight icing conditions. Section 4.4 illustrates the application of MDP policies with a case study involving an aircraft experiencing in-flight icing. Finally, Section 4.5 provides conclusions.

4.2 Literature Review

Several researchers have investigated and developed aircraft ice protection systems. The Smart Icing System (SIS) developed by Bragg et al [21] could sense ice accumulation based on its effect on aircraft stability and performance [87], adapt flight control laws to accommodate the degraded flight performance [88] and inform the crew to improve their situational awareness [89]. Gingras et al. [20] developed the Ice Contamination Envelope Protection (ICEPro) system. ICEPro focused exclusively on estimating degraded airplane performance to inform the flight crew about the degraded flight envelopes via cues presented in the flight displays. Lombaerts et al. [90, 91] developed an icing-related LOC prevention system that was conceptually similar to ICEPro and predicted envelope violations over a finite horizon.

The primary focus of all the work cited above was to develop robust identification and control techniques that can adapt to progressive aerodynamic performance changes due to ice accretion and thereby reduce the risk of LOC. All these systems relied on the crew to

enter control and flight plan changes that keep the aircraft within a safe operating region and navigate the aircraft to a safe landing or to an atmospheric volume with less or no icing. This work uses the capabilities of the Envelope-Aware FMS as an efficient augmentation to the conventional control authority (i.e. pilot). Consequently, this chapter contributes a novel decision making system that assesses risk associated with the current flight condition and selects the appropriate control authority to ensure the aircraft remains within the safe flight envelope. FSAM operates at a higher decision making level enabling flight plan and guidance input changes when flight envelope protection and warnings are not sufficient to mitigate risk along the crew prescribed flight path.

4.3 FSAM MDP Formulation for In-Flight Icing

4.3.1 State formulation

The ideal MDP state formulation must encode information about aircraft dynamics and controls, aircraft health, pilot and environment characteristics to make override decisions and reduce LOC risk [92]. Because the full FSAM formulation would be ideally described by a large suite of continuous and discrete variables, abstraction and decomposition are essential to manage complexity. At the top level, the MDP can be decomposed into a sequence of MDP formulations for the different phases of flight (i.e. takeoff, climb, cruise, descent and landing). Furthermore, compact abstractions of aircraft performance and flight envelopes pertinent to FSAM override decisions can be constructed to reduce state space complexity for each phase of flight.

This work focuses on icing as the primary hazard, assuming that aircraft health, pilot characteristics and environment features (except icing) do not pose additional risks with respect to flight safety and override. The compact FSAM MDP state representation for icing is defined as the tuple:

$$s = (\bar{V}, \bar{A}, \bar{\Theta}, \bar{\Phi}, \bar{H}, \bar{T}, \bar{F}, \bar{I}, \bar{S}, \bar{M}) \tag{4.1}$$

Here $\bar{V} = \{\bar{v}_1, \ldots, \bar{v}_6\}$ represents intervals of aircraft airspeed V. Each $\bar{v}_i$ represents a partition defined over controllable airspeeds in the interval $[V_{max}, V_{min}]$ as illustrated in Eqn (4.2) and Fig 4.1. $\bar{v}_1$ is a high risk state due to the possibility of aerodynamic stall while $\bar{v}_6$ poses high structural damage risk due to high aero-structural loads. $\bar{V}$ encodes information about the proximity to airspeed envelope boundaries with $\Delta V = V_{max} - V_{min}$ and

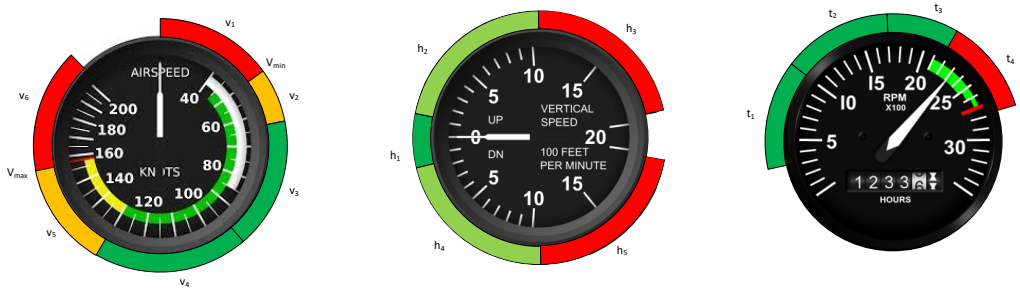

Figure 4.1: Partitions in Airspeed (left), Vertical speed (center), Thrust (right) [3–5]

$0 < n_{v_1} < n_{v_2} < n_{v_3} < 1.$

$$
\begin{aligned}
\bar{v}_1 &= \{V \mid V < V_{min}\} \\
\bar{v}_2 &= \{V \mid 0 < V - V_{min} \leq n_{v_1} \Delta V\} \\
\bar{v}_3 &= \{V \mid n_{v_1} \Delta V < V - V_{min} \leq n_{v_2} \Delta V\} \\
\bar{v}_4 &= \{V \mid n_{v_2} \Delta V < V - V_{min} \leq n_{v_3} \Delta V\} \\
\bar{v}_5 &= \{V \mid n_{v_3} \Delta V < V - V_{min} \leq \Delta V\} \\
\bar{v}_6 &= \{V \mid V > V_{max}\}
\end{aligned}
\tag{4.2}
$$

The graphical representations of the state partitions in Fig 4.1 are illustrated with instruments found in a typical Cessna 172 type aircraft. While values are specific to the C-172, the partition set generalizes to any fixed-wing aircraft type. Note that n_{v_i} can be defined based on a reachable set analysis (see references [50, 56, 86]).

$\bar{A} = \{\bar{\alpha}_1, \bar{\alpha}_2, \bar{\alpha}_3\}$ represents partitions of the adverse aerodynamic envelope boundaries introduced by Wilborn et al. [93]. It encodes information about proximity to a stall condition. Figure 4.2 illustrates the partitions of the adverse aerodynamic envelope. Note that $\bar{\alpha}_3$ represents a high risk state where aerodynamic stall is highly likely. Let $X = [\alpha_m, \beta_m]$ represent a vector whose components are the normalized angle of attack and sideslip angles [93]. Adverse aerodynamic envelope abstractions are formally defined as follows:

$$
\begin{aligned}
\bar{\alpha}_1 &= \{X \mid A_\alpha X \leq B_{\alpha 1}\} \\
\bar{\alpha}_2 &= \{X \mid (A_\alpha X \leq B_{\alpha 2}) \setminus \bar{\alpha}_1\} \\
\bar{\alpha}_3 &= \{X \mid \mathbb{R}^2 \setminus (A_\alpha X \leq B_{\alpha 2})\}
\end{aligned}
$$

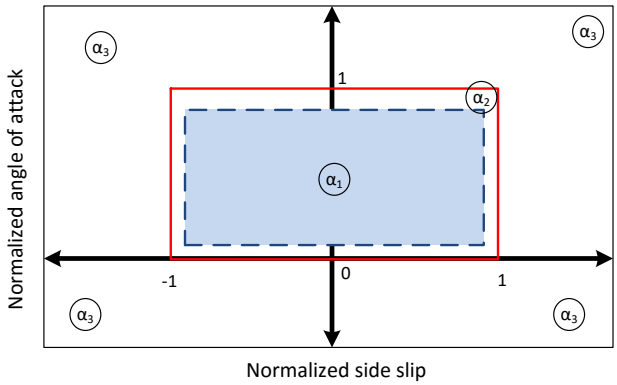

Figure 4.2: Adverse aerodynamic envelope partitions

where $A_\alpha, B_{\alpha 1}, B_{\alpha 2}$ are defined below. n_{α_1,α_2} are positive scalars and $0 < n_{\alpha_1} < n_{\alpha_2} < 1$.[1]

$$A_\alpha = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, B_{\alpha 1} = \begin{bmatrix} n_{\alpha_2} \\ n_{\alpha_1} \\ n_{\alpha_2} \\ n_{\alpha_2} \end{bmatrix}, B_{\alpha 2} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$\bar{\Theta} \in \{\bar{\theta}_1, \bar{\theta}_2, \bar{\theta}_3, \bar{\theta}_4, \bar{\theta}_5, \bar{\theta}_6\}$ is a compact representation of aircraft pitch θ, pitch rate q and elevator control δ_e. Specifically, $\bar{\Theta}$ is a discretization of the dynamic pitch control envelope introduced by Wilborn et al. [93]. The dynamic pitch θ' is defined as $\theta + q$. $\bar{\Theta}$ is illustrated in Fig 4.3. Let $X = [\theta', \delta_e]$ represent a vector whose components are the dynamic pitch attitude and elevator deflection [93]. Dynamic pitch control envelope abstractions are formally defined as follows:

$$\begin{aligned}
\bar{\theta}_1 &= \{X \mid A_\theta X \leq B_{\theta 1}\} \\
\bar{\theta}_2 &= \{X \mid (A_\theta X \leq B_{\theta 2}) \setminus \bar{\theta}_1\} \\
\bar{\theta}_3 &= \{X \mid Q_1 \setminus \bar{\theta}_2\} \\
\bar{\theta}_4 &= \{X \mid Q_2 \setminus \bar{\theta}_2\} \\
\bar{\theta}_5 &= \{X \mid Q_3 \setminus \bar{\theta}_2\} \\
\bar{\theta}_6 &= \{X \mid Q_4 \setminus \bar{\theta}_2\}
\end{aligned}$$

where $Q_i, i = 1,2,3,4$ denote the first, second, third and fourth quadrants in $\mathbb{R}^2$. $A_\theta, B_{\theta 1}, B_{\theta 2}$

[1] Suitable values for $n_{\alpha_1}, n_{\alpha_2}$ can be defined similar to n_{v_i}.

are defined below:

$$A_\theta = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ -m_1 & 1 \\ m_2 & -1 \end{bmatrix}, B_{\theta 1} = \begin{bmatrix} \delta_{e_{max}} \\ \theta_{max} \\ \delta_{e_{min}} \\ \theta_{min} \\ \theta_{max} \\ \theta_{min} \end{bmatrix}, B_{\theta 2} = n_\theta B_{\theta 1}$$

where m_1, m_2 are the slopes of the envelope boundaries in Q_2 and Q_4 respectively (see Figure 4.3). $\delta_{e_{min}}, \delta_{e_{max}}$ represent elevator saturation limits while $\theta_{min}, \theta_{max}$ represent safe pitch attitude limits. n_θ is a positive scalar and $0 < n_\theta < 1$.[2]

$\bar{\Phi} \in \{\bar{\phi}_1, \ldots, \bar{\phi}_6\}$ is a compact representation of aircraft roll ϕ, roll rate p and aileron control δ_a. Dynamic roll ϕ' is defined as $\phi + p$. $\bar{\Phi}$ is a discretization of the dynamic roll control envelope as specified in Wilborn et al. [93] $\bar{\phi}_1, \bar{\phi}_2$ represent safe operating envelope regions. The dynamic roll control envelope partitions are defined as follows:

$$\begin{aligned} \bar{\phi}_1 &= \{X \mid A_\phi X \leq B_{\phi 1}\} \\ \bar{\phi}_2 &= \{X \mid (A_\phi X \leq B_{\phi 2}) \setminus \bar{\phi}_1\} \\ \bar{\phi}_3 &= \{X \mid Q_1 \setminus \bar{\phi}_2\} \\ \bar{\phi}_4 &= \{X \mid Q_2 \setminus \bar{\phi}_2\} \\ \bar{\phi}_5 &= \{X \mid Q_3 \setminus \bar{\phi}_2\} \\ \bar{\phi}_6 &= \{X \mid Q_4 \setminus \bar{\phi}_2\} \end{aligned}$$

$A_\phi, B_{\phi 1}, B_{\phi 2}$ are defined as follows:

$$A_\phi = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ -m_1 & 1 \\ m_2 & -1 \end{bmatrix}, B_{\phi 1} = \begin{bmatrix} \delta_{a_{max}} \\ \phi_{max} \\ \delta_{a_{min}} \\ \phi_{min} \\ \phi_{max} \\ \phi_{min} \end{bmatrix}, B_{\phi 2} = n_\phi B_{\phi 1}$$

where m_1, m_2 are the slopes of the envelope boundaries in Q_2 and Q_4 respectively (see Figure 4.3). $\delta_{a_{min}}, \delta_{a_{max}}$ represent aileron saturation limits while ϕ_{min}, ϕ_{max} represent safe

[2] A suitable value for n_θ can be defined similar to n_{v_i}.

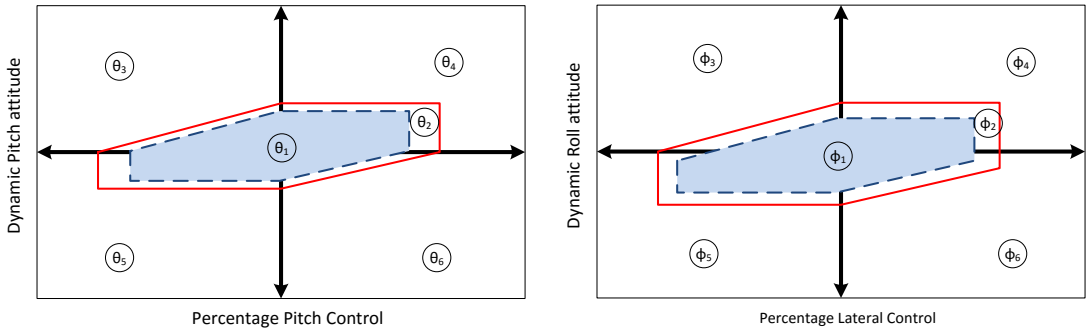

Figure 4.3: Dynamic pitch and roll control envelope partitions

roll attitude limits. n_ϕ is a positive scalar and $0 < n_\phi < 1$.[3]

$\bar{H} = \{\bar{h}_1, \ldots, \bar{h}_5\}$ partitions the aircraft's vertical speed as shown in Fig 4.1. Let $\dot{h}_{max}, \dot{h}_{min}$ denote the maximum and minimum climb rates for safe operation and let $\dot{h}_0 > 0$ denote an appropriate value of climb rate close to zero. Partition intervals are defined as follows:

$$\bar{h}_1 = \{\dot{h} \mid |\dot{h}| < \dot{h}_0 \text{ ft/min}\}$$
$$\bar{h}_2 = \{\dot{h} \mid \dot{h}_0 \text{ ft/min} \leq \dot{h} < \dot{h}_{max} \text{ ft/min}\}$$
$$\bar{h}_3 = \{\dot{h} \mid \dot{h} \geq \dot{h}_{max} \text{ ft/min}\}$$
$$\bar{h}_4 = \{\dot{h} \mid -\dot{h}_0 \text{ ft/min} \geq \dot{h} > \dot{h}_{min} \text{ ft/min}\}$$
$$\bar{h}_5 = \{\dot{h} \mid \dot{h} \leq \dot{h}_{min} \text{ ft/min}\}$$

$\bar{T} \in \{\bar{t}_1, \bar{t}_2, \bar{t}_3, \bar{t}_4\}$ denote partitions of the thrust control input space as shown in Fig 4.1. T_{max} denotes the maximum thrust output and $0 < n_{t_1} < n_{t_2} < n_{t_3} < 1$. The partitions are defined as follows [4]:

$$\bar{t}_1 = \{T \mid 0 \leq T \leq n_{t_1} T_{max}\}$$
$$\bar{t}_2 = \{T \mid n_{t_1} T_{max} < T \leq n_{t_2} T_{max}\}$$
$$\bar{t}_3 = \{T \mid n_{t_2} T_{max} < T \leq n_{t_3} T_{max}\}$$
$$\bar{t}_4 = \{T \mid n_{t_3} T_{max} < T \leq T_{max}\}$$

$\bar{F}$ represents current flight plan information with triple (F_c, F_t, F_s) that characterizes

[3] A suitable value for n_ϕ can be defined similar to n_{v_i}

[4] Values for n_{t_i} can be chosen to denote partitions of power setting ranges corresponding to takeoff, climb/cruise, approach and idle

Table 4.1: Flight plan state composition

$\bar{f}_1$	(level,straight,slow)	$\bar{f}_2$	(level,straight,med)	$\bar{f}_3$	(level,straight,fast)
$\bar{f}_4$	(level,turn,slow)	$\bar{f}_5$	(level,turn,med)	$\bar{f}_6$	(level,turn,fast)
$\bar{f}_7$	(climb,straight,slow)	$\bar{f}_8$	(climb,straight,med)	$\bar{f}_9$	(climb,straight,fast)
$\bar{f}_{10}$	(climb,turn,slow)	$\bar{f}_{11}$	(climb,turn,med)	$\bar{f}_{12}$	(climb,turn,fast)
$\bar{f}_{13}$	(descent,straight,slow)	$\bar{f}_{14}$	(descent,straight,med)	$\bar{f}_{15}$	(descent,straight,fast)
$\bar{f}_{16}$	(descent,turn,slow)	$\bar{f}_{17}$	(descent,turn,med)	$\bar{f}_{18}$	(descent,turn,fast)

climb, turn and airspeed.

$$F_c \in \{\text{level,climb,descent}\}$$
$$F_t \in \{\text{straight,turn}\}$$
$$F_s \in \{\text{slow,med,fast}\}$$

Values 'level','climb' and 'descent' are defined as longitudinal flight conditions with zero, positive and negative climb rates, respectively. 'Straight' and 'turn' are defined as flight conditions with zero and non-zero turn rates, respectively. 'Slow' is defined as the set of flight conditions where $\bar{V} \in \{\bar{v}_1, \bar{v}_2\}$, 'med' includes flight states with $\bar{V} \in \{\bar{v}_3, \bar{v}_4\}$ and 'fast' states have $\bar{V} \in \{\bar{v}_5, \bar{v}_6\}$. Thus, $\bar{F}$ is abstracted into $\{\bar{f}_1, \ldots, \bar{f}_{18}\}$ as shown in Table 4.1.

$\bar{I}$ encapsulates information about predicted exposure to icing conditions based on a given flight plan and expected atmospheric (icing, wind) conditions. Let t_{pte} be defined as the predicted time of exposure to icing. The critical exposure time t_{critical} is defined as the duration beyond which further exposure to icing is most likely to result in stall conditions. Note that the critical exposure time depends on several factors such as icing severity, maximum thrust available, commanded airspeed during icing conditions, usage of deicing fluids prior to takeoff, and capacity of the anti-icing system [94]. $\bar{I}$ is defined as the tuple $(n_{\text{pte}}, n_{\text{ice}})$ where $n_{\text{pte}} \in \{t_{\text{ pte}} = 0,\ 0 < t_{\text{pte}} < t_{\text{critical}},\ t_{\text{pte}} > t_{\text{critical}}\}$ denotes partitions in the predicted time of exposure and $n_{\text{ice}} \in \{0, 1\}$ where 0 denotes flight outside icing clouds and 1 denotes that the aircraft is flying in icing conditions. $\bar{I}$ is compactly represented as $\{\bar{i}_0, \bar{i}_1, \bar{i}_2, \bar{i}_3, \bar{i}_4\}$ as shown in Table 4.2. Figure 4.4 graphically illustrates these states.

$\bar{M} \in \{P, EA\}$ represents the current control mode. Here P denotes that the pilot is in control while EA indicates envelope-aware control. $\bar{S} \in \{P_s, EA_s\}$ represents a mode select switch with which the pilot can request pilot control authority be maintained/restored (P_s) or else can manually engage or maintain Envelope-Aware control (EA_s).

Note that the airspeed, angle of attack, pitch and roll partitions described above are parameterized in terms of envelope boundaries. Consequently, as these parameters are

Table 4.2: Icing intensity state abstraction

$\bar{i}_0$	($0 < t_{\text{pte}} < t_{\text{critical}}$, 0)
$\bar{i}_1$	($t_{\text{pte}} \geq t_{\text{critical}}$, 0)
$\bar{i}_2$	($0 < t_{\text{pte}} < t_{\text{critical}}$, 1)
$\bar{i}_3$	($t_{\text{pte}} \geq t_{\text{critical}}$, 1)
$\bar{i}_4$	($t_{\text{pte}} = 0$, 0)

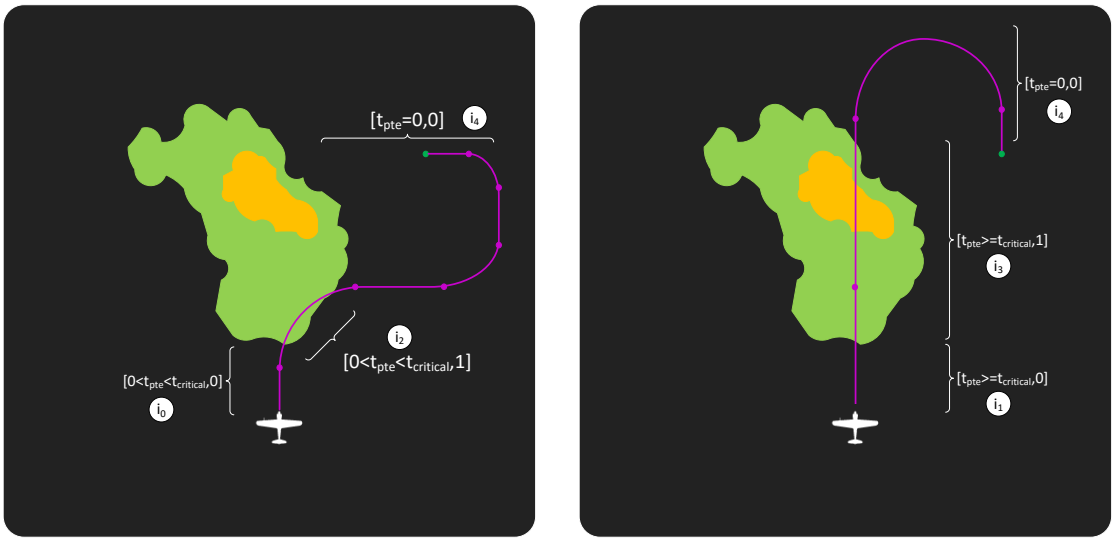

Figure 4.4: Abstraction for icing intensity based on available flight plan

updated by envelope estimation, these state representations capture the evolving risks due to icing.

4.3.2 Action Set

FSAM is a high-level watchdog that passively monitors for LOC risk and overrides only when necessary to avoid LOC. As explained in the previous chapter, there are two actions: NOOP (No Operation) and TOGL (Toggle). FSAM remains passive by selecting NOOP. TOGL is selected if switching control authority is necessary to mitigate risk or restore control authority to the crew and nominal automation after LOC risk has been adequately mitigated.

4.3.3 Transition Probabilities

The transition probabilities can be modeled as described in Chapter 3 (see Section 3.3.3). For a given mode, the transition probabilities are specified as follows:

$$\begin{aligned}\mathcal{T}_M(s_j|s_i) &= \mathcal{P}_1(\bar{V}_j \mid \bar{V}_i,\bar{A}_i,\bar{\Theta}_i,\bar{\Phi}_i,\bar{H}_i,\bar{T}_i,\bar{F}_i,\bar{I}_i,\bar{M}) \times \mathcal{P}_2(\bar{A}_j \mid \bar{V}_i,\bar{A}_i,\bar{\Theta}_i,\bar{\Phi}_i,\bar{H}_i,\bar{T}_i,\bar{F}_i,\bar{I}_i,\bar{M}) \\ &\times \mathcal{P}_3(\bar{\Theta}_j \mid \bar{V}_i,\bar{A}_i,\bar{\Theta}_i,\bar{\Phi}_i,\bar{H}_i,\bar{T}_i,\bar{F}_i,\bar{I}_i,\bar{M}) \times \mathcal{P}_4(\bar{\Phi}_j \mid \bar{V}_i,\bar{A}_i,\bar{\Theta}_i,\bar{\Phi}_i,\bar{H}_i,\bar{T}_i,\bar{F}_i,\bar{I}_i,\bar{M}) \\ &\times \mathcal{P}_5(\bar{H}_j \mid \bar{V}_i,\bar{A}_i,\bar{\Theta}_i,\bar{\Phi}_i,\bar{H}_i,\bar{T}_i,\bar{F}_i,\bar{I}_i,\bar{M}) \times \mathcal{P}_6(\bar{T}_j \mid \bar{V}_i,\bar{A}_i,\bar{\Theta}_i,\bar{\Phi}_i,\bar{H}_i,\bar{T}_i,\bar{F}_i,\bar{I}_i,\bar{M}) \\ &\times \mathcal{P}_7(\bar{F}_j \mid \bar{V}_i,\bar{A}_i,\bar{\Theta}_i,\bar{\Phi}_i,\bar{H}_i,\bar{T}_i,\bar{F}_i,\bar{I}_i,\bar{M}) \times \mathcal{P}_8(\bar{I}_j \mid \bar{F}_i,\bar{I}_i,\bar{M}) \times \mathcal{P}_9(\bar{S}_j \mid \bar{S}_i,\bar{M})\end{aligned} \tag{4.3}$$

The factored representation of $\mathcal{T}_M$ in Eqn 4.3 provides flexibility since it facilitates the incorporation of data from several sources. Distributions $\mathcal{P}_1,\ldots,\mathcal{P}_9$ can be estimated via one or more methods such as Monte Carlo sampling of physics-based models, flight data mining and human subject experiments. With $\mathcal{T}_M$, the transition probability for each action is expressed as described in Eqn 3.23-3.24, Chapter 3.

4.3.4 Reward formulation

FSAM's goal is to ensure that the aircraft avoids states with high LOC risk while minimizing authority shifts away from pilot-designated mode $\bar{S}$. Consequently, an additive reward $\mathcal{R}(s,a) = \sum_i w_i \mathcal{R}_i$ is defined where the $\mathcal{R}_i$'s penalize unsafe states and inconsistent authority switches while the w_i are weighting parameters.

$$\mathcal{R}_1 = \begin{cases} -1 \; if \; \bar{V} \in \{\bar{v}_1, \bar{v}_6\} \\ 0 \; otherwise \end{cases} \qquad \mathcal{R}_2 = \begin{cases} -1 \; if \; \bar{A} \in \{\bar{\alpha}_3\} \\ 0 \; otherwise \end{cases} \tag{4.4}$$

$$\mathcal{R}_3 = \begin{cases} -1 \; if \; \bar{\Theta} \in \{\bar{\theta}_3, \bar{\theta}_4, \bar{\theta}_5, \bar{\theta}_6\} \\ 0 \; otherwise \end{cases} \qquad \mathcal{R}_4 = \begin{cases} -1 \; if \; \bar{\Phi} \in \{\bar{\phi}_3, \bar{\phi}_4, \bar{\phi}_5, \bar{\phi}_6\} \\ 0 \; otherwise \end{cases} \tag{4.5}$$

$$\mathcal{R}_5 = \begin{cases} -1 \; if \; \bar{H} \in \{\bar{h}_5\} \\ 0 \; otherwise \end{cases} \tag{4.6}$$

$$\mathcal{R}_6 = \begin{cases} -o_1 \; if \; M = P \wedge \bar{S} = P_s \wedge a = TOGL \\ -o_2 \; if \; M = P \wedge \bar{S} = EA_s \wedge a = NOOP \\ -o_3 \; if \; M = EA \wedge \bar{S} = P_s \wedge a = NOOP \\ -o_4 \; if \; M = EA \wedge \bar{S} = EA_s \wedge a = TOGL \\ 0 \; otherwise \end{cases} \tag{4.7}$$

Above, $o_{i=1,\ldots,4} \in [0,1]$. Note that setting $(o_1, o_2, o_3, o_4) = (1,0,1,0)$ only discourages EA mode when the crew selects P_s. Persistence in envelope-aware control mode might be penalized to encourage transfer of authority to the crew once any high-risk condition prompting FSAM TOGL to $M = EA$ is mitigated. $(o_1, o_2, o_3, o_4) = (1,1,1,1)$ encourages the policy to satisfy the crew's mode select request. For the case study discussed in this work, the following parameters are chosen: $w_1 = 100$, $w_2 = 100$, $w_3 = 50$, $w_4 = 50$, $w_5 = 100$, $w_6 = 10$ and $o_1 = o_2 = o_3 = o_4 = 1$.

4.3.5 FSAM MDP policy

To find the optimal policy for the above MDP formulation, the distributions $\mathcal{P}_1, \ldots, \mathcal{P}_9$ must be estimated a priori. $\mathcal{P}_1, \ldots, \mathcal{P}_7$ and $\mathcal{P}_9$ can be estimated as described in Section 4.3.3. Note that $\mathcal{P}_8$ describes the transitions in predicted exposure to icing $\bar{I}$ which depend on the current flight plan. The current flight plan can change due to air-traffic control constraints, environmental constraints, crew preferences and on-board anomalies. Thus, estimating transition probabilities with respect to $\bar{I}$ a priori can be hard. To overcome this difficulty, we adopt a hierarchical solution approach. First, we obtain the optimal values $\mathcal{V}^*(s)$ for each state s using Value Iteration [62] assuming that the feature $\bar{I}$ remains constant. Later, onboard the aircraft, after changes to the flight plan are known, the corresponding probabilities for $\bar{I}$ are updated appropriately and Value Iteration is executed online using $\mathcal{V}^*$ as the initial guess. Since only a small portion of the transition matrix is updated online, using $\mathcal{V}^*$ significantly reduces the number of iterations required to converge to the optimal solution. [5] Alternately, the values $\mathcal{V}^*$ computed offline can be utilized online to choose an action at each state according to the following rule:

$$\pi^*(s_i) = \underset{a}{\text{argmax}} \Big\{ \mathcal{Q}(s_i, a = NOOP), \mathcal{Q}(s_i, a = TOGL) \Big\} \tag{4.8}$$

[5] Other online MDP algorithms such as real-time dynamic programming and asynchronous value iteration can also be used.

where

$$\mathcal{Q}(s_i, a) = \mathcal{R}(s_i, a) + \gamma \sum_{s_k} \mathcal{T}_M(s_k|s_j)\mathcal{V}^*(s_k) \tag{4.9}$$

Note that Eqn (4.8) is a greedy policy that uses the transition probabilities updated online with values $\mathcal{V}^*$ computed offline. In Eqn (4.9), $M \in \{P, EA\}$ in $\mathcal{T}_M$ is specified by state s_j. $s_j = s_i$ if the policy prescribes $a = NOOP$ in state s_i while M is toggled in s_j relative to s_i if $a = TOGL$. Note that if switching control authorities with $TOGL$ results in a new flight plan with a different predicted exposure time to icing (which can be estimated online), s_j allows for instantaneous changes in icing intensity feature values relative to s_i.

4.4 In-flight Icing Case Study

Consider an aircraft on approach to Buffalo Niagara International Airport[6] (KBUF) Runway 23 as shown in Fig 4.5. The flight plan progresses nominally as follows. From Initial Approach Fix (IAF) **SUSKE** to **BUFST** the aircraft maintains steady level flight at a medium speed ($\bar{F} = \bar{f}_2$). From **BUFST** onwards, the aircraft starts a straight descent at medium speed ($\bar{F} = \bar{f}_{14}$). At **ZADUM** the aircraft starts a descending right turn ($\bar{F} = \bar{f}_{17}$) toward Final Approach Fix (FAF) **BIILS** and then continues with a straight descent while decelerating ($\bar{F} = \bar{f}_{13}$) to a nominal touchdown speed. The state transition probabilities used in this case study are described in Appendix D. To succinctly describe and compute transition probability distributions, this work assumes that the angle of attack, side-slip and dynamic pitch attitude values always remain within the safe operating envelope (i.e. $\bar{A} = \bar{\alpha}_1, \bar{\Theta} = \bar{\theta}_1$). The mode select switch is always assumed set to request pilot authority (i.e. $\bar{S} = P_s$). With the state transition probabilities described in Appendix D and the reward formulation described in Section 4.3.4, the optimal policy for the MDP is obtained using value iteration.

4.4.1 Flight without icing conditions

Consider the case where $\bar{I} = \bar{i}_4$, i.e. predicted time of exposure $t_{pte} = 0$ and the aircraft is free from icing conditions $n_{ice} = 0$. Suppose the aircraft is following the flight segment between **SUSKE** and **BUFST** prescribed by straight and steady level flight. For this flight segment, consider the policy actions for MDP states $s = [\bar{V}, C_1, M] \in \mathcal{S}$ where $C_1 = [\bar{\alpha}_1, \bar{\theta}_1, \bar{\phi}_1, \bar{h}_1, \bar{t}_2, \bar{i}_4, \bar{f}_2, P_s]$ represents features presumed to remain constant. As an ex-

[6] This case study is motivated by the crash of Colgan Air Flight 3407. [95]

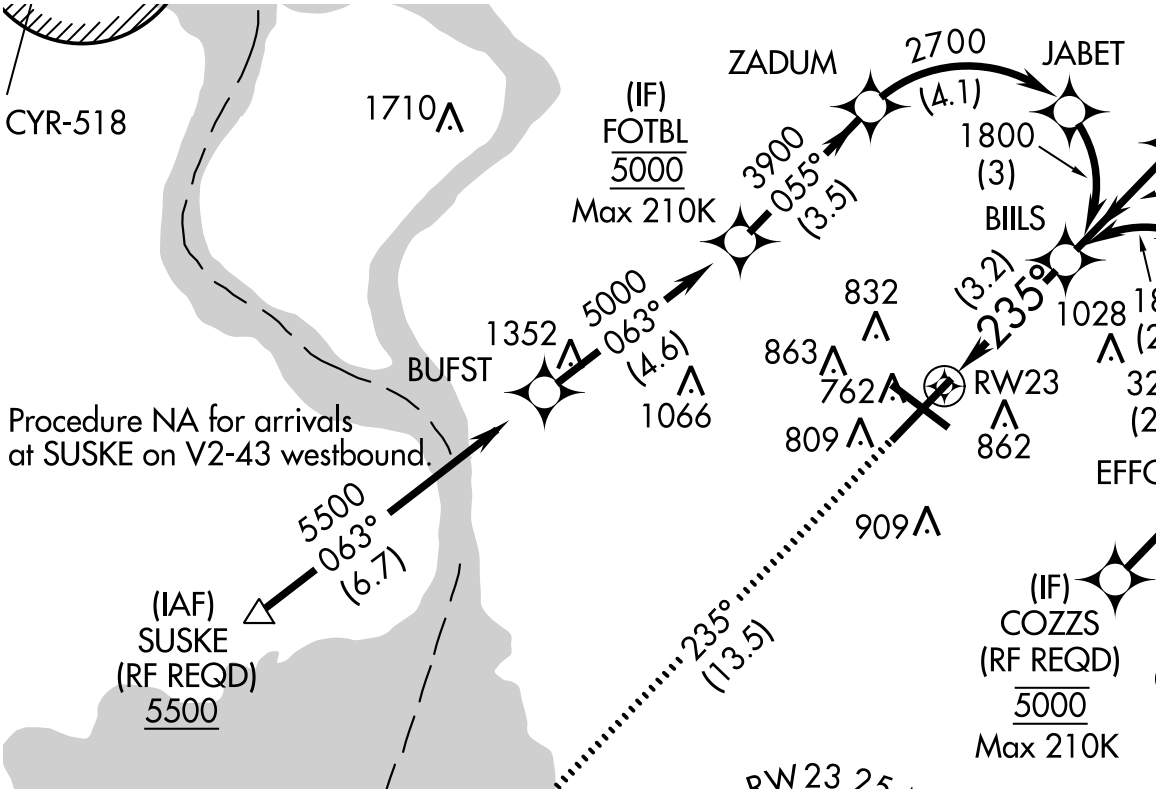

Figure 4.5: Approach flight plan [source: www.airnav.com]

ample, the optimal policy action for state $s_1 = [\bar{V} = \bar{v}_3, C_1, M = P]$ is computed by evaluating its state-action utility:

$$\mathcal{Q}(s_1, NOOP) = \mathcal{R}(s_1, NOOP) + \gamma \sum_{s_k \in \mathcal{S}} \mathcal{T}_P(s_k|s_1)\mathcal{V}^*(s_k) = -0.57 \tag{4.10}$$

$$\mathcal{Q}(s_1, TOGL) = \mathcal{R}(s_1, TOGL) + \gamma \sum_{s_k \in \mathcal{S}} \mathcal{T}_{EA}(s_k|s_2)\mathcal{V}^*(s_k) = -10.40 \tag{4.11}$$

where $s_2 = [\bar{V} = \bar{v}_3, C_1, M = EA]$. From these calculations the optimal action in s_1 is $NOOP$. Table 4.3 illustrates the final state-action values for a subset of the other airspeed states. Note that when the flight crew has control in $\bar{V} \in \{\bar{v}_3, \bar{v}_4\}$, the policy selects $NOOP$. When airspeed is $\bar{V} = \bar{v}_2$, FSAM elects $TOGL$ to $M = EA$ because selecting $NOOP$ results in a very low state-action utility due to the relatively high likelihood of entering a high risk state ($\bar{V} = \bar{v}_1$). Consequently, the policy favors $TOGL$ when $\bar{V} = \bar{v}_2$ because the EA controller has a higher probability of transitioning to $\bar{v}_3$ and zero probability of transitioning to $\bar{v}_1$ (see Table D.1) hence eliminating a stall risk. Note that control is given back to the pilot when $\bar{V} = \bar{v}_3$. The policy behavior for states $s = [\bar{V}, C_1, M] \in \mathcal{S}$ is summarized in Fig 4.6.

Now consider the case where the aircraft is at **ZADUM** and is initiating a descending

Table 4.3: State action utilities for $s = [\bar{V}, C_1, M]$. Left $M = P$, Right $M = EA$

$s \in \mathcal{S}$	$\mathcal{Q}(s, NOOP)$	$\mathcal{Q}(s, TOGL)$
$[\bar{v}_1, P]$	-125.30	**-112.17**
$[\bar{v}_2, P]$	-16.35	**-10.85**
$[\bar{v}_3, P]$	**-0.57**	-10.40
$[\bar{v}_4, P]$	**-0.19**	-10.19

$s \in \mathcal{S}$	$\mathcal{Q}(s, NOOP)$	$\mathcal{Q}(s, TOGL)$
$[\bar{v}_1, EA]$	**-107.17**	-125.30
$[\bar{v}_2, EA]$	**-5.85**	-16.35
$[\bar{v}_3, EA]$	-5.40	**-0.57**
$[\bar{v}_4, EA]$	-5.19	**-0.19**

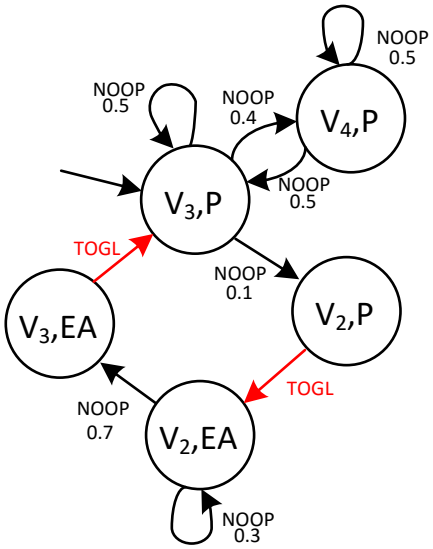

Figure 4.6: State transition graph for the nominal no-icing case

right turn at a constant airspeed. Let $C_2 = [\bar{\alpha}_1, \bar{\theta}_1, \bar{h}_4, \bar{t}_2, \bar{i}_4, \bar{f}_{17}, P_s]$ describe the constant state features during this stage. The final state-action utilities in this flight segment are described in Table 4.4. In nominal conditions $[\bar{v}_3, \bar{\phi}_1, C_2, P]$, the policy favors $NOOP$. However, as the bank angle steepens $[\bar{v}_3, \bar{\phi}_2, C_2, P]$, the probability of stalling the aircraft increases (Appendix Tables D.4 and D.5) so the policy indicates a $TOGL$ to EA. With EA control, the probability of transitioning to a bank angle state that reduces stall risk is higher. Control is transferred back to the pilot when a lower risk state is attained.

Note that the repeated selection of $TOGL$ leading to mode cycles between P and EA is discouraged by the reward term $\mathcal{R}_6$. However, the underlying transition probability models do not adequately capture pilot behavior that can lead to mode cycles due to the Markov assumptions. The following chapter explores formulations to adequately prevent mode cycling behaviors of the policy.

Table 4.4: State action utilities for $s = [\bar{V}, \bar{\Phi}, C_2, M]$. Left $M = P$, Right $M = EA$

$s \in \mathcal{S}$	$\mathcal{Q}(s, NOOP)$	$\mathcal{Q}(s, TOGL)$	$s \in \mathcal{S}$	$\mathcal{Q}(s, NOOP)$	$\mathcal{Q}(s, TOGL)$
$[\bar{v}_1, \bar{\phi}_1, P]$	-136.49	**-113.08**	$[\bar{v}_1, \bar{\phi}_1, EA]$	**-108.08**	-136.49
$[\bar{v}_2, \bar{\phi}_1, P]$	-29.24	**-12.72**	$[\bar{v}_2, \bar{\phi}_1, EA]$	**-7.72**	-29.24
$[\bar{v}_3, \bar{\phi}_1, P]$	**-4.40**	-12.72	$[\bar{v}_3, \bar{\phi}_1, EA]$	-7.72	**-4.40**
$[\bar{v}_4, \bar{\phi}_1, P]$	**-2.70**	-12.17	$[\bar{v}_4, \bar{\phi}_1, EA]$	-7.17	**-2.70**
$[\bar{v}_1, \bar{\phi}_2, P]$	-145.25	**-113.08**	$[\bar{v}_1, \bar{\phi}_2, EA]$	**-108.08**	-145.25
$[\bar{v}_2, \bar{\phi}_2, P]$	-38.18	**-12.72**	$[\bar{v}_2, \bar{\phi}_2, EA]$	**-7.72**	-38.18
$[\bar{v}_3, \bar{\phi}_2, P]$	-13.56	**-12.72**	$[\bar{v}_3, \bar{\phi}_2, EA]$	**-7.72**	-13.56
$[\bar{v}_4, \bar{\phi}_2, P]$	-13.14	**-12.17**	$[\bar{v}_4, \bar{\phi}_2, EA]$	**-7.17**	-13.14

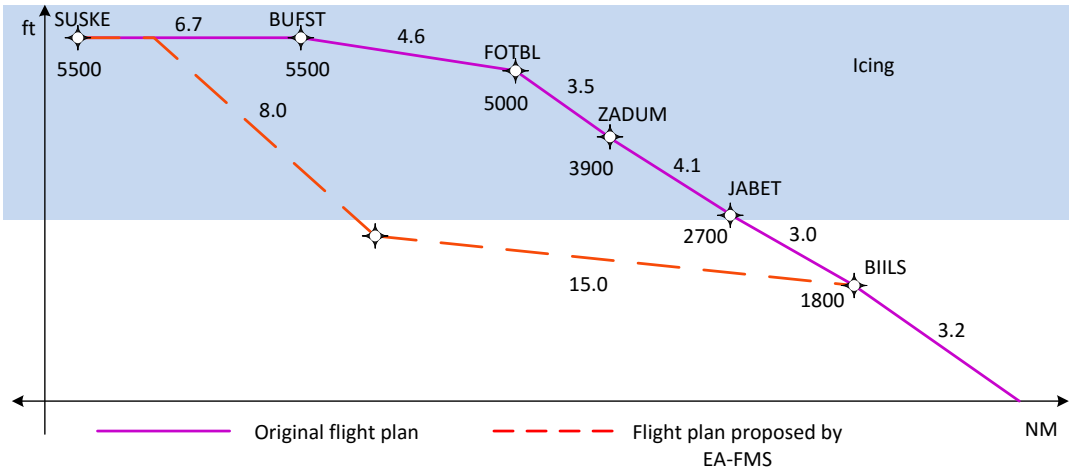

Figure 4.7: Vertical profile of the original flight plan indicating icing conditions and the flight plan proposed by the EA-FMS.

4.4.2 Flight with icing conditions

Consider the approach flight plan in Fig 4.5 but with icing conditions illustrated by the shaded region in Fig 4.7. Fig 4.7 illustrates the altitude profile of the flight plan in Fig 4.5. For this case the aircraft arrives at **SUSKE** in icing ($n_{ice} = 1$) such that the predicted time of exposure for the current flight plan is greater than the critical exposure time $t_{\text{pte}} > t_{\text{critical}}$, i.e. $\bar{I} = \bar{i}_3$.

Let $C_3 = [\bar{\alpha}_1, \bar{\theta}_1, \bar{\phi}_1, \bar{h}_1, \bar{t}_3, \bar{f}_2, P_s]$ denote features that remain constant during this seg-

ment of the flight plan. The two actions in $s_3 = [\bar{v}_3, C_3, \bar{i}_3, P]$ have utility:

$$\mathcal{Q}(s_3, NOOP) = \mathcal{R}(s_3, NOOP) + \gamma \sum_{s_k \in \mathcal{S}} \mathcal{T}_P(s_k|s_3)\mathcal{V}^*(s_k) = -15.2 \tag{4.12}$$

$$\mathcal{Q}(s_3, TOGL) = \mathcal{R}(s_3, TOGL) + \gamma \sum_{s_k \in \mathcal{S}} \mathcal{T}_{EA}(s_k|s_4)\mathcal{V}^*(s_k) = -11.6 \tag{4.13}$$

where $s_4 = [\bar{v}_3, C_3, \bar{i}_2, P]$. Selecting the higher-utility $TOGL$ action results in an indirect change of the expected icing intensity exposure feature from $\bar{I} = \bar{i}_3$ to $\bar{I} = \bar{i}_2$ in addition to changing the mode feature from $M = P$ to $M = EA$. The instantaneous reduction in expected icing indicated by $\bar{I}$ due to a $TOGL$ to $M = EA$ occurs due to an EA flight plan that directs the aircraft out of icing conditions quickly as illustrated in Fig 4.7. Switching to the Envelope-Aware controller within EAFMS in this case would result in a state where $t_{\text{pte}} <= t_{\text{critical}}$, $(\bar{I} = \bar{i}_2)$ since the risk of stalling is lower than in $\bar{I} = \bar{i}_3$ (see Appendix Tables D.2 and D.3). Following the new flight plan that has a lower time of exposure to icing minimizes the risk of in-flight icing-induced stall. Once the aircraft is out of icing conditions, the policy described in Section 4.4.1 applies and therefore control is handed back to the flight crew. Note that the reward formulation in Section 4.3.4 does not explicitly penalize states where $\bar{I} = \bar{i}_3$. However, the MDP policy is able to infer $\bar{i}_3$ poses high risk because continuing the flight plan under icing conditions when $\bar{I} = \bar{i}_3$ incurs a heavy future penalty due to the higher likelihood of stalling as shown in Table D.3. This work assumes that the flight crew adopts the new flight plan provided by the EAFMS. Future work will focus on better integrating nominal FMS information into FSAM to avoid mode cycling behavior.

Figures 4.8-4.10 illustrate numerical simulations of the icing case study presented above. Fig 4.8 indicates the altitude response of the aircraft with and without FSAM augmentation. As explained in Eqn 4.12-4.13, FSAM switches to the Envelope-Aware control law when the flight planning module indicates the availability of a flight plan that reduces exposure to icing conditions. When the flight crew adopts the new flight plan provided by EAFMS or enters a safe alternative, FSAM transfers control back to the flight crew. Without FSAM augmentation, as seen in Fig 4.10, flight through icing conditions requires increased usage of thrust and elevator commands to maintain the required altitude and airspeed. The increase in the control effort is attributed to the decreased lift and increased drag due to ice accretion.

On February 12, 2009, Colgain Air Flight 3407 [95] encountered icing conditions on a similar approach flight plan into **KBUF**. The flight crew failed to manage the aircraft's degrading airspeed while flying in icing conditions. Furthermore, the flight crew's inappropriate response to the subsequent stick-shaker warnings eventually resulted in loss of

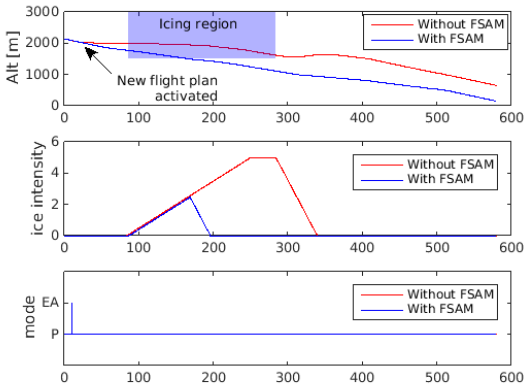

Figure 4.8: Altitude, icing intensity and control mode response

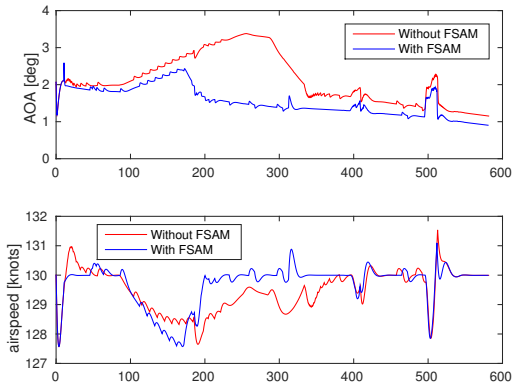

Figure 4.9: Angle of attack and airspeed response

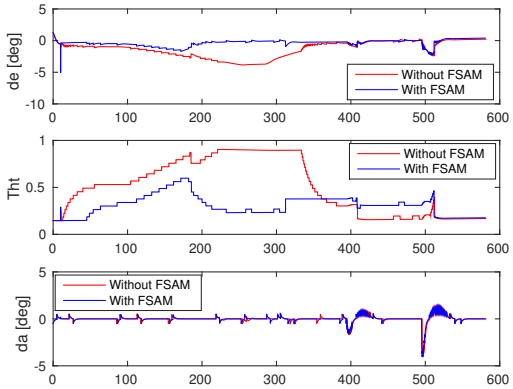

Figure 4.10: Control response

control. As described in the case study above, EAFMS including FSAM would have prevented the aircraft's airspeed from decreasing below the stall speed. FSAM would also have triggered a flight plan change based on the flight planning capabilities of the EAFMS [86] and enabled an alternate flight plan to a safe landing that had a minimum time of exposure to icing. Real world icing related accidents such as Colgan Air Flight 3407 and American Eagle Flight 4184 [96] can be avoided in the future once EAFMS capabilities are infused in cockpit automation (and unmanned aircraft automation).

4.5 Conclusions

This chapter presented a Markov Decision Process formulation for Flight Safety Assessment and Management that can make control authority switching decisions to minimize or reduce loss of control risk given icing conditions. A discrete state-space abstraction was designed to efficiently capture pertinent information regarding aircraft dynamics, control, and expected time of exposure to icing conditions. Abstractions of these state features were constructed based on parameterized flight envelopes updated during progressive icing via online estimation. Reward functions penalized states with high loss of control risk and any control authority mode disagreeing with crew-indicated preference. Icing and no-ice landing approach case studies were presented. Several assumptions were made to simplify construction of transition probabilities. Ultimately transition probabilities and reward weights must be constructed from data collected over simulations, in-flight testing and data analysis, and focused pilot subject experiments.

CHAPTER 5

Managing FSAM MDP Complexity for Online Execution

5.1 Introduction

The optimal policy for an MDP can be obtained using algorithms such as value iteration, policy iteration or linear programming [62] as described in Chapter 3. Such algorithms explicitly enumerate all MDP states. The MDP formulations for FSAM described thusfar focused on compactly representing the MDP states but used the traditional solvers. Though abstraction enabled us to reduce the computational complexity relative to variable representation over a fine grid of values, solving the MDP can still be prohibitively expensive once simplifying assumptions are relaxed. Consequently, this chapter explores an approximate method to find near-optimal solutions using a sparsely sampled Monte Carlo tree search algorithm [69]. This chapter contributes an online implementation of the FSAM MDP. An online implementation enables incorporating changes to aircraft dynamics, health and weather information in real-time as opposed to relying on a database of policies applicable to different LOC situations constructed offline [86]. Furthermore, the sparse sampling algorithm reduces the knowledge engineering required to compactly define the MDP state space partitions as described in previous chapters.

5.2 Sparse Sampling for Large MDPs

Sparse sampling for large state-space MDPs was originally introduced by Kearns et al. [69]. Given a generative model $\mathcal{G}$ of an MDP[1], the sparse sampling algorithm executes the following steps:

[1] A generative model takes as input a state-action pair (s,a) and outputs $\mathcal{R}(s,a)$ and state s', where s' is randomly drawn from next state distribution $\mathcal{P}(s'|s,a)$.

1. For each action a, the generative model computes $\mathcal{R}(s,a)$ and independently samples S_a of C states from next-state distribution $\mathcal{P}(s'|s,a)$.

2. For each state in S_a, Step 1 is repeated until horizon H to construct a finite look ahead tree (Fig 5.1).

3. The estimate of optimal value $\mathcal{V}^*(s)$ is given by:

$$\hat{\mathcal{V}}^*_H(s) = \max_a \left\{ R(s,a) + \gamma \frac{1}{C} \sum_{s' \in S_a} \hat{\mathcal{V}}^*_{H-1}(s) \right\} \tag{5.1}$$

   Note that Eqn (5.1) computes $\hat{\mathcal{V}}^*_H(s)$ recursively from $\hat{\mathcal{V}}^*_0(s) = 0$.

4. The optimal action is then given by:

$$\arg\max_a \left\{ \mathcal{R}(s,a) + \gamma \frac{1}{C} \sum_{s' \in S_a} \hat{\mathcal{V}}^*_{H-1}(s) \right\} \tag{5.2}$$

Branching factor C and horizon length H can be chosen to manage approximation error (i.e. $||\mathcal{V}^*(s) - \hat{\mathcal{V}}^*(s)||$) as described in [69]. Note that this algorithm does not require enumeration of all MDP states. It can be applied when the MDP state-space is discrete, continuous, or mixed. Computation time can be further reduced by independently evaluating each branch at the root node using multi-core processors or GPUs.

5.3 Sparse Sampling Applied to FSAM

The FSAM MDP formulations described in Chapter 4 and in Chapter 3 can be solved using the Sparse-Sampling algorithm. The original MDP formulation consisted of a factored state representation involving various features such as aircraft dynamics, aircraft health, flight crew and environmental characteristics. This work assumes that the aircraft health, flight crew and environmental features remain constant and will focus only on the aircraft dynamics features. Thus, the MDP state is given as:

$$\begin{aligned} s &= [F_{11}, F_{12}, F_{13}, F_{14}] \\ F_{11} &= [u, v, w, p, q, r, \phi, \theta, \psi, x, y, h] \\ F_{12} &= [\delta_e, \delta_a, \delta_r, \delta_t] \\ F_{13} &= [c_g, c_f, c_p] \\ F_{14} &= [\bar{M}, \bar{S}, \bar{N}, \bar{T}] \end{aligned} \tag{5.3}$$

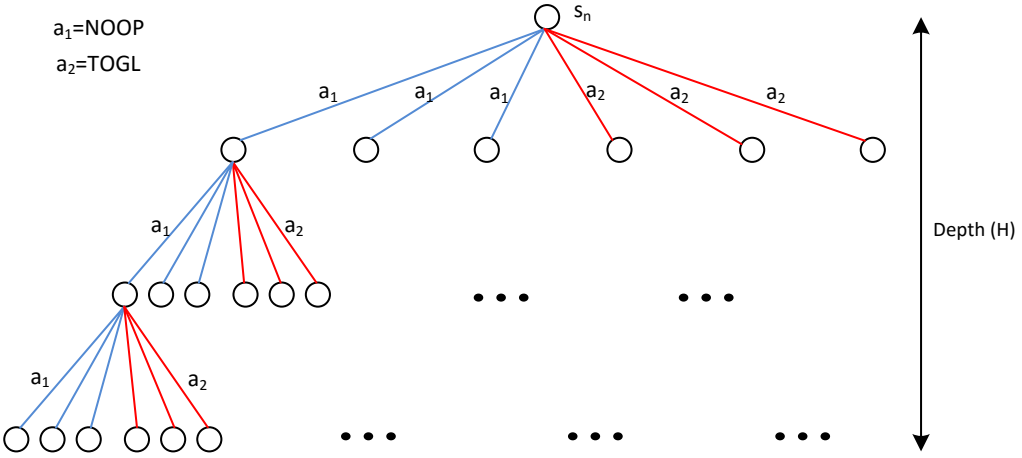

Figure 5.1: Sparse sampled look-ahead tree with two actions and a branching factor of three

Here F_{11} describes traditional aircraft *physical* state [70]. u, v, w describe aircraft linear velocity in the body frame, p, q, r are the body axis angular rates, ϕ, θ, ψ represent Euler angle attitude, and x, y, h denote 3-D position relative to a ground fixed frame. F_{12} describes fixed-wing control inputs elevator (δ_e), aileron (δ_a), rudder (δ_r), and throttle (δ_t). Throttle is assumed to provide symmetric thrust to the airframe. F_{13} describes the configuration of the aircraft in terms of flaps (c_f), spoilers (c_p) and landing gear (c_g). F_{14} specifies current control mode $\bar{M}$, the number of override directives previously issued $\bar{N}$, and time elapsed in the current control mode $\bar{T}$. $\bar{S}$ denotes the mode select switch. F_{11} and F_{12} contain continuous-valued variables, F_{13} takes discrete values. $\bar{M}, \bar{S}, \bar{N}$ in F_{14} are discrete, and $\bar{T}$ is continuous. All F_1 parameters are observable from onboard sensors.

Instead of specifying transition probabilities as in the previous MDP formulations, here we specify a generative model. The generative model is a function that takes as inputs the current state s^n, action $a^n \in \{NOOP, TOGL\}$ and outputs the reward $\mathcal{R}(s^n, a^n)$ and the next state s^{n+1} chosen according to state distribution $\mathcal{P}(s^{n+1}|s^n, a^n)$. The next state distribution is expressed in terms of the state features as follows:

$$\mathcal{P}(s^{n+1}|s^n, a^n) = \mathcal{P}(F_{11}^{n+1}, F_{12}^{n+1}, F_{13}^{n+1}, F_{14}^{n+1}|F_{11}^n, F_{12}^n, F_{13}^n, F_{14}^n, a^n) \tag{5.4}$$

Conditional independence among state features can be exploited to simplify Eqn (5.4):

$$\begin{aligned}\mathcal{P}(s^{n+1}|s^n,a^n) =&\mathcal{P}_{11}(F_{11}^{n+1}|F_{11}^n,F_{12}^n,F_{13}^n)\mathcal{P}_{12}(F_{12}^{n+1}|F_{11}^{n+1},F_{12}^n,F_{13}^n,a^n)\times\\ &\mathcal{P}_{13}(F_{13}^{n+1}|F_{13}^n,F_{14}^n)\mathcal{P}_{14}(F_{14}^{n+1}|F_{14}^n,a^n)\end{aligned} \tag{5.5}$$

Here, $\mathcal{P}_{11}$ represents the aircraft state transition model. Samples are drawn from $\mathcal{P}_{11}$ using a stochastic model of the aircraft dynamics:

$$X_{n+1} = X_n + \mathcal{F}(X_n, U_n)\Delta t + W_n \tag{5.6}$$

where $\mathcal{F}$ represents the equations of motion and W is a state disturbance vector with Gaussian noise. Δt is the discretization time step. A Twin-Otter aircraft model [97] is used in Eqn (5.6). $X = F_{11}$ is aircraft physical state, $U = F_{12}$ represents physical control inputs.

$\mathcal{P}_{12}$ describes the control input distribution. We assume there are two control authorities, a pilot/crew and an envelope-aware safety controller. Pilot control inputs are modeled as human operator transfer functions [74] with parameters chosen according to a user-specified distribution. The envelope-aware controller is modelled as a Linear Quadratic Regulator (LQR) control law [98] designed by linearizing the aircraft dynamics about a steady, level flight trim-condition at a specific airspeed (55 m/s) and altitude (2500 m) for this study.

$\mathcal{P}_{13}$ represents transitions in aircraft configuration. For cruise flight the configuration is constant at no flaps, gear up, no spoilers. $\mathcal{P}_{14}$ represents transitions in control mode. A transition from one control authority to another occurs when $a_n = TOGL$.

The reward function $\mathcal{R}(s^n, a^n)$ is a "cost" (negative reward) function that penalizes unsafe aircraft states but discourages routine override directives. A weighted additive reward function is defined:

$$\mathcal{R} = \sum_{i=0}^{n} w_i \mathcal{R}_i \tag{5.7}$$

The $\mathcal{R}_i$'s penalize unsafe states and unnecessary override actions while w_i's represent tunable weighting parameters that may vary depending as a function of flight condition. For example, the penalty for violating the stall constraint at high altitude can be lower than the stall penalty at low altitude due to the availability of ample altitude to recover. Appropriate choice of weighting parameters may also be learned from accident flight data. The reward

functions used in this work are discontinuous. [2]

In this work, the following reward terms are proposed: $\mathcal{R}_1$ penalizes excursion outside the valid airspeed envelope defined by stall speed V_{min} and never-exceed speed V_{max} above which structural over-stressing can occur.

$$\mathcal{R}_1 = \begin{cases} -1 \;\; if \; (V \leq V_{min}) \; \vee \; (V \geq V_{max}) \\ 0 \;\; otherwise \end{cases} \tag{5.8}$$

$\mathcal{R}_2$ imposes a penalty on out-of-envelope bank attitude, where ϕ_{min}, ϕ_{max} indicate acceptable bank limits.

$$\mathcal{R}_2 = \begin{cases} -1 \;\; if \; (\phi \leq \phi_{min}) \; \vee \; (\phi \geq \phi_{max}) \\ 0 \;\; otherwise \end{cases} \tag{5.9}$$

$\mathcal{R}_3$ penalizes altitude constraint violations. Factors such as filed flight plan, terrain, flight ceiling, engine failure, and cabin de-pressurization impose altitude constraints.

$$\mathcal{R}_3 = \begin{cases} -1 \;\; if \; (h \leq h_{min}) \; \vee \; (h \geq h_{max}) \\ 0 \;\; otherwise \end{cases} \tag{5.10}$$

$\mathcal{R}_4$ penalizes deviations from the prescribed flight plan:

$$\mathcal{R}_4 = \begin{cases} -1 \;\; if \; \|X - X_0\| > \epsilon \\ 0 \;\; otherwise \end{cases} \tag{5.11}$$

Here $\|X - X_0\|$ represent position deviation from the nominal flight plan. ϵ represents the acceptable deviation from the nominal flight plan. $X = [x, y, h]$ is the position of the aircraft.

To ensure override actions are not issued by FSAM unnecessarily, $\mathcal{R}_5$ imposes a penalty on choosing an override action. $\mathcal{R}_6$ prevents repeated switching between control authorities by imposing a penalty for an override that is inversely proportional to the duration since

[2] For continuous valued variables, rewards may become continuous barrier functions that prevent constraint violations.

Table 5.1: Computation times with a fixed decision epoch

Δt	H	$C = 40$	$C = 64$	$C = 80$
0.1	5 (5s)	0.041s	0.176s	0.4242s
	10 (10s)	0.623s	5.950s	21.636s
	15 (15s)	$> 25s$	$> 25s$	$> 25s$
0.5	5 (5s)	0.007s	0.033s	0.129s
	10 (10s)	0.113s	1.130s	3.690s
	15 (15s)	3.460s	$> 25s$	$> 25s$

the last override. $\mathcal{R}_7$ penalizes the total number of overrides.

$$\mathcal{R}_5 = \begin{cases} -1 \; if \; (\bar{M} = P) \wedge (a \neq NOOP) \\ 0 \; otherwise \end{cases} \tag{5.12}$$

$$\mathcal{R}_6 = \begin{cases} -\frac{1}{T} \;\; if(\bar{T} > 0) \wedge (a = TOGL) \\ 0 \;\; otherwise \end{cases} \tag{5.13}$$

$$\mathcal{R}_7 = -\bar{N}_{TOGL} \tag{5.14}$$

In this work, the Sparse-Sample MDP formulation is only described with respect to the aircraft dynamics state feature. A formulation taking into account other features such as aircraft health, pilot characteristics and environmental features are provided in [99].

We now illustrate the application of the sparse-sampling algorithm to make override decisions on simple loss of control scenarios. For this illustration, $1000, 1000, 1000, 10, 5, 0.8, 1$ were chosen as weights for the reward functions $\mathcal{R}_1, \ldots, \mathcal{R}_7$ respectively. A discount factor γ of 0.7 was chosen. The numerical values for the weights and discount factor were chosen after several trials with different values in search of the desired policy[3]. The parameters required to construct the sparse look-ahead tree are branching factor C, look-ahead horizon H and time-step (decision epoch) ΔT. Branching factor varies as a function of tree depth m as $C_m = \gamma^{2m} C$ to reduce computation time while maintaining a good approximation of the optimal solution [69]. Parameter values and their effects on computation time are shown in Tables 5.1 and 5.2.

Consider a scenario where the aircraft is prohibited from flying below $2400m$ (i.e. $h_{min} = 2400m$) due to terrain hazards. Suppose the pilot pushes the elevator down to initiate a dive with altitude loss. The red plot in Fig 5.2 indicates the aircraft's response

[3] A desirable policy in this case prevents the aircraft from entering unsafe regions of the flight envelope

Table 5.2: Computation times with variable decision epoch

Δt	H	$C = 40$	$C = 64$	$C = 80$
0.1	5 (15s)	0.119s	0.693s	2.455s
	10 (55s)	7.239s	$> 25s$	$> 25s$
0.5	5 (15s)	0.018s	0.130s	0.388s
	10 (55s)	1.272s	12.672s	$> 25s$

without FSAM intervention and the blue plot indicates the response with FSAM intervention. FSAM remains passive until the airplane is near the envelope boundary then overrides the pilot to prevent attitude constraint violation. Control is restored to the pilot after the envelope-aware controller recovers and climbs to 2500m. Note that h_{min} and the other parameters in $\mathcal{R}$ may vary depending on the flight phase, surrounding terrain, environmental conditions and airplane performance. Thus, appropriate policies can be constructed for different flight conditions. For example, in a landing phase MDP policy, h_{min} would be defined based on the surrounding terrain and FSAM would typically not override the pilot unless collision with surrounding terrain was imminent.

Policy behavior can be changed by tuning Eqn (5.7) weights. For example, varying the weight on $\mathcal{R}_6$ controls the duration the envelope-aware controller stays active after overriding the pilot. Similarly, the weight on $\mathcal{R}_7$ controls the number of overrides issued. Thus, if the pilot behaves inappropriately by repeating the above nose-down pitch inputs continuously, control will be eventually transferred to the envelope-aware controller and not returned to the pilot.

Fig 5.3 illustrates a scenario where the policy prevents aerodynamic stall. FSAM overrides for this case when the airspeed approaches the stall speed. The envelope-aware controller then increases the airspeed to prevent the stall. The optimal policy was computed on a desktop with a 3.6 GHz, 8 core-Intel Xeon processor and 8 GB RAM. Each search tree branch is independent so the expansion of the branches can be parallelized to reduce the computation time (see *root parallelization* in [100]). In this work, the computations were distributed across 8 cores. Table 5.1 lists the time taken to compute the sparse look-ahead solution for different tree parameters and model complexities. Δt denotes the discretization time used to forward propagate the aircraft dynamics in Eqn (5.6). Thus, the decision epoch ΔT must be an integer multiple of discretization time-step Δt (i.e. $\Delta T = n\Delta t$). Consequently, the generative model needs to be forward propagated n times to reach the next decision epoch. Table 5.1 shows results for a fixed decision epoch $\Delta T = 1s$. As expected,

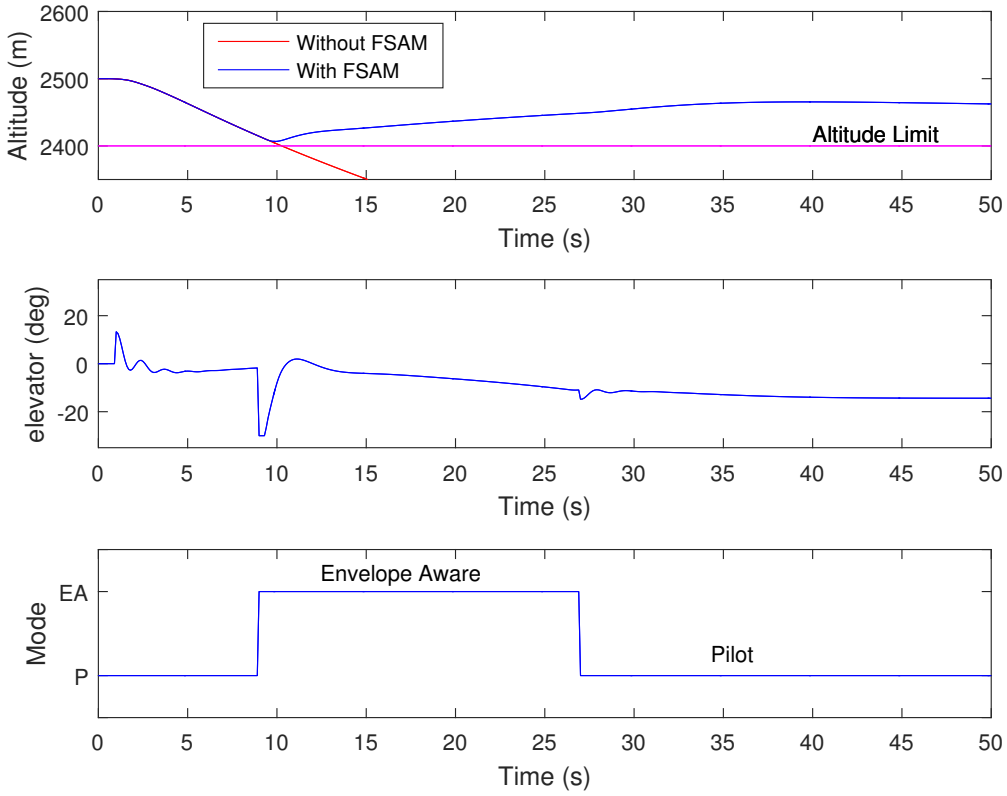

Figure 5.2: Altitude recovery

computation time decreases as the complexity of the generative model decreases. With a fixed decision epoch, real-time execution requires that only a short horizon be used for the finite-look ahead search. This is sufficient to avoid LOC events with fast dynamics. To address events such as controlled flight into terrain, a longer horizon may be preferable. Table 5.2 illustrates results obtained using a variable decision epoch. Here $\Delta T = m$ where m denotes the current depth in the tree. Note that with a variable decision epoch, it is possible to increase horizon without substantial computation penalty.

5.4 Discussion and Conclusions

This chapter contributes an online decision-theoretic framework for a Flight Safety Assessment and Management system. A comprehensive, integrated state feature set enables FSAM to base its decisions on system-wide information describing the aircraft (vehicle), people, and environment. The presented list can be expanded in future work. The applied

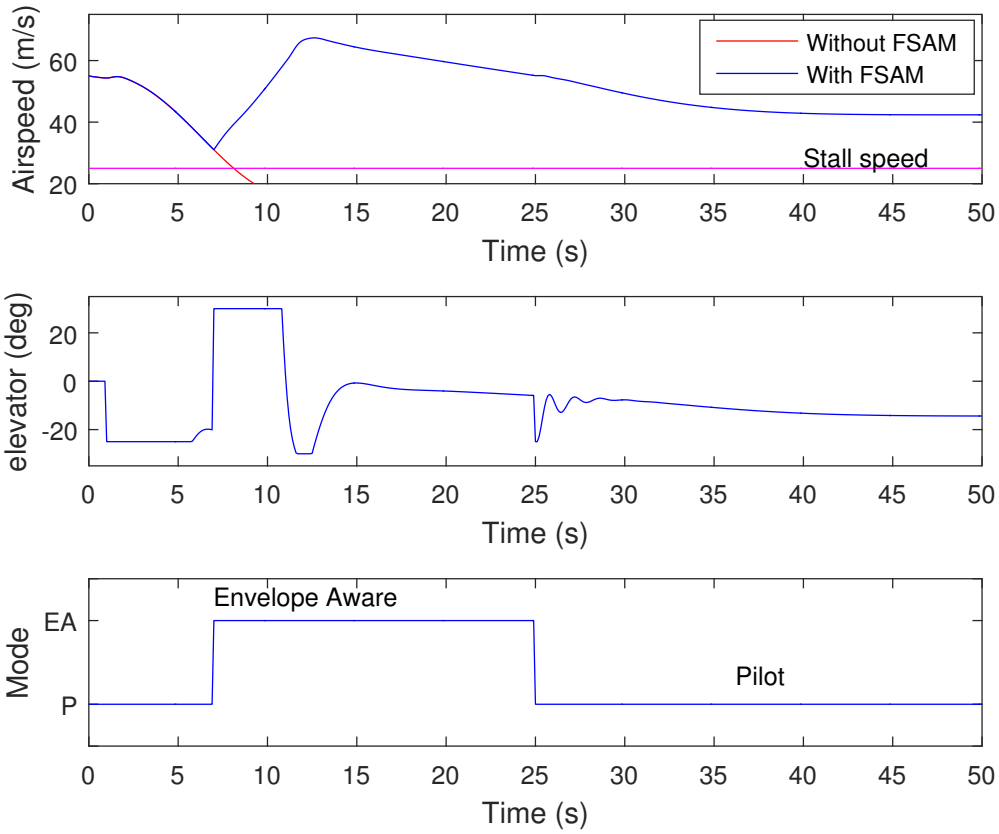

Figure 5.3: Stall recovery

sparse sampling algorithm develops near-optimal solutions efficiently by eliminating the need to explicitly enumerate the state space. Though run time doesn't depend on state-space size, it does depend on horizon length and look-ahead tree branching factor.

The use of a linearized aircraft model to generate state distributions reduces computation times significantly in comparison to a detailed non-linear aircraft model with aerodynamic look-up tables. The online sparse sampling algorithm supports interleaved planning and execution which facilitates online model updates. System identification techniques can be used to update models based on real-time flight data. Observations of pilot behavior can be used to update the human transfer function model and predict pilot intentions.

The MDP formulation can be simplified via state and reward formulations specific to phase of flight (takeoff, climb, cruise, descent, landing). State feature time scale separation or sequencing can also be exploited to further decompose the MDP into several simpler MDPs. Control authority switching might be specified with finite state machines but man-

ually generating state machines as specified in Chapter 2, but online tuning would then not be possible.

This work illustrated a case study focused on aircraft dynamics and controls while assuming remaining state features are constant. Models describing the transitions (dynamics) of the remaining features must be developed in future work. Recognizing scenarios where the underlying assumptions of a given MDP formulation fail is also essential to ensure FSAM policies don't pose new risk in perceived LOC scenarios. Future research directions will formally analyze such scenarios and develop strategies to ensure that the actions of FSAM will not jeopardize nominal aircraft operations.

CHAPTER 6

Verification Guided Refinement of FSAM

6.1 Introduction

The previous chapters describe methods to construct a control mode switching strategy for FSAM. However, for a safety critical Flight Management System augmentation such as FSAM, it is crucial to ensure that the control mode switching strategy prevents or minimizes the risk associated with LOC. It is also important to ensure that FSAM does not interfere unnecessarily with nominal flight crew operations. This chapter investigates the formal verification of FSAM.

Verification and Validation (V&V) are essential steps in the traditional V-model [101] for systems engineering as illustrated in Fig 6.1. 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 6.1) can be labor-intensive and costly but has been shown an effective means to organize system development.

Formal methods such as model checking [44] and deductive techniques [102] efficiently augment traditional V&V [101]. Formal methods help establish the correctness of a system design with respect to specified requirements prior to 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 [103]. Rockwell Collins used a theorem proving approach to verify the functionality of a flight guidance system [104, 105]. Joshi et al [106] proposed a model-

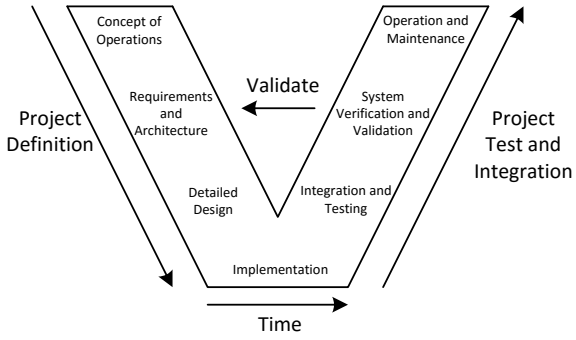

Figure 6.1: V model for System Engineering [6]

based safety analysis which extends model checking with fault trees used to analyze safety-critical components. The idea of a sandbox controller was introduced by Bak et al [18] where a nominal system is augmented with a safety controller and a decision module to prevent the system from entering an unsafe state. Lygeros et al [107] used an automaton-based method to verify their Traffic Collision Avoidance System (TCAS) conflict resolution algorithm.

The above 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 et al [108]. In [108] interactions between a human operator and a machine are formally analyzed to guide the design of the interfaces between human and machine and to develop better training manuals. Bolton et al [109] presented an approach to verify human automation interaction using task analytical models. In [109], a 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.

To verify safety properties of dynamical systems, several researchers have focused on estimating the reachable states of the system and ensuring this reachable set does not contain the unsafe states [110–114]. The use of backward reachable sets to verify safety of dynamical systems has been considered by Tomlin et al. [42, 50, 115]. Prajna et al [116] and Tobenkin et al [117] explored the use of barrier certificates that guarantee that trajectories of the dynamical system do not leave a safe set of states. The above methods typically make assumptions on the nature of the dynamical system and are only applicable only to systems of low dimensions. The use of probabilistic model checking techniques [118–120] have been widely applied to verify safety properties for systems whose models are not available analytically but are available in the form of a black box.

This chapter contributes (i) a general approach to verify a switched control system whose switching policy is realized by a deterministic finite state Moore machine [30] and (ii) a model checking framework to guide the refinement and verification of 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 first established. Next, a discrete transition system that encodes an over-approximation 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 [121], a popular model checking tool. In this chapter, 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 [122] to facilitate model checking. Counterexamples obtained from the model checking process identify necessary refinements of the underlying FSAM switching protocol.

Section 6.2 provides background on the tools necessary to perform model checking, Section 6.3 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 6.4 describes the proposed approach for model checking and the results of verification. Section 6.5 discusses refinements to FSAM based on the results of verification, while Section 6.6 considers validation of FSAM. Section 6.7 provides a discussion of our verification approach while Section 6.8 presents conclusions and future extensions.

6.2 Background

Model checking is the process of ensuring that a system satisfies a set of requirements. This section introduces the specification formalisms used in this paper to enable FSAM model checking. The safety requirements for takeoff are expressed in Linear Temporal Logic. Formal definitions are provided below.

6.2.1 Linear Temporal Logic

Linear Temporal Logic (LTL) is a formal specification language [44, 123] 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

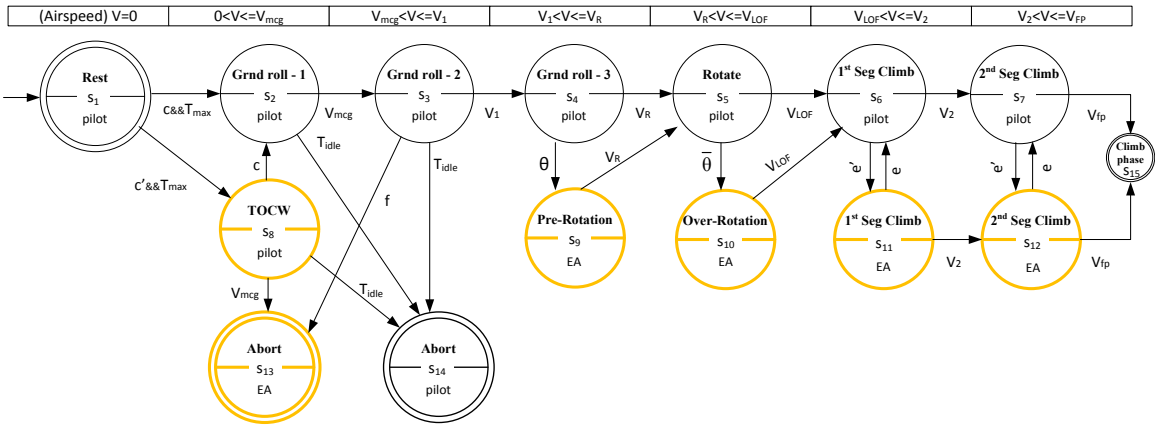

Figure 6.2: Longitudinal FSAM moore machine

$\bigcirc$ (*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: 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 [123, 124] 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 dynamical systems, i.e., discrete-time semantics of LTL [125].

6.3 Problem Formulation

This section formulates the model checking problem for the FSAM Deterministic Moore Machines (DMM) introduced in Chapter 2. 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 work is outlined.

6.3.1 Longitudinal Dynamics for Takeoff

To illustrate verification of the longitudinal FSAM Deterministic Moore Machines (DMM) (see Fig 6.2), the following simplifying assumptions are made:

- The lateral dynamics is well behaved, i.e., there are no lateral disturbances so the aircraft can maintain runway heading while staying on runway centerline throughout takeoff.
- The engines, control surfaces, instruments and all subsystems function nominally.
- There is no runway incursion risk.
- The only pilot behaviors impacting FSAM decisions are related to modeled configuration settings and control inputs.
- Envelope-aware control and guidance algorithms are capable of maintaining or recovering a safe state in any DMM where 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 [108,109,126]. The above assumptions support the usage of simple pilot behavior models (human operator transfer functions [74]) in verifying the FSAM DMM.

The full nonlinear equations describing the dynamics of the aircraft during takeoff are provided in Appendix A. The above 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 near constant until achieving rotation airspeed V_R. At or above V_R, the pilot applies control inputs to pitch the nose of the aircraft up. When lift-off speed V_{lof} is reached, the aircraft climbs (note $V_{lof} > V_R$). Thus, the longitudinal dynamics for takeoff can be split into segments $V < V_{lof}$ and $V \geq V_{lof}$:

$$
\begin{aligned}
\dot{x} &= \begin{cases} v & v < V_{lof} \\ v cos(\gamma_0) & v \geq V_{lof} \end{cases} \\
\dot{v} &= \begin{cases} A_1 - B_1 v^2 & v < V_{lof} \\ A_2 - B_2 v^2 & v \geq V_{lof} \end{cases} \\
\dot{h} &= \begin{cases} 0 & v < V_{lof} \\ v sin(\gamma_0) & v \geq V_{lof} \end{cases} \\
\dot{q} &= \begin{cases} A_3 q + B_3 u_e & v < V_R \\ A_4 q + B_4 u_e & v \geq V_R \end{cases} \\
\dot{\theta} &= q
\end{aligned}
\tag{6.1}
$$

where x is the longitudinal position of the aircraft on the runway, v is airspeed, θ is pitch, q is angular rate, h is altitude and γ is flight path angle. Terms $A_1, B_1, A_2.B_2$ are defined using a formulation from [41]:

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

Here T represents the takeoff thrust, W is aircraft takeoff weight, ρ is atmospheric density, μ is wheel rolling friction coefficient, and γ_0 is 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 flap/slat settings. Pitch dynamics are approximated as a piece-wise 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, Eqn 6.1 is represented compactly as $\dot{X} = f(X, U)$. Here $X \in \mathcal{X} \subseteq \mathbb{R}^5$ represents the state vector $[x, v, h, \theta, q]^T$, where $\mathcal{X}$ is a compact hyper-rectangle. $U \in \Omega \subseteq \mathbb{R}$ represents the elevator control input u_e. $\{P, EA\}$ are the available control authorities.

When off-nominal conditions are encountered during takeoff, FSAM transfers control to the Envelope Aware (EA) safety controller that attempts LOC prevention or recovery. The pilot and controller models used in this work are described in Appendix B.

6.3.2 Safety Requirements for the 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 never enters an unsafe state. A discussion of safe and unsafe states during takeoff can be found in [85]. For the purpose of illustration, in this work the primary focus is on verifying safety requirements or properties specified in Part 25 [127] of the FARs (Airworthiness Standards: Transport Category Aircraft). This paper verifies that the longitudinal FSAM DMM (Fig 6.2) meets the requirements listed in Table 6.1. Table 6.1 also provides the LTL expression for each requirement. In Table 6.1, θ_{ng} is the pitch attitude at which the nose gear first leaves the ground, θ_0 is

Table 6.1: Requirements and their LTL specifications

#	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)]	$\Box((\theta \geq \theta_{ng}) \rightarrow (V \geq V_R))$
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]	$\Box((h \leq h_{lof}) \rightarrow (\theta < \theta_{tail}))$
3	The slope of the airborne part of the takeoff path must be positive at each point. [FAR 25.111.(c).1]	$\Box((h > h_{lof}) \rightarrow (\theta > \theta_0))$
4	The airplane must reach V_2 before it is 35 feet above the takeoff surface. [FAR 25.111.(c).2]	$\Box((h \geq h_{obs}) \rightarrow (V \geq V_2))$

the pitch attitude that provides a non-negative flight path angle [1]. θ_{tail} is the pitch attitude at which the tail contacts the ground prior to liftoff (i.e., when $h \leq h_{lof}$), and h_{obs} is the nominal obstacle clearance height, typically 35 ft for commercial aircraft [127].

6.3.3 Verification problem specification and approach

Let $f(.)$ denote the dynamics of takeoff and let $Reach(f, I)_{\mathcal{A}}$ 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 $\bar{\mathcal{U}}$ denote the set of unsafe states identified by the requirements (e.g., Table 6.1). The safety verification problem then reduces to checking the validity of the following expression [113]:

$$Reach(f, I)_{\mathcal{A}} \cap \bar{\mathcal{U}} = \phi \tag{6.2}$$

Computation of the reachable set $Reach(f, I)_{\mathcal{A}}$ can be challenging especially if the underlying dynamics f is nonlinear [114]. Several authors have developed different ap-

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

proaches to compute reachable sets. The successes of these approaches are typically determined by the representations used to approximate the reachable sets. In this work, a discrete over-approximation [113, 128] of the dynamics in the form of a finite transition system is developed with the following steps: (i) 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. (ii) Abstract the dynamics as a finite transition system that takes into account the behavior of the pilot and the EA controller, (iii) Compose the abstraction with FSAM to obtain an over-approximation of the closed loop behavior. To verify Eqn 6.2, an automaton-theoretic approach is used wherein the over-approximation 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, an existing model checker, SPIN [121], is used. The three steps of this model checking process are discussed in detail below.

6.4 Verification of Takeoff FSAM

6.4.1 State Space Abstraction

The first step to verification requires defining a set of atomic propositions for model checking. These propositions above capture thresholds essential to verify requirements. The requirements defined above are only related to airspeed (V), pitch (θ) and altitude (h), yielding propositions, Π_V, Π_θ, Π_H. Here $\Pi_V = \{\pi_{v1}, ..., \pi_{v8}\}$ is the set of propositions that defines a discrete set of airspeed values, $\Pi_\theta = \{\pi_{\theta 1}, ..., \pi_{\theta 5}\}$ defines a discrete set of pitch values, and $\Pi_H = \{\pi_{H1}, ...\pi_{H4}\}$ defines a discrete set of altitude values. The propositions (shown in Table 6.2) 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[2]. For example, a tail strike (a state where the tail of the aircraft strikes the runway) is identified by the propositions indicated in Fig 6.3.

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 $X^* = [x, v^*, h^*, \theta^*, q]$ where $0 \leq v^* < V_{mcg}$, $\theta_1 \leq \theta^* < \theta_2$ and $h_1 \leq h^* < h_2$.

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

Table 6.2: Atomic propositions

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

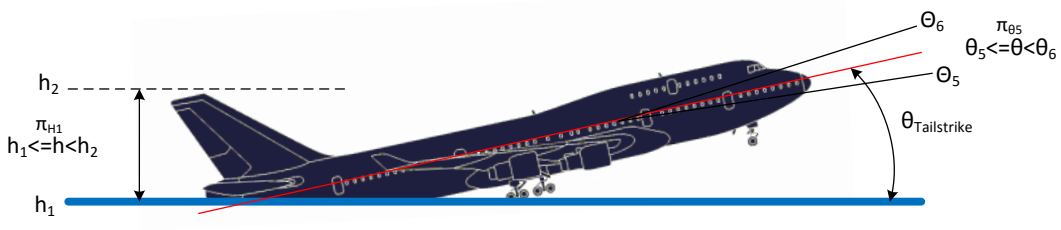

Figure 6.3: Partitions that enable identification of a tail strike

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 hyper-rectangle) formed by the set $\{\mathcal{H}^{-1}(\pi_{vi}) \times \mathcal{H}^{-1}(\pi_{\theta j}) \times \mathcal{H}^{-1}(\pi_{hk})\}$. 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 $Q := \{\bar{q}_1, \ldots, \bar{q}_n\}$ is associated with each element of this partition. As a result, the observation map induces the abstraction $\bar{T} : \mathcal{X} \to Q$ that maps each state $X \in \mathcal{X}$ into the finite set Q. The map $\bar{T}$ is proposition-preserving if and only if

$$\bar{T}(X_1) = \bar{T}(X_2) \Rightarrow \mathcal{H}(X_1) = \mathcal{H}(X_2), \ \ \forall X_1, X_2 \ \in \mathcal{X}, \tag{6.3}$$

Eqn 6.3 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 belong to the facets [3] of the polytope $\bar{T}^{-1}(\bar{q})$. In this chapter, 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 $\overset{\circ}{q}_i \in \mathcal{F}(\bar{q})$ is referred to as the i^{th} facet of cell $\bar{q}$.

The atomic propositions in Table 6.2 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

[3] A facet of a polytope of n dimensions is a face that has $(n-1)$ dimensions.

to an unsafe cell must pass through the facet of the unsafe cell. Hence, for the verification of FSAM according to Eqn 6.2, 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.

6.4.2 A Discrete Representation of Reachable States

In principle, given a proposition-preserving partition and the dynamics, it is possible to use the methods in [50, 51, 110, 111, 115, 123] 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 $h(t)$ described by the dynamics in Eqn 6.1 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 Section 6.3.2 are only invariance requirements [44]. These properties simplify construction of a discrete abstraction of the reachable states.

The method used to construct the discrete abstraction is shown in Algorithm 6.1. Inputs include discrete state space partition Q, the takeoff dynamics model f and the two controller formulations (P, EA) described in Appendix B. The algorithm returns a discrete transition system $\mathcal{B} := (\mathcal{Q}, \mathcal{Q}_0, \mathcal{P}, \Gamma_{\mathcal{B}}, \Pi, \mathcal{L})$ whose states $\mathcal{Q}$ represent facets of the cells in the partition. Actions $\mathcal{P} = \{P, EA\}$ denote the two control authorities and $\Gamma_{\mathcal{B}}$ describes the transitions between facets under the two control authorities. The function $isReach(\overset{\circ}{q}, k, \overset{\circ}{q}')$ returns true if for $k \in \{P, EA\}$, there exists t_0, t_1, $X(t_0) \in \overset{\circ}{q}$, $X(t_1) \in \overset{\circ}{q}'$ such that $X(t) \in \bar{q}$ for all $t \in [t_0, t_1]$ where $\bar{q}$ is the cell containing the two facets $\overset{\circ}{q}$ and $\overset{\circ}{q}'$. The function $isReach(\overset{\circ}{q}, k, \overset{\circ}{q}')$ can in general be evaluated using methods described in [110–112, 129]. A description of the $isReach(\overset{\circ}{q}, k, \overset{\circ}{q}')$ function used in this work and specific numerical values describing the state space partition can be found in Table E.1.

Each state $\overset{\circ}{q} \in \mathcal{Q}$ in $\mathcal{B}$ contains transitions induced by the pilot (P) and the envelope-aware safety controller (EA). 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 $\overset{\circ}{q} \in \mathcal{Q}$ are extracted by constructing the composition (product) of the transition system $\mathcal{B}$ and FSAM DMM $\mathcal{A}$.

6.4.3 Composite Transition System

The composition of the discrete transition system $\mathcal{B} := (\mathcal{Q}, \mathcal{Q}_0, \mathcal{P}, \Gamma_{\mathcal{B}}, \Pi, \mathcal{L})$ and the FSAM DMM $\mathcal{A} := (\mathcal{S}, \mathcal{S}_0, \Sigma, \Lambda, \mathcal{T}, \mathcal{G})$ yields a new transition system $\mathcal{C} := (\mathcal{D}, \mathcal{D}_0, \mathcal{P}, \Gamma_{\mathcal{C}}, \Pi, \mathcal{L})$ where

Algorithm 6.1 Algorithm to construct the discrete transition system

Inputs: state space partitions Q, dynamics f , control inputs $u_e(t)|_P$ and $u_e(t)|_{EA}$

1. Initialize transition system $\mathcal{B} = (\mathcal{Q}, \mathcal{Q}_0, \mathcal{P}, \Gamma_\mathcal{B}, \Pi, \mathcal{L})$ where
   $\mathcal{Q} = \{\overset{\circ}{q} \mid \overset{\circ}{q} \in \mathcal{F}(q), \forall q \in Q\}$, $\mathcal{Q}_0 \subset \mathcal{Q}$, $\mathcal{P} = \{P, EA\}$, $\Pi = \{\Pi_V, \Pi_H, \Pi_\Theta\}$, $\Gamma_\mathcal{B} = \{\}$, $\mathcal{L} = \mathcal{H}$
2. **for** k **in** $\{P, EA\}$
3. **for** $\bar{q}$ **in** Q
4. **for** $\overset{\circ}{q}_i$ **in** $\mathcal{F}(\bar{q})$
5. **for** $\overset{\circ}{q}_j$ **in** $\mathcal{F}(\bar{q})$
6. **if** $(isReach(\overset{\circ}{q}_i, k, \overset{\circ}{q}_j))$
7. //Add transition to discrete system $\mathcal{B}$ if valid transition exists.
8. $\Gamma_\mathcal{B} = \Gamma_\mathcal{B} \cup \{(\overset{\circ}{q}_i, k, \overset{\circ}{q}_j)\}$
9. **for** $\bar{q}_i$ **in** Q
10. **for** $\bar{q}_j$ **in** Q
11. **for** $\overset{\circ}{q}_m$ **in** $\mathcal{F}(q_i)$
12. **for** $\overset{\circ}{q}_n$ **in** $\mathcal{F}(q_j)$
13. **if** $(\overset{\circ}{q}_m \equiv \overset{\circ}{q}_n)$
14. // Add transitions between facets that are common to adjacent cells
15. $\Gamma_\mathcal{B} = \Gamma_\mathcal{B} \cup \{(\overset{\circ}{q}_m, k, \overset{\circ}{q}_n)\}\ \forall k \in \mathcal{P}$
16. **Return** $\mathcal{B}$

$\mathcal{D} = \mathcal{Q} \times \mathcal{S}$, $\mathcal{D}_0 = \mathcal{Q}_0 \times \mathcal{S}_0$. Let $d_i, d_j \in \mathcal{D}$ where $d_i = (\overset{\circ}{q}, s)$, $d_j = (\overset{\circ}{q}', s')$. Then, $(d_i, p, d_j) \in \Gamma_\mathcal{C}$ if and only if $(\overset{\circ}{q}, p, \overset{\circ}{q}') \in \Gamma_\mathcal{B}$ and $(s, \sigma, s') \in \mathcal{T}$, where $p = \mathcal{G}(s)$ and $\sigma = \mathcal{L}(\overset{\circ}{q}')$. 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 $\overset{\circ}{q}$ 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 is used for model checking process it depicts the behavior of the aircraft during takeoff as governed by FSAM. The goal in verification is to ensure that model $\mathcal{C}$ satisfies the requirements imposed on takeoff in Section 6.3(6.3.2). Fig 6.4 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.

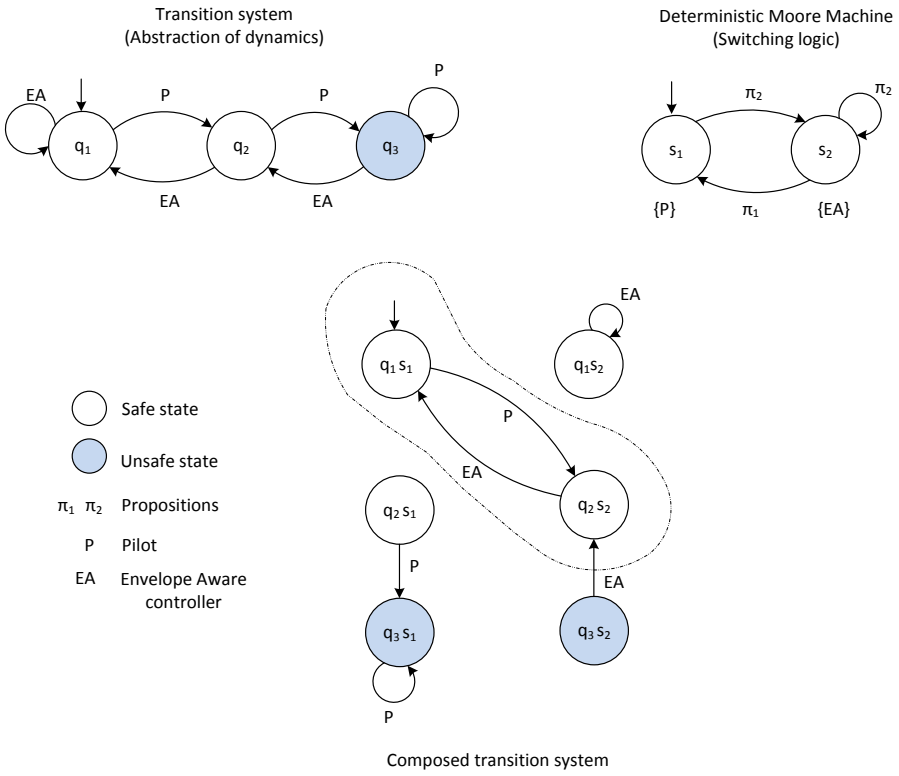

Figure 6.4: Composition of a transition system with a DMM

6.4.4 Model Checking

The requirements (Φ) for model checking expressed in LTL with the propositions defined in the previous section are as follows:

$$\begin{aligned}
\Phi_1 := \quad & \square((\theta \geq \theta_{ng}) \rightarrow (V \geq V_R)) && = \square((\pi_{\theta 4} \vee \pi_{\theta 5}) \rightarrow (\pi_{v4} \vee \pi_{v5} \vee \pi_{v6})) \\
\Phi_2 := \quad & \square((h < h_{lof}) \rightarrow (\theta < \theta_{tail})) && = \square((\pi_{H1} \vee \pi_{H2}) \rightarrow (\pi_{\theta 1} \vee \pi_{\theta 2} \vee \pi_{\theta 3} \vee \pi_{\theta 4})) \\
\Phi_3 := \quad & \square((h \geq h_{lof}) \rightarrow (\theta \geq \theta_0)) && = \square(\neg(\pi_{H1} \vee \pi_{H2}) \rightarrow \neg\pi_{\theta 1}) \\
\Phi_4 := \quad & \square((h \geq h_{obs}) \rightarrow (V \geq V_2)) && = \square((\pi_{H4}) \rightarrow \pi_{v6})
\end{aligned} \tag{6.4}$$

The composed transition system $\mathcal{C}$ and requirements Φ are input into the SPIN model checker. Fig 6.5 illustrates an overview of model checking. If the model satisfies all requirements, the verification is considered complete. If violations exist, analysis of each counter example 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 also distinguish counter examples that could be false positives. Most often, false positives

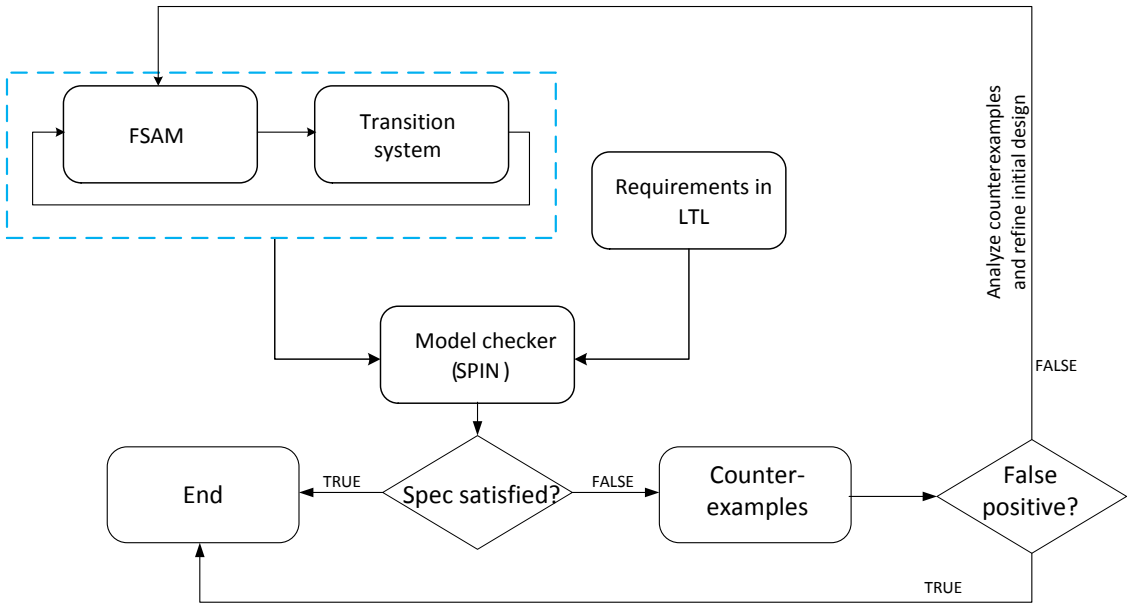

Figure 6.5: Model checking process

are artifacts of the abstraction technique itself so it is possible to use these counter examples to refine the abstractions in a manner that eliminates false positives [130].

Using the above model checking approach, requirements Φ_1 and Φ_2 were violated with the baseline DMM shown in Fig 6.2. In other words, the underlying FSAM logic could not prevent premature rotations and tail strikes.

6.5 Refinement of FSAM

As discussed above the model checker revealed that requirements Φ_1 and Φ_2 were violated in $\mathcal{C}$. Three causes were identified: (i) Specific pilot behaviors could result in the violation, (ii) The EA controller could be poorly designed and/or inadequate to deal with the off-nominal conditions (iii) The switching logic (FSAM DMM) might be incorrect/incomplete. According to the system dynamics in (6.1), Φ_1 could be violated if the pilot rotates the nose of the aircraft in the $V_{mcg} \leq V < V_1$ airspeed range. This is due to the fact that protection against premature rotation while in the $V_{mcg} \leq V < V_1$ 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 $V_R \leq V < V_{lof}$. After analyzing the above counterexamples, appropriate changes to FSAM's DMM $\mathcal{A}$ were made. These changes are highlighted in Fig 6.6 and were also carried into our archival FSAM [85] DMM specification. The updated DMM protects from

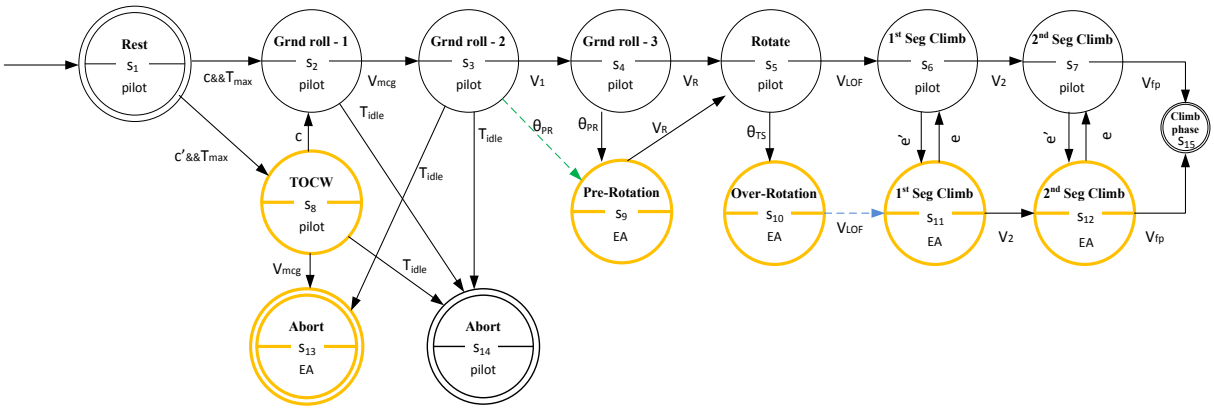

Figure 6.6: Revised FSAM DMM

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.

A Monte Carlo analysis was performed using the full nonlinear equations of motion describing takeoff dynamics. Details of the Monte-Carlo simulations can be found in Appendix B.2. Fig 6.7 illustrates the aircraft responses after several Monte Carlo trials with the original uncorrected FSAM DMM. Fig 6.7 shows many instances of tail strikes (i.e., $\theta \geq \theta_{tail}$ & $h < h_{lof}$) 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 (Fig 6.8).

6.6 Validation

The Monte Carlo simulations discussed in the previous section confirm that the refinements made in response to the counter examples 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, 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 mission. 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

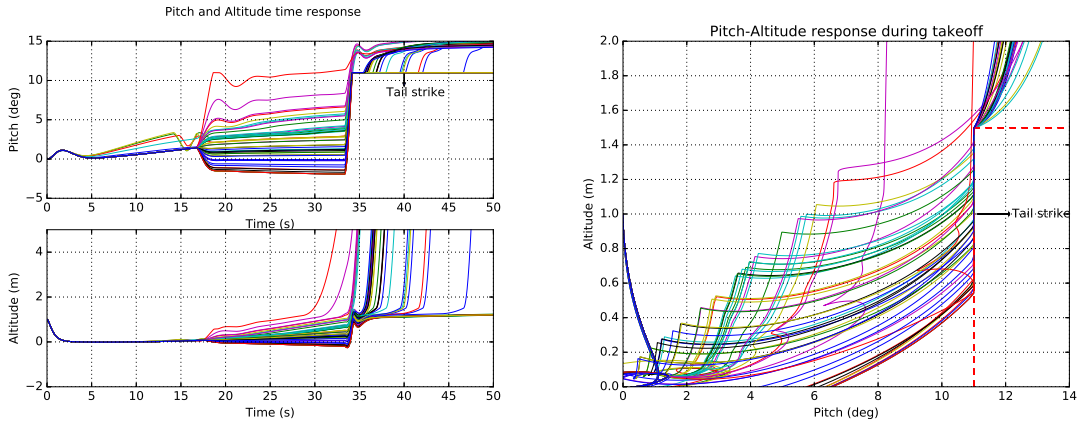

Figure 6.7: Monte Carlo simulations of the takeoff phase with original FSAM DMM

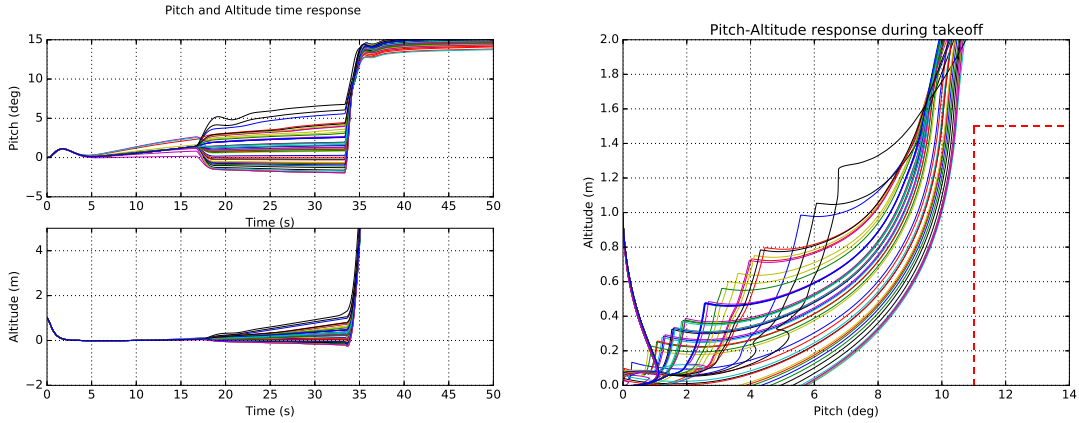

Figure 6.8: Monte Carlo simulations of the takeoff phase with revised FSAM DMM

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 [4]. 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 prematurely. 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 6.6 would then potentially prevent the pilot from maintaining acceptable nose wheel loading. This may result in the nose gear digging into

[4] GA is covered in a separate FAR section, but tail strike is still an issue

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 6.6) to ensure satisfaction of the 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 where it is actually required. As this example illustrates, the FSAM verification process must always take into account pertinent operational requirements in addition to baseline FARs.

6.7 Discussion

This chapter 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 over-approximation on which model checking was performed. The simplified dynamics presented in this work 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 counter examples 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 6.1 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 which 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 counter examples during model checking, due to nondeterminism induced in the abstract model, that were determined to be false positives.

It is also possible to automate the FSAM logic refinement process using tools such as CEGAR (see [130] and references therein). 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 [45, 85] is a higher order non-linear 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 non-linear dynamics within the proposed framework and use methods described in [50, 110–113, 129, 131] within Algorithm 6.1.

6.8 Conclusions

This work 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 system is verified via three main steps: (i) Select an abstract representation of the underlying state space, (ii) Construct a discrete transition system which over-approximates the reachable states under the various control authorities, (iii) 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 non-linear dynamics to eliminate false positives. This work 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 must consider different scenarios such as rejected takeoffs, engine failure scenarios, and crosswind conditions. In such cases, the simplified models described in this work 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, longitudinal and cross track position to capture safe versus unsafe states. It is important to recognize that FSAM only activates when safety is verifiable, 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 work are verifiable and thus ultimately certifiable using current regulation practices.

CHAPTER 7

Temporal Logic Falsification via Guided Monte-Carlo Search

7.1 Introduction

The pilot and nominal autopilot, envelope-aware controller, and FSAM can be collectively viewed as a hybrid control system. Verifying the correctness of this hybrid system with respect to formal requirements using conventional model checking approaches described in the previous chapter can be tedious because it is hard to explicitly specify a pilot model that captures all possible behaviors. Additionally the aircraft dynamics are generally non-linear, aerodynamic parameters are uncertain and subject to changes due to varying environmental conditions, and the envelope-aware controller adapts over time to changing aircraft dynamics.

Another method for verifying the FSAM switching protocol is the notion of "falsifying" system requirements. Instead of explicitly searching for traces that violate system requirements in a discrete abstraction as in the previous chapter, one can search for sequences of inputs yielding trajectories that violate system requirements. Searching for possible pilot inputs that violate system constraints is a convenient method to verify requirements given that an explicit model of pilot behavior is hard to obtain. A straightforward way to accomplish this is via randomized testing, i.e. Monte Carlo simulations. The underlying system can be considered a black box that maps input signals to output signals. The inputs can then be chosen from a suitable distribution to yield output signals that violate system requirements. Such randomized testing can address the concerns listed above but can require a large number of trials to identify a falsifying trajectory. The key to finding falsifying trajectories quickly and efficiently is to sample from a distribution that yields falsifying system behaviors with high probability.

This chapter focuses on the use of a rare-event simulation technique called the cross-entropy method [132, 133] to guide the search for pilot behaviors that can falsify specified

requirements. The falsifying trajectories can then be analyzed to aid the refinement of the FSAM switching protocol which is either specified by finite state machines or via an MDP policy.

7.2 Preliminaries

7.2.1 Robustness

To falsify a given requirement, we would like to find trajectories of the system that are close to violating the given requirements. In other words, we would like to determine the robustness of trajectories with respect to requirement violations. In this context, robustness is defined as follows:

Let $y \in Y$ be a point and let $\mathcal{S} \subset Y$ be a set defined by system requirements. Here, the set $\mathcal{S}$ could denote unsafe states that must be avoided at all times. Let d be a distance metric on Y. Then, the signed distance from the point y to the set $\mathcal{S}$ is given as:

$$\mathbf{Dist}_d(y, \mathcal{S}) = \begin{cases} -\inf \left\{ d(y, y') \mid y' \in \mathcal{S} \right\} & \text{if } y \notin \mathcal{S} \\ \inf \left\{ d(y, y') \mid y' \in Y \backslash \mathcal{S} \right\} & \text{if } y \in \mathcal{S} \end{cases} \tag{7.1}$$

Robustness of a trajectory parameterized by an initial condition $\vec{x_0}$ and a parameter $\vec{v}$ is defined as

$$R(\vec{x_0}, \vec{v}) = \inf \left\{ \mathbf{Dist}_d(y, \mathcal{S}) \mid y \in \vec{y} : [0, t_f] \to Y \right\} \tag{7.2}$$

If the robustness value is zero, then the smallest perturbation of the trajectory may result in the trajectory violating the requirement. Negative robustness values indicate requirement violation.

7.2.2 Cross Entropy Method

This section first provides a summary of the cross entropy method for falsifying temporal properties and then outlines the algorithm used for FSAM [132, 133]. As described previously, we would like to sample inputs from a distribution Ω that generates trajectories minimizing robustness. A suitable representation for Ω that satisfies the above criteria is given below:

$$\Omega(\vec{x_0}, \vec{v}) = \frac{1}{W} \mathrm{e}^{-kR(\vec{x_0}, \vec{v})} \tag{7.3}$$

where W normalizes the total mass of the distribution. Note that R is unknown for a given $\vec{x_0}$ and $\vec{v}$ a priori and hence it is not possible to sample from Eqn 7.3. Alternately, one can start from a family of distributions P_ϵ (parameterized by ϵ) and find an ϵ that yields a distribution close to Ω.

The similarity between distributions is defined by Kullback-Liebler (KL) divergence [133]: Let $p()$ and $q()$ denote two distributions over some set of support S, such that $\forall x \in S$, $p(x) \neq 0, q(x) \neq 0$. Then, the KL divergence is given by:

$$\mathcal{D}(p,q) = \mathbb{E}_p\left[\log \frac{p(x)}{q(x)}\right] = \int_{x \in S} \log \frac{p(x)}{q(x)} p(x) dx \tag{7.4}$$

where $\mathbb{E}_p$ denotes the expectation with respect to distribution p. Note that $\mathcal{D}(p,q) \neq \mathcal{D}(q,p)$. In this work, we seek the ϵ that minimizes $\mathcal{D}(\Omega, P_\epsilon)$. This process is called *tilting*. Let w denote the tuple $(\vec{x_0}, \vec{v})$.

$$\epsilon = \arg\min_\epsilon \mathcal{D}(\Omega, P_\epsilon) \tag{7.5}$$

$$= \arg\min_\epsilon \left\{ \int \log \frac{\Omega(w)}{P_\epsilon(w)} \Omega(w) dw \right\} \tag{7.6}$$

$$= \arg\min_\epsilon \left\{ - \int \Omega(w) \log P_\epsilon(w) dw \right\} \tag{7.7}$$

Note that the integral term in Eqn 7.7 denotes $\mathbb{E}_\Omega[\log P_\epsilon]$. Evaluating this term requires knowledge of $R(\vec{x_0}, \vec{v})$. Furthermore, empirical evaluations require samples from unknown distribution Ω. To overcome this difficulty, $\mathbb{E}_\Omega[\log P_\epsilon]$ can be evaluated using samples from a known distribution using importance sampling [133]. Thus, Eqn 7.7 becomes

$$\epsilon = \arg\min_\epsilon \left\{ - \int \frac{\Omega(w)}{P_*(w)} P_*(w) \log P_\epsilon(w) dw \right\} \tag{7.8}$$

$$= \arg\max_\epsilon \left\{ \mathbb{E}_{P_*}\left[\frac{\Omega(w)}{P_*(w)} \log P_\epsilon(w) \right] \right\} \tag{7.9}$$

Here P_* is a known distribution from which samples of $(\vec{x_0}, \vec{v})$ can be drawn. Thus, the expectation in Eqn 7.9 can now be evaluated empirically as follows:

$$\epsilon = \arg\max_\epsilon \left\{ \frac{1}{m} \sum_{i=1}^{m} \gamma_i \log P_\epsilon(w) \right\} \tag{7.10}$$

where $\gamma_i = \frac{\Omega(w_i)}{P_*(w_i)}$ is the likelihood ratio. The selection of the importance sampling dis-

tribution P_* and the algorithm for evaluating Eqn 7.10 is outlined below. The following algorithm is referred to as the *variance minimization* algorithm [133].

- Draw N samples of $(\vec{x}_0, \vec{v})$ from a distribution defined by the current parameter set ϵ_n (i.e. $P_* = P_{\epsilon_n}$).
- For each sample propagate system dynamics according to $(\vec{x}_0, \vec{v})$ and compute $R(\vec{x}_0, \vec{v})$. Note, $\vec{x}_0$ defines the initial condition and $\vec{v}$ denotes the control parameters required to generate a trajectory. Let $(\vec{x}_0, \vec{v})_0, \ldots, (\vec{x}_0, \vec{v})_N$ be the samples sorted in descending order of their Ω values.
- Choose the top m samples.
- The old distribution is 'tilted' to obtain a new distribution by minimizing the KL divergence:

$$\epsilon_{n+1} = \arg\max_{\epsilon} \left\{ \frac{1}{m} \sum_{i=1}^{m} \gamma_i \log P_\epsilon(w) \right\} \tag{7.11}$$

  Note that $\gamma_i = \frac{\Omega(w_i)}{P_{\epsilon_n}(w_i)}$ can be evaluated for each sample up to some fixed but known positive scaling factor W.
- Repeat the above steps until a suitable convergence criteria is satisfied. This work uses the average robustness across all simulations as the convergence criteria.

If we assume that the distribution P_ϵ is Gaussian and $\epsilon = (\mu, \sigma^2)$ denotes the mean and variance of the distribution respectively, then the required $\epsilon = (\mu, \sigma^2)$ in Eqn 7.11 is given by [133]:

$$\hat{\mu} = \frac{\sum_{i=1}^{m} \gamma_i w_i}{\sum_{i=1}^{m} \gamma_i} \tag{7.12}$$

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{m} \gamma_i (w_i - \hat{\mu})^2}{\sum_{i=1}^{m} \gamma_i} \tag{7.13}$$

7.3 Application to FSAM

Consider the following description of the Flight Safety Assessment and Management (FSAM) problem.

$$x_{k+1} = \mathcal{F}(x_k, u_k) \tag{7.14}$$

$$u_k = \begin{cases} u_k^p \; if \; c_k = p \\ u_k^a \; if \; c_k = a \end{cases} \tag{7.15}$$

$$c_k = \mathcal{G}(x_k, u_k^p, u_k^a) \tag{7.16}$$

Above, $x \in \mathbb{R}^n$ describes the system state, $u \in \mathbb{R}^m$ is a control input, $u_k \in \mathbb{R}^m$ denotes the control inputs where u^p, u^a describes the control input of the pilot and the safety controller respectively, $c \in \{p, a\}$ describes the control authority, $\mathcal{F}$ describes the evolution of the continuous states, and $\mathcal{G}$ describes a strategy to select the control mode such that loss of control situations are mitigated.

The *goal* is to find the sequence of pilot inputs $u_k^p, k = 1, \ldots, n$ that leads to LOC situations. In other words, given a strategy $\mathcal{G}$ to select the current control authority c_k, what sequence of pilot inputs u^p result in trajectories with low robustness? In this work we focus on using the cross-entropy based random sampling technique to find an input sequence that can falsify a given requirement.

7.4 Falsification of Requirements for the Takeoff FSAM System

Let $\mathcal{F}$ in Eqn 7.14 denote aircraft takeoff dynamics. A complete description of the nonlinear aircraft dynamics for takeoff and the envelope-aware control laws can be found in Appendix A-B. $\mathcal{G}$ in Eqn 7.16 denotes the switching strategy imposed by FSAM. Let $[0, t_f]$ denote the duration of the takeoff phase. Our goal is to find the sequence of pilot inputs $\{u_0, u_1, \ldots, u_n | u_k \in \mathcal{R}^m\}$ that can lead to requirement falsification under the original FSAM DMM formulation illustrated in Fig 6.2.

Let $\mathcal{U}$ denote the set of all possible input sequences. Sampling a sequence of inputs from $\mathcal{U}$ arbitrarily may require a large number of trials, especially if the sequence length n is very large, to find a falsifying trajectory. This process is therefore inefficient. In this work, we choose a parametric representation for $\mathcal{U}$ that can capture a wide range of pilot behaviors. The resulting parameter space has fewer dimensions than $\mathcal{U}$; hence it is easier to

sample from a suitable distribution that yields falsifying trajectories with high probability.

The pilot's control surface deflection and throttle commands are represented as a PD control law:

$$\delta_e = k_{p_e}(\theta - \theta_{ref}) + k_{d_e}\dot{\theta} \tag{7.17}$$

$$\delta_a = k_{p_a}(\phi - \phi_{ref}) + k_{d_a}\dot{\phi} \tag{7.18}$$

$$\delta_r = k_{p_r}(\psi - \psi_{ref}) + k_{d_r}\dot{\psi} \tag{7.19}$$

$$\delta_t = \tau_{ref} T_{max} \tag{7.20}$$

where $\delta_e, \delta_a, \delta_r$ represent the control surface deflections of the elevator, aileron and rudder, respectively. δ_t denotes the thrust input where $t_{ref} \in [0, 1]$ is the reference thrust command and T_{max} is the maximum thrust available. We assume symmetric thrust is applied by a multi-engine aircraft in this formulation. $\theta_{ref}, \phi_{ref}, \psi_{ref}, \tau_{ref}$ denote the pitch, roll, heading and thrust reference commands that the pilot follows while executing the takeoff sequence. The k_{p_i}, k_{d_i} values represent proportional and derivative gains for each control effector $i \in \{e, a, r\}$. To model a wide range of behaviors using Eqns 7.17-7.20 the reference commands are chosen as follows.

The takeoff sequence from $[0, t_f]$ is divided into N disjoint sub-intervals: $[0, t_1), [t_1, t_2)$, $\ldots$, $[t_{N-1}, t_f)$. For each sub-interval, the reference commands are drawn from a Gaussian distribution. In this work, the takeoff interval is 3 sub-intervals. For example, the pitch reference command is sampled as follows

$$\theta_{ref} = \begin{cases} \theta_1 & t \in [0, t_1) \\ \theta_2 & t \in [t_1, t_2) \\ \theta_3 & t \in [t_2, t_f) \end{cases} \tag{7.21}$$

where $\theta_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. The pilot's pitch control inputs for takeoff are characterized by a total of 6 parameters. The ailerons, rudder and thrust reference commands can be parameterized in a similar fashion.

We first illustrate the robustness of the original Moore machine formulation for FSAM with respect to the tail strike requirement: $\square(h < h_0 \rightarrow \theta < \theta_0)$. This requirement must be satisfied to prevent tail strikes during takeoff. Robustness of trajectories are defined with respect to the unsafe set defined by the region where $\theta > \theta^*$ and $h < h^*$. Here θ^*, h^* represent the pitch attitude and altitude at which tail strikes can occur. We first start with the initial distribution shown in Fig 7.1-(a). Then using the cross-entropy method described above, we search for the distribution which reduces the robustness of the trajectories with

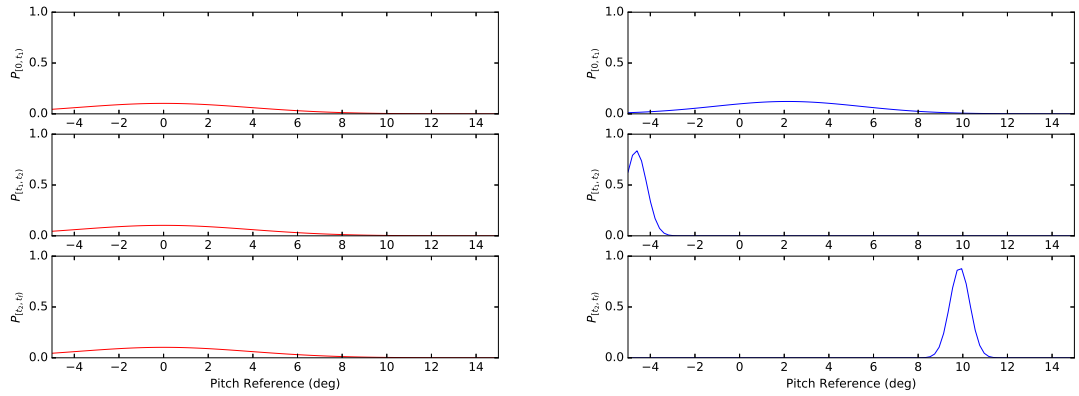

Figure 7.1: (a) Initial pitch distributions (b) Pitch distributions after 6 iterations of the cross entropy method.

respect to the specified requirement. Fig 7.2 (a) summarizes the performance of the cross-entropy method across several iterations. Each iteration consists of 1000 Monte Carlo trials. Fig 7.2-(a) illustrates the average robustness for each iteration and their corresponding standard deviations. Fig 7.1-(b) illustrates the final distributions on the pitch reference commands at the end of 6 iterations of the cross entropy method. Fig 7.2-(b) illustrates the falsifying trajectories in the $\theta - H$ space. The dashed red lines denote the tail strike constraint. Note that toward the end of the takeoff phase, high pitch reference commands are sampled resulting in trajectories with tail strikes during rotation.

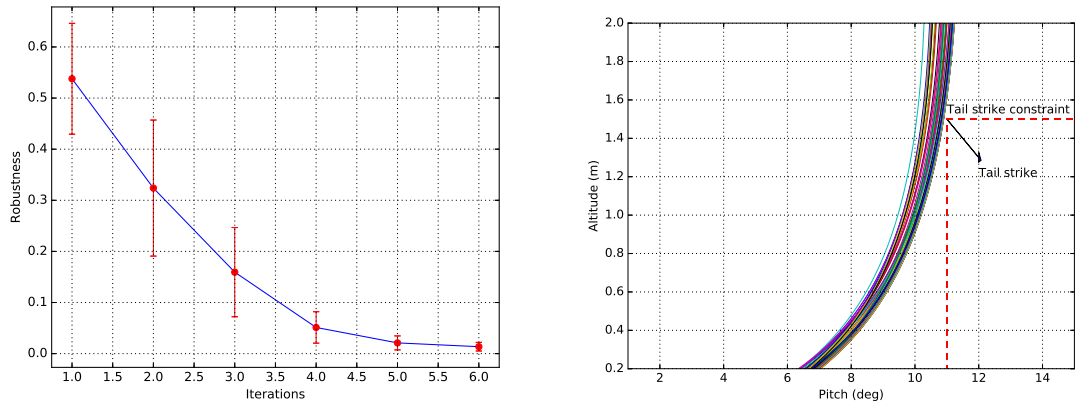

Figure 7.2: (a) Average robustness, (b) Falsifying trajectories obtained from the final pitch distributions

The falsifying trajectories in Fig 7.2-(b) were due to a missing transition (also identified in Chapter 6) in the original Moore machine formulation for FSAM. After incorporating

the missing transition, the cross-entropy analysis was performed again. Fig 7.3-(a) illustrates the average robustness across several iterations of the cross-entropy method with the revised DMM formulation. The trajectories drawn from this new distribution are shown in Fig 7.3-(b). The cross-entropy formulation considered in this work independently samples the parameters describing the pilot model. It is also possible to sample the parameters taking into consideration the correlation between parameters (see [132] for more details).

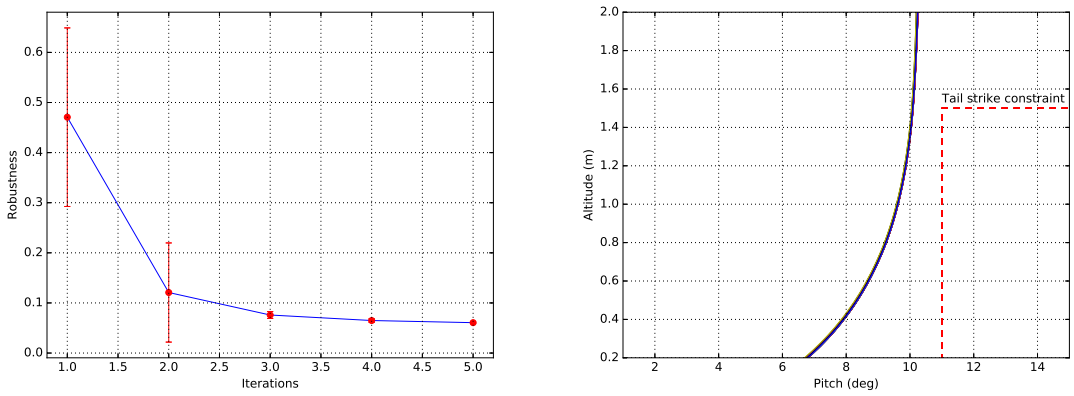

Figure 7.3: (a) Average robustness, (b) Falsifying trajectories obtained from the final distribution

Next we illustrate the robustness of the revised Moore machine formulation to runway overrun constraints expressed in LTL as follows: $\square((V \geq V_2) \wedge (x \geq R_{max}) \rightarrow (h > h_{obs}))$. The unsafe set is described by the region $V < V_2, x > R_{max}, h < h_{obs}$ where V_2 represents the takeoff safety V-speed, R_{max} represents the available runway length and h_{obs} represents the runway obstacle height (see dashed lines in Fig 7.5-(b)). Fig 7.4-(a) illustrates the initial distribution on the thrust reference command. Fig 7.4-(b) illustrates the final falsifying distributions obtained from the cross-entropy method. The average robustness values across several iterations of the cross-entropy method are illustrated in Fig 7.5-(a) while Fig 7.5-(b) illustrates trajectories that overshoot the runway and hence violate the given requirement. Note that in the beginning of the takeoff sequence, the reference thrust command is sampled from the takeoff thrust range. Midway through the takeoff roll, past the decision point, the takeoff is rejected so that the ensuing trajectory result in runway overruns. The violation of the runway overrun constraint is attributed to the inappropriate modulation of throttle δ_t inputs during the ground roll. Recall that the FSAM DMMs were formulated under the following assumptions: (i) pilot would not exhibit inappropriate behavior and (ii) at least one engine will function properly. Consequently, it is not surprising that the current DMM formulation violates the given requirement.

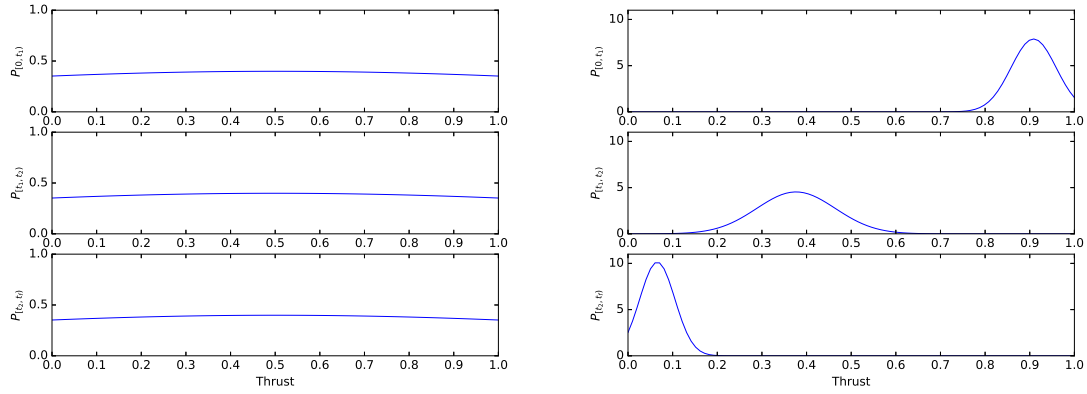

Figure 7.4: (a) Initial distribution (b) Final distribution after 7th iteration of cross entropy method

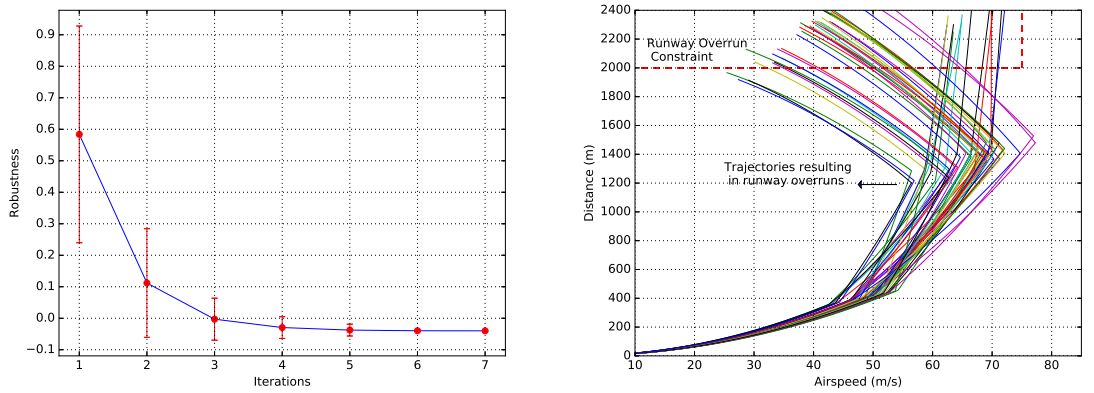

Figure 7.5: (a) Average robustness, (b) Falsifying trajectories after 7 iterations of the cross entropy method

7.5 Discussion

The cross-entropy examples presented in this chapter divided the takeoff duration into three intervals and searched for the reference commands $\theta_{1,2,3}$ and $\tau_{1,2,3}$ corresponding to each time interval. In future work, other parameters such as the duration of each interval and the proportional and derivative pilot control model gains can also be included as search parameters in the cross-entropy framework. The number and duration of the takeoff time interval set can also be varied to capture specific pilot behaviors.

In several examples, the variance of the parameter distributions converged to zero. In these cases, instead of searching for a distribution over trajectory parameters, one can treat the problem deterministically and directly search for the reference command values that

yield trajectories with low robustness using numerical methods (e.g. shooting methods). Identification of such cases might reduce computation time in future case studies.

7.6 Conclusions

The cross entropy method presented in this chapter guided Monte Carlo trials to sample from distributions that yield trajectories most likely to violate system requirements. Conventional model checking techniques only verify that execution traces satisfy a given requirement. Analysis of robustness provides more information about system proximity to a constraint boundary. Monte Carlo-based requirement falsification was performed on the full nonlinear takeoff dynamics model with the concurrent execution of the two FSAM DMM formulations (i.e. longitudinal and lateral DMMs).

CHAPTER 8

Conclusions and Future Work

8.1 Conclusions

This thesis introduced the Flight Safety Assessment and Management (FSAM) system. FSAM monitors flight conditions for loss of control risk and makes control authority switching decisions that mitigate risk. FSAM switches between a nominal crew-directed FMS and the Envelope-Aware Flight Management System (EAFMS) capable of adapting to anomalies and recovering from loss of control situations. Two main approaches were explored in this work to formulate FSAM. The first approach relied on manually engineered switching protocols for FSAM as Deterministic Moore Machines (DMM). The second approach relied on formulating FSAM as a Markov Decision Process (MDP). Finally, methods to verify FSAM state machines and MDP policies were presented.

The contributions of this work are: (i) takeoff envelopes to aid control mode switching decisions, (ii) a DMM formulation to address common takeoff related LOC scenarios, (iii) takeoff-specific and in-flight icing-specific MDP formulations, (iv) an online implementation for FSAM MDP, (v) verification methods to ensure FSAM satisfies safety requirements, and (vi) evaluations of FSAM showing it could have prevented the accidents occuring in several real world case studies.

The innovations of this work are: (i) a control authority switching mechanism that is generalizable across a suite of loss of control (LOC) scenarios, (ii) takeoff DMM formulations based on V-speed transitions that manage state space complexity, improve readability and promote flight crew understanding, (iii) a compact abstraction for MDP states enabling tractable solutions, (iv) constrained MDP formulations to eliminate manual tuning of policies, and (v) a compositional approach to verify FSAM DMMs.

The DMM and MDP FSAM formulations presented in this work have pros and cons. Finite state machines require a system designer to directly define the states where control mode overrides should occur based on knowledge of reachable or recoverable set bound-

aries defining thresholds that trigger control mode transitions. Altering transitions in finite state machines to accommodate reachable set boundaries and flight crew preferences is straightforward thus a pro of the DMM. Precise control over the transition timings is also straightforward with timed automata extensions. However, while the finite state machines offer a computationally tractable and verifiable means of realizing FSAM, engineers must manually envision and create DMM states and transitions for all possible exception events and all possible exception event combinations when constructing control mode switching machines.

In contrast, the decision theoretic approach using MDPs doesn't require the system designer to reason directly about control mode overrides. Instead, the key focus of the MDP approach is on selecting a suitable representation for the MDP states and actions, defining a reward function that encodes the desired goals, and computing transition probabilities. Domain knowledge can be exploited to compactly represent the MDP states to enable a tractable solution. The optimal policy prescribes the control mode switching decision in each MDP state. However, the obtained policy is only as good as the underlying transition probability models, reward and state formulations, and in this work, substantial tuning of weights was typically required to obtain desirable and safe policies.

Verification is crucial to ensure control mode switching strategies defined by either DMM or MDP FSAM realizations do not violate critical safety requirements. Refinements to the DMM/MDP formulations can be guided by verification. Verifying FSAM with respect to safety and operational requirements ensures an FSAM capabilities are applicable across the wide spectrum of missions and LOC risk scenarios.

8.2 Future Work

An ideal FSAM formulation should be capable of making risk mitigating control authority or mode selections under any loss of control situation. An FSAM module applicable to a set of elevated takeoff risks was primarily considered in this work, along with a specific extension to in-flight icing. General FSAM modules applicable to other phases of flight (i.e. climb, cruise, approach and landing) are required to ensure LOC prevention and recovery across the entire flight regime. The progression of flight phases motivates the need for efficient algorithms to safely enable transition between the FSAM modules. Within each phase, further decomposition based on an elevated set of risk factors (e.g engine failure, actuator failure, cabin de-pressurization) must be considered. A hierarchical suite of DMMs/MDPs applicable under specific LOC scenarios can be indexed as a (small) database online based on phase of flight and the specific elevated LOC risks actually ob-

served. The database approach requires a verified matching algorithm to efficiently select the appropriate DMM or MDP policy.

The development of FSAM in this work was simplified by several assumptions. Imposing assumptions on FSAM reduces system complexity which in turn facilitates verification. However, for real world applications, algorithms must be developed to recognize scenarios where the underlying assumptions are violated so that no FSAM database policy is applicable. An initial implementation of FSAM must recognize situations in which it has no applicable DMM/MDP solution, in which case FSAM would simply remain passive (no FSAM overrides are issued). Of course the crew could still activate adaptive EAFMS capabilities in such scenarios. This approach has been adopted by industry to-date with specific LOC prevention capabilities (e.g. envelope protection, Traffic Collision Avoidance, Runway Overrun Protection, Ground Collision Avoidance systems) that deactivate in situations where their correct operation is no longer assured.

To achieve a generalized FSAM capability across the suite of envisioned loss of control situations, assumptions must be relaxed. For example, actuators and sensors might fail during takeoff. A fully-functional takeoff FSAM would at least incorporate rules to manage single sensor system failures along with power system and control surface failures that might occur during takeoff. Further, the MDP formulations in this work assumed complete observability of the underlying states. In practice, the state feature estimates are subject to uncertainty, and sensor failures might render states normally considered observable as unobservable. Extending the MDP to a Partially Observable Markov Decision Process (POMDP) formulation can help make decisions amidst uncertainty in state estimates at the cost of additional computational complexity.

While a more general formulation was envisioned, the actual FSAM DMM and MDP realizations in this work focused on control authority decisions based on aircraft dynamics, control effector, and limited system health state features. This constraint allowed transition probabilities of the aircraft dynamics states to be estimated using physics-based Monte Carlo simulations. A comprehensive MDP formulation requires incorporating other state features such as vehicle health, flight crew and environment characteristics. Adequately capturing the effect of state features such as vehicle health, crew behavior and weather phenomena can be difficult with physics based models. In future work, suitable transition models may be constructed by mining long-term flight data. Targeted human subject experiments and more comprehensive environment data and models must also be incorporated.

Another challenge facing the MDP FSAM implementation was in selecting appropriate reward function objective weights. In future work, sensitivities of MDP policies to the underlying transition and reward function models must be characterized to understand the

robustness of the MDP policies to modeling errors. This dissertation has proposed DMM and MDP FSAM formulations along with tradeoffs between the two strategies. However, FSAM formulations to-date have been based on either the DMM or the MDP. Future work should consider mixed strategies where control mode switching is defined by a combination of DMMs and MDP policies to exploit the benefits of both approaches. Well-characterized LOC risk scenarios with manageable state-space complexity might be best handled with intuitive DMMs, whereas scenarios with complex interdependencies between state functions or scenarios where statistics might even adapt in real-time would be better accommodated with the MDP formulation.

This work used real-world aviation accidents supplemented by Monte Carlo simulations as case studies to evaluate FSAM state machines and MDP policies. NTSB accident and airline data must be mined in future work to assure FSAM either handles or remains passive across the suite of accident and exception scenarios previously encountered and documented. Another important avenue of research is the design of appropriate interfaces and warnings that communicate to the pilot the decisions made by FSAM. Though verification and validation techniques serve as an efficient augmentation to flight tests, human in the loop experiments will be necessary to understand how a typical flight crew would interact and react to override decisions made by FSAM. These human subject experiments and flight crew preferences can in turn be used to refine the FSAM formulations.

In summary, the Flight Safety Assessment and Management system introduced in this dissertation can mitigate loss of control risk which in turn will reduce risk of incidents and accidents. As autonomy and authority questions continue to emerge in manned and unmanned aviation, the FSAM foundation laid by this dissertation will inform the community regarding the control authority management and switching decisions essential to address in the coming decade and beyond.

APPENDIX A

Aircraft Takeoff Dynamics

This section describes aircraft dynamics for takeoff. Modeling ground roll dynamics of the aircraft requires knowledge of the reaction forces and moments exerted by the ground on the airframe [134] as well as aerodynamic forces that become significant as airspeed progressively increases during the takeoff roll.

Aircraft landing gear is modeled as a spring-mass-damper system for each assembly [135–137]. Based on knowledge of inertial position and velocity of the center of gravity (CG) and attitude of the aircraft, one can estimate the compression and rate of compression of the oleo struts and then compute the normal forces and moments exerted by the ground on the airframe.

Assuming that the three struts are exactly vertical, the normal force F_z exerted by the ground on the aircraft (expressed in the inertial frame) is given by

$$F_{z_i} = -K_i z_i - C_i \dot{z}_i \quad i = N_w, L_w, R_w \tag{A.1}$$

K_i and C_i are the spring constants and the damping coefficients of the nose, left and right oleo struts of the landing gear. z and $\dot{z}$ are the compression and rate of compression of the oleo struts expressed in the inertial frame. N_w, L_w and R_w represent the nose, left and right wheels. The gear model is shown in Fig A.1.

The wheels experience friction due to contact between the tire and runway surface. Longitudinal forces acting on the wheels are due to the longitudinal slip and the normal forces experienced by the wheels. The longitudinal slip ratio is given by [137]:

$$\sigma_s = \frac{V_x - \omega R_0}{V_x} \tag{A.2}$$

where V_x is the translational speed of the wheel in the longitudinal direction, ω is angular speed of the wheel, and R_0 is wheel radius including tire.

Coefficient of friction μ is related to the longitudinal slip ratio σ_s of the wheels by the

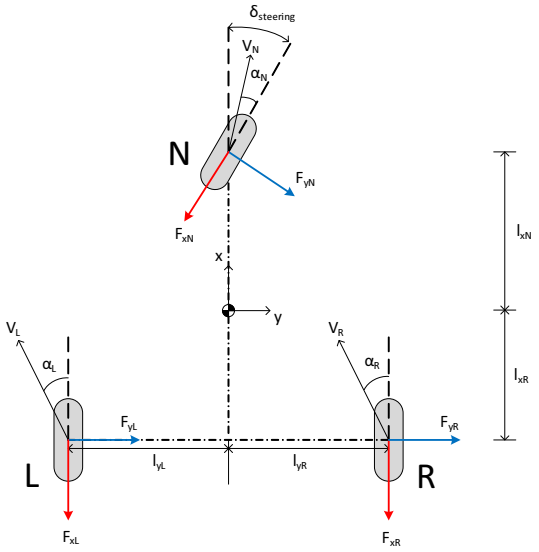

Figure A.1: Tri-cycle landing gear configuration

empirical formula known as the "magic formula" [137, 138].

$$\mu = \bar{D}\sin(\bar{C}\tan^{-1}(\bar{B}\sigma_s)) \tag{A.3}$$

Here $\bar{B}$,$\bar{C}$ and $\bar{D}$ are constants pertaining to the runway surface type. The longitudinal frictional forces F_x exerted by the ground on the wheel are given by:

$$F_{x_i} = \mu_i F_{z_i} \quad i = N_w, L_w, R_w \tag{A.4}$$

The wheels also experience side force F_y due to lateral slip of the wheels. Lateral slip ratio (α_s) is given by:

$$\alpha_{s_i} = \tan^{-1}\left(\frac{V_{y_i}}{V_{x_i}}\right), \quad i = L_w, R_w \tag{A.5}$$

$$\alpha_{s_{N_w}} = -\delta_{steer} + \tan^{-1}\left(\frac{V_{y_{N_w}}}{V_{x_{N_w}}}\right) \tag{A.6}$$

where V_x, V_y are the translational and lateral wheel speeds given by

$$V_x = u - rl_y \tag{A.7}$$

$$V_y = v + rl_x \tag{A.8}$$

Here (u,v) are the (x,y) components of aircraft velocity in the body frame, respectively. l_{yi} and l_{xi} are distances of the wheels from the CG as shown in Fig A.1. r is the yaw

angular rate. δ_{steer} is the nose wheel steering angle. Note that V_x, V_y are obtained using the Transport theorem and the fact that the pitch and roll angular rates are negligible during the takeoff ground roll. The side force F_y is given by [135, 136]

$$F_{y_i} = \frac{2F_{y_{max_i}}\alpha_{s_{opt_i}}\alpha_{s_i}}{\alpha_{s_{opt_i}}^2 + \alpha_{s_i}^2}, \;\; i = N_w, L_w, R_w \tag{A.9}$$

Here $F_{y_{max}}$ is the maximum attainable side force at the optimal slip angle α_{opt}. $F_{y_{max}}$ and α_{opt} are experimentally-derived parameters. In this work, we use the side force model given in [135, 136]:

$$F_{ymaxN_w} = -3.53\times10^{-6}F_{zN_w}^2 + 8.33\times10^{-1}F_{zN_w} \tag{A.10}$$

$$F_{ymaxL_w,R_w} = -7.39\times10^{-7}F_{zL_w,R_w}^2 + 5.11\times10^{-1}F_{zL_w,R_w} \tag{A.11}$$

$$\alpha_{s_{opt_N}} = 3.52\times10^{-9}F_{zN_w}^2 + 2.8\times10^{-5}F_{zN_w} + 13.8 \tag{A.12}$$

$$\alpha_{s_{opt_{L,R}}} = 1.34\times10^{-10}F_{zL_w,R_w}^2 + 1.06\times10^{-5}F_{zL_w,R_w} + 6.72 \tag{A.13}$$

The net ground reaction force components F_x, F_y and F_z can be computed as shown in Equations (A.1),(A.4) and (A.9). The moments M_x, M_y, M_z due to the reaction forces can be obtained by taking the product of the reaction forces and the respective moment arms about the aircraft center of gravity.

The net ground reaction forces and moments are transformed into the aircraft body frame. The transformed forces and moments can then be added to the conventional six degree of freedom aircraft equations of motion [70] to obtain the complete nonlinear set of equations that simulate the takeoff phase of flight. The takeoff equations of motion (expressed in body frame) are given by:

- Translational Momentum

$$\begin{aligned} m(\dot{u} - vr + wq) = &-(\sin\theta)mg - (\cos\beta)(\cos\alpha)\mathcal{D} \\ &+ (\sin\alpha)\mathcal{L} + (\cos\phi_T)F_T + F_{x_{gear}} \end{aligned} \tag{A.14}$$

$$m(\dot{v} + ur - wp) = (\sin\phi)(\cos\theta)mg - (\sin\beta)\mathcal{D} + F_{y_{gear}} \tag{A.15}$$

$$\begin{aligned} m(\dot{w} - uq + vp) = &(\cos\phi)(\cos\theta)mg - (\cos\beta)(\sin\alpha)\mathcal{D} \\ &- (\cos\alpha)\mathcal{L} - (\sin\phi_T)F_T + F_{z_{gear}} \end{aligned} \tag{A.16}$$

- Rotational Momentum

$$I_{xx}\dot{p} + (I_{zz} - I_{yy})qr - I_{xz}(\dot{r} + pq) = L_{aero} + L_{thrust} + L_{gear} \tag{A.17}$$

$$I_{yy}\dot{q} + (I_{xx} - I_{zz})pr + I_{xz}(p^2 - r^2) = M_{aero} + M_{thrust} + M_{gear} \tag{A.18}$$

$$I_{zz}\dot{r} + (I_{yy} - I_{xx})pq + I_{xz}(qr - \dot{p}) = N_{aero} + N_{thrust} + N_{gear} \tag{A.19}$$

- Wheel Dynamics

$$I_{w_N}\dot{\omega_N} = F_{x_N}R_{wheel_N} + \tau_{roll_N} \tag{A.20}$$

$$I_{w_L}\dot{\omega_L} = F_{x_L}R_{wheel_L} + \tau_{roll_L} + \tau_{brake_L} \tag{A.21}$$

$$I_{w_R}\dot{\omega_R} = F_{x_R}R_{wheel_R} + \tau_{roll_R} + \tau_{brake_R} \tag{A.22}$$

Here u, v and w represent the translational velocity in the aircraft body frame, p, q and r are body frame angular rates. ϕ, θ and ψ are the roll, pitch and yaw angles, $\mathcal{L}$, $\mathcal{D}$ are total lift and drag respectively, F_T is the total thrust force, and ϕ_T represents the angle, the thrust vector makes with the longitudinal axis. L_i, M_i and N_i are the roll, pitch and yaw moments where $i = aero$, $i = thrust$, $i = gear$ represent moments induced by aerodynamic, thrust and gear forces respectively. I_{xx}, I_{yy}, I_{zz} and I_{xz} are the moments of inertia of the aircraft, $I_w = \frac{m_w R_{wheel}^2}{2}$ is the moment of inertia of the wheel where m_w is the wheel mass and R_{wheel} is the radius of the wheel. $\tau_{roll} = \mu_r F_z R_{wheel}$ is the rolling resistance moment. μ_r is the rolling resistance coefficient. $\tau_{brake} = c_b \delta_B$ is the braking torque produced on the wheels due to the application of brakes δ_B. c_b is a scaling parameter chosen to convert the brake input δ_B to braking torque τ_{brake}. Eqns (A.20)-(A.22) model the effect of differential braking during the ground roll [134].

The numerical values for the various aircraft parameters are listed in Table A.1. The aerodynamic forces and moments in Eqns (A.14)-(A.19) are obtained from the NASA Generic Transport Model (GTM) [97]. The landing gear parameters are chosen to ensure that the spring mass damper model shown in Eqn (A.1) has sufficient damping characteristics. The friction parameters in Eqn (A.3) correspond to a dry tarmac runway.

Symbol	Parameter	Value
m	Aircraft mass	45420 Kg
I_{xx}, I_{yy}	Moments of inertia	$2.262e^6, 3.172e^6$ Kgm^2
I_{zz}, I_{xz}	Moments of inertia	$3.337e^6, -1.5e^3$ Kgm^2
S_{ref}	Planform area	122.4 m^2
$\bar{b}$	Wing span	34.10 m
$\bar{c}$	Chord length	4.194 m
K_{L_W}, K_{R_W}	Left,Right oleo stiffness	$2e5$ Nm^{-1}
K_{N_W}	Nose oleo stiffness	$4e4$ Nm^{-1}
C_{L_W}, C_{R_W}	Left,Right oleo damping coefficients	$1.5e^5$ Nsm^{-1}
C_{N_W}	Nose oleo damping coefficient	$5e4$ Nm^{-1}
l_{x_N}	Landing gear offsets from CG (Fig A.1)	10 m
l_{x_L}, l_{x_R}	Landing gear offsets from CG (Fig A.1)	2.932 m
l_{y_L}, l_{y_R}	Landing gear offsets from CG (Fig A.1)	$-3.795, 3.795$ m
$\bar{B}, \bar{C}, \bar{D}$	Magic formula parameters	$10, 1.9, 1$
F_T	Maximum thrust available (Twin engines)	150 kN
$\mu_r, \mu_b, m_w, R_{wheel}, c_b$	Wheel parameters	$2e^{-3}, 0.2, 87kg, 0.6m, 8e^3$
θ^*, h^*	Tail strike pitch attitude, altitude	$11°, 0.5m$
θ_1, θ_2	Pitch reference commands	$10°, 8°$
$C_{L_{max}}, C_{L_g}, C_{D_g}$	Lift and drag coefficients	$1.60, 0.46, 0.02$
$C_{m_q}, C_{m_{\delta_e}}$	Pitching moment coefficients	$-44.43, -1.785$
$C_{n_r}, C_{n_{\delta_r}}$	Yawing moment coefficients	$-0.405, -0.129$

Table A.1: Numerical paramters

APPENDIX B

Controllers

B.1 Envelope-Aware Control Law

In this work, the translational, rotational and directional dynamics are controlled independently by PD control laws. The pitch attitude control law is given by Eqn B.1:

$$\delta_e = K_{p_e}(\theta_{ref} - \theta) + K_{d_e}\dot{\theta} \tag{B.1}$$

$$\theta_{ref} = \begin{cases} 0 \;\; if \;\; (V < V_r) \\ \theta_1 \; if \;\; (V \geq V_r) \\ \theta_2 \; if \;\; (V \geq V_r) \wedge (\theta \leq \theta^*) \wedge (h < h^*) \end{cases} \tag{B.2}$$

$\theta_{ref} = 0$ enables the aircraft to accelerate with zero pitch until the rotation airspeed V_r is reached. A positive pitch attitude θ_1 enables the airplane to rotate and climb when the rotation speed is reached. $\theta_2 < \theta^*$ inhibits over-rotation thus reducing the risk due to a tail-strike.

The thrust input $\delta_T = T_{max}$ is used when continuing takeoff is safe whereas $\delta_T = 0$ is used to reject the takeoff when continuing takeoff is no longer safe. A discussion of safe versus unsafe states for different engine operating conditions (i.e. All engines operational/one engine inoperational) is provided in the next section.

The roll attitude control law is chosen to hold a zero bank angle $\phi_{ref} = 0$ during the takeoff:

$$\delta_a = K_{p_a}(\phi_{ref} - \dot{\phi}) + K_{d_a}\dot{\phi} \tag{B.3}$$

The directional control law is chosen as shown in Eqn B.5 enabling the aircraft to track the

runway center-line.

$$\delta_r = K_{p_r}(\psi_{ref} - \dot{\psi}) + K_{d_r}\dot{\psi} \tag{B.4}$$

$$\psi_{ref} = K_{p_y} y \tag{B.5}$$

The brake inputs are modeled as follows:

$$\delta_{B_l} = \begin{cases} \text{sat}(K_b y)\ if\ y > 0 \\ 0\ if\ \ y = 0 \end{cases} \tag{B.6}$$

$$\delta_{B_r} = \begin{cases} \text{sat}(K_b y)\ if\ y < 0 \\ 0\ if\ \ y = 0 \end{cases} \tag{B.7}$$

where *sat* denotes the saturation function. The use of differential braking (Eqn B.6-B.7) enables the controller to counteract cross wind forces especially while the rudder is ineffective at low airspeeds. Fig B.1 illustrates the aircraft's lateral response (cross track error) to a constant crosswind of 16 knots under different scenarios. From Fig B.1, it can be seen that the aircraft veers off the runway due to the crosswind when no control input is applied. With braking input alone (right brake input of 800 Nm), the aircraft can maintain runway centerline at low airspeeds. At higher airspeeds, braking does not provide sufficient cornering forces to counteract the crosswind. The rudder input alone is ineffective at low airspeeds, however as the airspeed increases, the rudder's effectiveness increases to provide sufficient yawing moment to return to the runway centerline. When rudder and braking are both used to maintain the runway centerline, the aircraft can be controlled with very small deviations from centerline.

B.2 Pilot Model

The pilot's elevator δ_e and rudder δ_r inputs are modeled as shown below, while 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 [11]).

$$\begin{aligned} \delta_e &= \begin{cases} k_{p_e}(\theta_{ref_1} - \theta(t-\tau)) + k_{d_e}\dot{\theta} & if\ (V \geq V_r) \\ k_{p_e}(\theta_{ref_2} - \theta(t-\tau)) + k_{d_e}\dot{\theta} & if\ (V < V_r) \end{cases} \\ \delta_r &= k_{p_r}(\psi_{ref} - \psi(t-\tau)) + k_{d_r}\dot{\psi} \\ \psi_{ref} &= k_Y y \end{aligned} \tag{B.8}$$

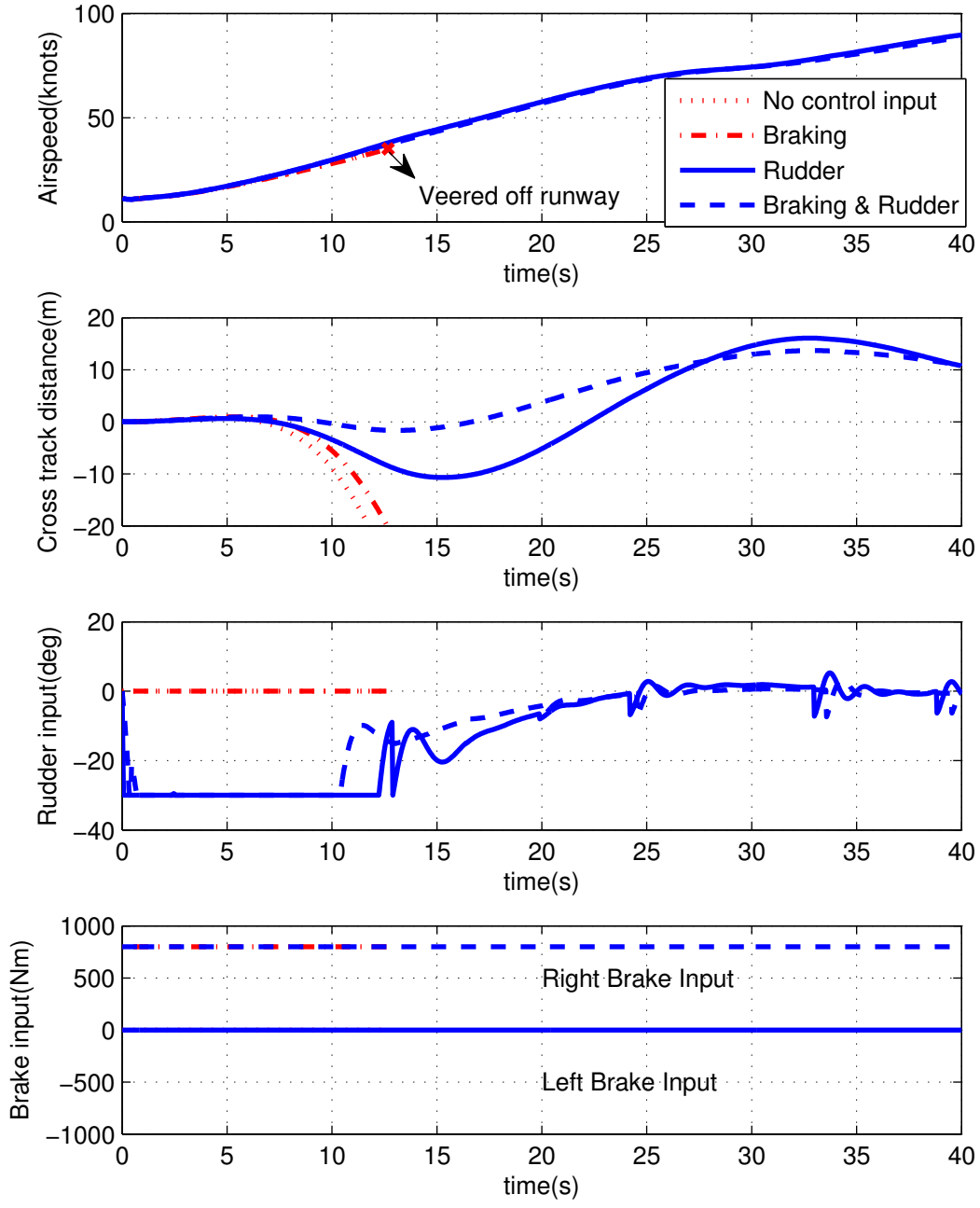

Figure B.1: Comparison of aircraft lateral response with braking and rudder control inputs

Eqn (B.8) represents a simple human operator model [74, 139] 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 center-line. 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 neuro-muscular interactions [74]. θ_{ref_1} is the appropriate pitch reference attitude during rotation. θ_{ref_2} is the reference pitch attitude before rotation (ideally zero). $\theta_{ref} - \theta(t-\tau)$ is the error in tracking the appropriate rotation attitude (θ_{ref}). V_r denotes the rotation airspeed perceived by the pilot, ideally V_R. The rudder control law enables center-line tracking.

Eqn (B.8) 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 [74]. 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 $[\theta_{ref_{min}}, \theta_{ref_{max}}]$, $[V_{r_{min}}, V_{r_{max}}]$, $[k_{p_{min}}, K_{p_{max}}]$, $[K_{d_{min}}, K_{d_{max}}]$ and $[\tau_{min}, \tau_{max}]$ respectively. The numerical values for these parameters are listed in Table A.1.

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 in-operative E_{OEI} or all engines can be inoperative E_{AEI}. For each takeoff sequence, the operational state of the engine $E \in \{E_{AEO}, E_{OEI}, E_{AEI}\}$ 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), 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 $t_{fail} \in [0, t_f]$ 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.

For a nominal takeoff sequence where all engines are operations E_{AEO}, the pilot sets the takeoff thrust. However, when an anomaly such as an engine failure occurs, we assume that the pilot executes either the appropriate actions (safely reject or continue takeoff) or inappropriate actions according to a specified distribution. A discussion of safe and unsafe behaviors in any given region is discussed in Section 2.6 of Chapter 2 and in Appendix C

APPENDIX C

Takeoff Envelopes

C.1 Translational Envelopes

To simplify the analysis of the translational takeoff envelopes, the non-linear aircraft equations of motion described above are simplified as described in [41]. Let (x, y) represent the longitudinal and lateral runway directions respectively, with $(0, 0)$ the ground roll initiation point on the runway centerline. Let V represent the airspeed, V_w represent the wind speed and V_{lof} represent the lift off airspeed. The simplified equations that describe takeoff and rejected takeoff dynamics are:

$$\dot{x} = (V \pm V_w)cos(\gamma) \tag{C.1}$$

$$\dot{V} = \begin{cases} A_1 - B_1 V^2 & V < V_{lof} \\ A_2 - B_2 V^2 & V \geq V_{lof} \end{cases} \tag{C.2}$$

where A_1, A_2, B_1, B_2 are defined as

$$A_1 = g(\frac{T}{W} - \mu) \tag{C.3}$$

$$B_1 = \frac{g}{W}(\frac{1}{2}\rho S_{ref}(C_{D_g} - \mu C_{Lg})) \tag{C.4}$$

$$A_2 = g(\frac{T}{W} - sin(\gamma)) \tag{C.5}$$

$$B_2 = \frac{g}{W}(\frac{1}{2}\rho S_{ref} C_{D_g}) \tag{C.6}$$

Here T represents the takeoff thrust with all engines operational, W represents the aircraft's takeoff weight, ρ represents the atmospheric density, μ represents the rolling friction coefficient. μ_b is the braking friction coefficient for RTO and, γ is the flight path angle. We assume $\gamma = 0$ when $V \leq V_{lof}$ and $\gamma = \gamma_0$ ($\gamma_0 > 0$) when $V > V_{lof}$. α represents the angle of attack, S_{ref} is the planform area, and C_{L_g} and C_{D_g} are the coefficients of lift and drag,

respectively, including ground effects and nominal flaps/slat settings for takeoff. Note that Eqn C.1 accounts for the wind speed as well and hence the ground speed is given as $V - V_w$ for a head wind and $V + V_w$ for a tail wind.

For a continued takeoff with one-engine in-operational, the airspeed dynamics is given as

$$\dot{V} = \begin{cases} A_3 - B_3 V^2 & if\ V < V_{lof} \\ A_4 - B_4 V^2 & if\ V \geq V_{lof} \end{cases} \tag{C.7}$$

where A_3, B_3, A_4, B_4 are defined as

$$A_3 = g(\frac{\eta T_{max}}{W} - \mu) \tag{C.8}$$

$$B_3 = \frac{g}{W}(\frac{1}{2}\rho S_{ref}(C_{D_g} - \mu C_{Lg})) \tag{C.9}$$

$$A_4 = g(\frac{\eta T_{max}}{W} - sin(\gamma)) \tag{C.10}$$

$$B_4 = \frac{g}{W}(\frac{1}{2}\rho S_{ref} C_{D_g}) \tag{C.11}$$

T_{max} represents the maximum available thrust output from all the engines. η represents the thrust reduction due to a single engine failure ($\eta = 0.5$ for twin engine aircraft and $\eta = 0.75$ for aircraft with four engines). For rejected takeoff, the airspeed dynamics is given by:

$$\dot{V} = A_5 - B_5 V^2 \ \ if\ V > 0 \tag{C.12}$$

where A_5, B_5 are defined as

$$A_5 = g(-\mu_b) \tag{C.13}$$

$$B_5 = \frac{g}{W}(\frac{1}{2}\rho S_{ref}(C_{D_g} - \mu_r C_{Lg})) \tag{C.14}$$

From Eqn C.1-C.2, the distance traveled S from $V = V_a$ to $V = V_b$ is obtained as

$$S(V_a, V_b)) = \int_{V_a}^{V_b} \frac{(V \pm V_w)cos(\gamma)}{A_i - B_i V^2} dV \tag{C.15}$$

where A_i, B_i are chosen appropriately for the given takeoff sequence as defined in Eqn C.2-C.14.

Unsafe states with respect to $V - x$ are derived by imposing the following constraints on the translational dynamics: For continued takeoff: (1) The aircraft must be airborne

on a limited runway length R_{max} with the available thrust $T \leq T_{max}$, (2) The aircraft must clear an obstacle height h_{obs} at an airspeed not less than V_2 [127]. (3) In case of a rejected takeoff, the aircraft must be able to stop within the available runway. The above constraints lead to the following non-linear program that can be used to find the initial conditions that give rise to trajectories that partition the state-space with respect to safe versus unsafe states as shown in figures 2.6-2.8.

$$\min \ T \tag{C.16}$$

subject to the following constraints:

$$S_1 = R_{max} \tag{C.17}$$
$$S_2 + S_3 = R_{max} \tag{C.18}$$
$$S_4 = S_5 \tag{C.19}$$
$$S_6 = S_5 \tag{C.20}$$
$$S_3 \tan(\gamma) \geq h_{obs} \tag{C.21}$$
$$V_1 \leq V_{lof} \tag{C.22}$$
$$T < T_{max} \tag{C.23}$$

$S_i, i = 1 \ldots 6$ are defined as

$$S_1 = S(V_{ab0}, \mp V_w) \qquad S_4 = S(V_{ef0}, V_1) \tag{C.24}$$
$$S_2 = S(V_{ef0}, V_{lof}) \qquad S_5 = S(V_{ab0}, V_1) \tag{C.25}$$
$$S_3 = S(V_{lof}, V_2) \qquad S_6 = S(\mp V_w, V_1) \tag{C.26}$$

where V_2 represents the required airspeed when clearing the obstacle height h_{obs} at R_{max} and is assumed given. V_{ef0} represents the minimum airspeed required to safely continue takeoff. V_{ab0} represents the maximum airspeed that the aircraft should posses to reject a takeoff safely. The above non-linear program is solved to find V_1, V_{ef0}, V_{ab0}, T which lead to the takeoff envelopes shown in Fig C.1.

C.1.1 Rotational envelopes

For rotational dynamics, the constraints $\theta < \theta^*, h < h^*$ prevent tail-strike scenario which can be encountered during the initial rotation to become airborne. Thus, the set of all initial conditions that lead to a tail-strike scenario must be avoided to eliminate any loss of control

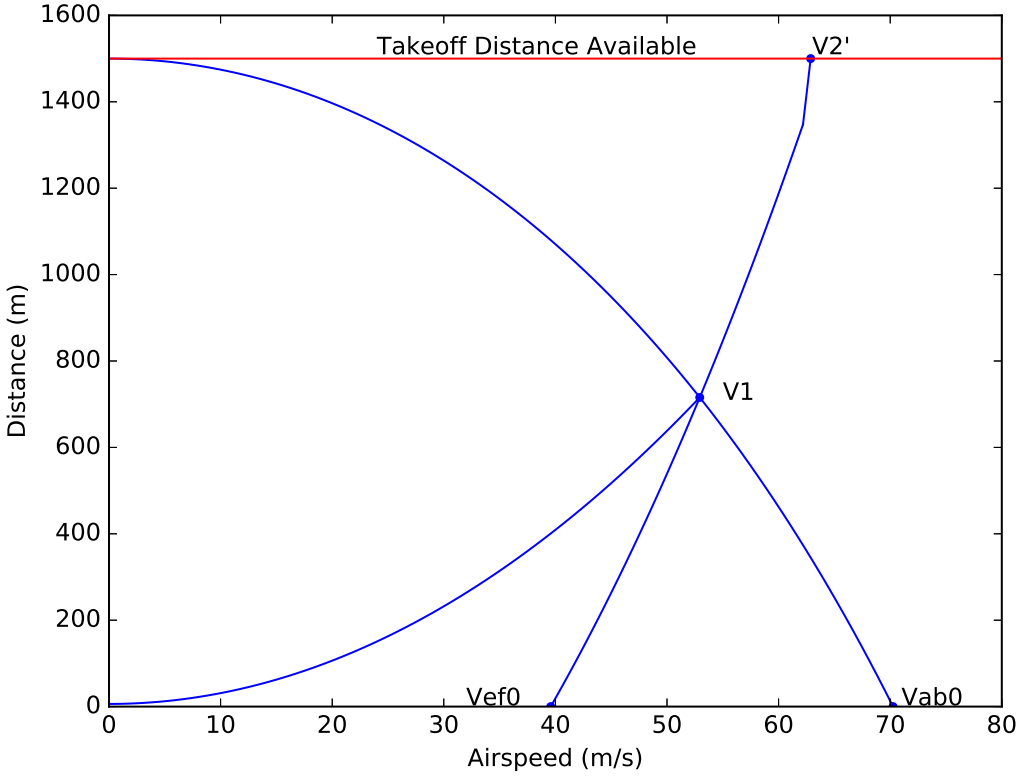

Figure C.1: Translational envelopes

risk due to a tail-strike. To estimate this set, we first we simplify Eqn A.18 to study the rotational dynamics when $h < h^*$ as follows.

$$\dot{\theta} = q \tag{C.27}$$

$$\dot{q} = \frac{1}{2I_{yy}} \rho V_*^2 S_{ref} \bar{c} (C_{m_q} q + C_{m_{\delta e}} \delta_e) \tag{C.28}$$

Where $C_{m_q}, C_{m_{\delta e}}$ represent the pitch moment coefficients. $\delta_e \in [-25°, 25]$ is chosen as defined in Eqn B.1. V_* is the airpseed at which the unsafe set is to be computed. The set of all initial conditions for Eqn C.27-C.28 that can result in state trajectories that lead to tail trike scenarios is estimated using the level set tool box [140].

C.1.2 Directional envelopes

For directional dynamics, the constraint $|y| < y_0$ prevents the aircraft from veering off the side of the runway. The directional dynamics are simplified as follows:

$$\dot{\psi} = r \tag{C.29}$$

$$\dot{r} = \frac{1}{2I_{zz}} \rho V_*^2 S_{ref} \bar{b} (C_{n_r} r + C_{n_{\delta r}} \delta_r) \tag{C.30}$$

$$\dot{y} = V_* sin(\psi) \tag{C.31}$$

Where $C_{n_r}, C_{n_{\delta r}}$ represent the yawing moment coefficients. $\delta_r \in [-25°, 25]$ is chosen as defined in Eqn B.5. V_* is the airpseed at which the unsafe set is to be computed. Using Eqn C.29-C.31, the set of all initial conditions that lead to constraint violation are estimated using the level set tool box [140].

APPENDIX D

MDP Formulation to Prevent In-Flight Icing Related Loss of Control

D.1 State Transition Probabilities

Consider the flight segment from **SUSKE** and **BUFST**. As indicated in Fig 4.5 and Fig 4.7, this segment requires the flight crew to maintain 5500 ft. Ideally, to express state transition probabilities in this scenario, one would require probability distributions conditioned on flight plan features of all speed ranges. However, for ease of illustration, this work only considers distributions conditioned on $\bar{f}_2$ (i.e. a level-flight condition at medium airspeed). Furthermore, this work only considers cases where the aircraft flies straight and level at a cruise power setting while maintaining angle of attack, side-slip, pitch and roll attitude within the safe operating envelopes i.e. $\bar{A} = \bar{\alpha}_1, \bar{\Theta} = \bar{\theta}_1, \bar{\Phi} = \bar{\phi}_1, \bar{H} = \bar{h}_1, \bar{T} = \bar{t}_3$. Conditioned on these state features, the state transition probabilities for the airspeed states under pilot authority ($M = P$) and Envelope-Aware automation authority ($M = EA$) are indicated in Tables D.1-D.3. Table D.1 lists the airspeed transition probabilities when the aircraft is free from icing conditions (i.e. $\bar{I} \in \{i_0, i_1, i_4\}$). Table D.2 lists transition probabilities for $\bar{I} = \bar{i_2}$ and Table D.3 lists the transition probabilities for $\bar{I} = \bar{i}_3$. Note that with the pilot in control $M = P$, the probability of entering a high-risk airspeed state $\bar{V} = v_1$ and stalling increases with adverse icing conditions. However, under envelope-aware control authority $M = EA$, the probability of entering a stall state is low. Better performance when $M = EA$ is attributed to the combined EA-FMS capabilities to identify changes in dynamics, adapt controllers to these changes, estimate degraded flight envelopes and construct and follow flight plans that respect degraded envelope constraints.

Consider the flight segment from **ZADUM** to **BIILS** where the aircraft is in a descending right turn. This case considers only distributions conditioned on $\bar{f}_{17}$ (i.e. a descending turn at medium airspeed) and assumes the following states remain constant: $\bar{A} = \bar{\alpha}_1, \bar{\Theta} = \bar{\theta}_1, \bar{H} = \bar{h}_4, \bar{T} = \bar{t}_2$. For this segment, both bank angle and airspeed state are

assumed likely to change. State transition probabilities for bank angle transitions under pilot $M = P$ and Envelope-Aware control $M = EA$ are indicated in Table D.4 while Table D.5 indicate distributions for airspeed states when $\bar{\Phi} = \bar{\phi}_3$. These distributions assume that $\bar{I} = \bar{i}_0$. Note that the probability of stalling increases when the bank angle is steep $\bar{\Phi} = \bar{\phi}_3$. This is attributed to the fact that during steep turns the stall speed increases [70].

D.2 Optimal Values

The optimal values are generated using Value Iteration [62] with the reward formulation, weighting factors and transition probabilities discussed in Chapter 4. A discount factor $\lambda = 0.7$ was used. These optimal values are used in the computation of the optimal policies illustrated in Section 4.4.

Table D.1: Distribution $\mathcal{P}_1(\bar{v}_j \mid \bar{v}_i, \bar{\alpha}_1, \bar{\theta}_1, \bar{\phi}_{1,2}, \bar{h}_1, \bar{t}_3, \bar{i}_{0,1,4}, \bar{f}_2, P_s, M)$. Left $M = P$, Right $M = EA$

	$\bar{v}_1$	$\bar{v}_2$	$\bar{v}_3$	$\bar{v}_4$
$\bar{v}_1$	0.6	0.4	0.0	0.0
$\bar{v}_2$	0.4	0.1	0.5	0.0
$\bar{v}_3$	0.0	0.1	0.5	0.4
$\bar{v}_4$	0.0	0.0	0.5	0.5

	$\bar{v}_1$	$\bar{v}_2$	$\bar{v}_3$	$\bar{v}_4$
$\bar{v}_1$	0.0	1.0	0.0	0.0
$\bar{v}_2$	0.0	0.3	0.7	0.0
$\bar{v}_3$	0.0	0.1	0.5	0.4
$\bar{v}_4$	0.0	0.0	0.5	0.5

Table D.2: Distribution $\mathcal{P}_1(\bar{v}_j \mid \bar{v}_i, \bar{\alpha}_1, \bar{\theta}_1, \bar{\phi}_{1,2}, \bar{h}_1, \bar{t}_3, \bar{i}_2, \bar{f}_2, P_s, M)$. Left $M = P$, Right $M = EA$

	$\bar{v}_1$	$\bar{v}_2$	$\bar{v}_3$	$\bar{v}_4$
$\bar{v}_1$	0.7	0.3	0.0	0.0
$\bar{v}_2$	0.5	0.2	0.3	0.0
$\bar{v}_3$	0.0	0.2	0.5	0.3
$\bar{v}_4$	0.0	0.0	0.5	0.5

	$\bar{v}_1$	$\bar{v}_2$	$\bar{v}_3$	$\bar{v}_4$
$\bar{v}_1$	0.1	0.9	0.0	0.0
$\bar{v}_2$	0.0	0.3	0.7	0.0
$\bar{v}_3$	0.0	0.1	0.5	0.4
$\bar{v}_4$	0.0	0.0	0.5	0.5

Table D.3: Distribution $\mathcal{P}_1(\bar{v}_j \mid \bar{v}_i, \bar{\alpha}_1, \bar{\theta}_1, \bar{\phi}_{1,2}, \bar{h}_1, \bar{t}_3, \bar{i}_3, \bar{f}_2, P_s, M)$. Left $M = P$, Right $M = EA$

	$\bar{v}_1$	$\bar{v}_2$	$\bar{v}_3$	$\bar{v}_4$
$\bar{v}_1$	0.98	0.02	0.00	0.00
$\bar{v}_2$	0.85	0.15	0.00	0.00
$\bar{v}_3$	0.00	0.90	0.10	0.00
$\bar{v}_4$	0.00	0.00	0.90	0.10

	$\bar{v}_1$	$\bar{v}_2$	$\bar{v}_3$	$\bar{v}_4$
$\bar{v}_1$	0.4	0.6	0.0	0.0
$\bar{v}_2$	0.2	0.5	0.3	0.0
$\bar{v}_3$	0.0	0.8	0.2	0.0
$\bar{v}_4$	0.0	0.0	0.6	0.4

Table D.4: Distribution $\mathcal{P}_4(\bar{\phi}_j \mid \bar{\phi}_i, \bar{v}_{1,2,3,4}, \bar{\alpha}_1, \bar{\theta}_1, \bar{h}_4, \bar{t}_2, \bar{i}_4, \bar{f}_{17}, P_s, M)$. Left $M = P$, Right $M = EA$

	$\bar{\phi}_1$	$\bar{\phi}_2$	$\bar{\phi}_3$
$\bar{\phi}_1$	0.80	0.20	0.00
$\bar{\phi}_2$	0.20	0.30	0.50
$\bar{\phi}_3$	0.00	0.01	0.99

	$\bar{\phi}_1$	$\bar{\phi}_2$	$\bar{\phi}_3$
$\bar{\phi}_1$	1.0	0.0	0.0
$\bar{\phi}_2$	1.0	0.0	0.0
$\bar{\phi}_3$	0.0	1.0	0.0

Table D.5: Distribution $\mathcal{P}_1(\bar{v}_j \mid \bar{v}_i, \bar{\alpha}_1, \bar{\theta}_1, \bar{\phi}_3, \bar{h}_4, \bar{t}_2, \bar{i}_4, \bar{f}_{17}, P_s, M)$. Left $M = P$, Right $M = EA$

	$\bar{v}_1$	$\bar{v}_2$	$\bar{v}_3$	$\bar{v}_4$
$\bar{v}_1$	0.9	0.1	0.0	0.0
$\bar{v}_2$	0.7	0.2	0.1	0.0
$\bar{v}_3$	0.0	0.8	0.2	0.0
$\bar{v}_4$	0.0	0.0	1.0	0.0

	$\bar{v}_1$	$\bar{v}_2$	$\bar{v}_3$	$\bar{v}_4$
$\bar{v}_1$	0.0	1.0	0.0	0.0
$\bar{v}_2$	0.0	0.6	0.4	0.0
$\bar{v}_3$	0.0	0.6	0.4	0.0
$\bar{v}_4$	0.0	0.0	1.0	0.0

Table D.6: Distributions $\mathcal{P}_7$ and $\mathcal{P}_8$ for $M \in \{P, EA\}$

	$\bar{f}_2$	$\bar{f}_{17}$
$\bar{f}_2$	0.5	0.5
$\bar{f}_{17}$	0.5	0.5

	$\bar{i}_2$	$\bar{i}_3$	$\bar{i}_4$
$\bar{i}_2$	1.0	0.0	0.0
$\bar{i}_3$	0.0	1.0	0.0
$\bar{i}_4$	0.0	0.0	1.0

APPENDIX E

Reachability algorithm for FSAM verification

E.1 Reachability Analysis

Algorithm E.1 describes the *isReach()* function used by Algorithm 6.1. It takes as inputs two facets $\overset{\circ}{q}$ and $\overset{\circ}{q'}$ of a cell and returns true if under the current control authority p_k there exists a trajectory starting from $\overset{\circ}{q}$ and ending in $\overset{\circ}{q'}$ while remaining within the cell containing the two facets. The main idea in Algorithm E.1 is to exploit the fact that the airspeed and altitude (in Eqn 6.1) monotonically increase with time (in the region of interest) and the pitch dynamics is piece-wise affine, therefore it is enough to propagate the extreme points of the facet. Algorithm E.1 propagates the initial condition $\bar{X}_0 = [x_0, v_0, h_0, \theta_0, q_0]$ obtained from each vertex of facet $\overset{\circ}{q}$, until the ensuing trajectory leaves the cell, to determine if facet $\overset{\circ}{q'}$ is reachable from $\overset{\circ}{q}$. [1] Since this work considers discrete-time semantics of LTL, state propagation is performed using the discrete-time version of the system dynamics in Eqn 6.1. $\bar{f}$ in Algorithm E.1 denotes the discrete-time equivalent of f. Each vertex of facet $\overset{\circ}{q}$ provides initial conditions for airspeed (v_0), pitch (θ_0) and altitude (h_0). The longitudinal position (x_0) is initialized at zero since 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. $\mathcal{V}(\overset{\circ}{q}) \subset \mathbb{R}^5$ denotes the set of initial conditions for a facet $\overset{\circ}{q}$. Note that since the pilot model described by Eqn B.8 consists of the parameters $\theta_{ref}, V_r, K_p, K_d, \tau$ whose 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, $K_p \in \{K_{p_{min}}, K_{p_{max}}\}, K_d \in \{K_{d_{min}}, K_{d_{max}}\}, \theta_{ref} \in \{\theta_{ref_{min}}, \theta_{ref_{max}}\}, V_r \in \{V_{r_{min}}, V_{r_{max}}\}$ and

[1] 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, where δ can be inferred from the sampling time and the Lipschitz constant of the dynamics [125].

$\tau \in \{\tau_{min}, \tau_{max}\}$. For $p_k = EA$, 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 Appendix B.2 (when $p_k = P$) or Appendix B.1 (when $p_k = EA$) using the parameters $\eta = (\theta_{ref}, V_r, K_p, K_d, \tau)$. Note that Algorithm E.1 requires the initialization of the delay term in the pilot control mode described by Eqn B.8. 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 E.1 *isReach()* function

function *isReach*$(\mathring{q}, p_k, \mathring{q}')$

1. **for** $\bar{X}_0$ **in** $\mathcal{V}(\mathring{q})$
2. $X = \bar{X}_0$
3. **for** η **in** $\mathcal{K}(p_k, X)$
4. $\bar{q}_i := \bar{q}_j := \mathcal{F}^{-1}(\mathring{q})$
5. **while** $\bar{q}_j = \bar{q}_i$
6. $X' := \bar{f}(X, U_\eta)$
7. $\bar{q}_j := \bar{T}(X')$
8. $X' := X$
9. **if** $\bar{q}_j \neq \bar{q}_i$
10. **if** $\mathring{q}' \in \mathcal{F}(\bar{q}_i) \cap \mathcal{F}(\bar{q}_j)$
11. **return** *TRUE*
12. **return** *FALSE*

Table E.1: Numerical parameters

Parameters	**Values**
m, S_{ref}, I_{yy}	45420 kg, 122.4 m^2, 0.3172e7 kgm^2
$\rho, \bar{c}$	1.225 kgm^{-3} ,4.19m
$\alpha_0, \gamma_0, T_{max}$	$0°, 8°, 300kN$
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	$(C_{m_q}, C_{m_{ue}}) \times \frac{1}{2I_{yy}} \rho V_{mcg}^2 S_{ref} \bar{c}$
A_4, B_4	$(C_{m_q}, C_{m_{ue}}) \times \frac{1}{2I_{yy}} \rho V_R^2 S_{ref} \bar{c}$
$\bar{K}_1, \bar{K}_3, \bar{K}_5, \bar{K}_7$	-3
$\bar{K}_2, \bar{K}_4, \bar{K}_6, \bar{K}_8$	0.1
$\theta_{PR}, \theta_{TS}, \theta_{ng}, \theta_{tail}, h_{TS}, h_{lof}$	$3°, 9°, 3°, 10°, 0.9m, 2.5m$
$(\theta_{ref_{min}}, \theta_{ref_{max}}), (V_{r_{min}}, V_{r_{max}})$	$(7°, 13°), (50, 70) ms^{-1}$
$(K_{p_{min}}, K_{p_{max}}), (\tau_{min}, \tau_{max})$	$(-5, -1), (0, 0.1) s$
$V_{mcg}, V_1, V_R, V_{lof}, V_2, V_{fp}$	$10, 47, 55, 66, 67.5, 80$ (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)
Number of Monte Carlo trials	1000

BIBLIOGRAPHY

[1] "Statistical Summary of Commercial Jet Airplane Accidents," Boeing technical issue, 2015, accessed September 30, 2015, http://www.boeing.com/news/techissues/pdf/statsum.pdf.

[2] Flight Safety Foundation, "Reducing the Risk of Runway Excursions, Report of the Runway Safety Initiative," May 2009, Accessed September 2012, http://www.skybrary.aero/bookshelf/books/900.pdf.

[3] Airspeed Indicator, Image source:https://commons.wikimedia.org/wiki/File:Airspeed_Indicator.svg[accessed Dec'15].

[4] Vertical Speed Indicator, Image source:https://commons.wikimedia.org/wiki/File:Vertical_Speed_Indicator.svg[accessed Dec'15].

[5] Jason Tatum, RPM gauge, Image Source: http://siminnovations.com/online_store/instruments/preview/32f9f183-506a-485d-b89b-716942245354.png[accessed Dec'15].

[6] Jacklin, S. A. and Lowry, M. R. and Schumann, J. M. and Gupta, P. and Bosworth, J. T. and Zavala, E. and Kelly, J. and Hayhurst, K. J. and 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*, Providence, RI, 2004.

[7] Belcastro, C. M. and Foster, J. V., "Aircraft Loss of Control Accident Analysis," *Proc. AIAA Guidance Navigation, and Control Conference*, No. AIAA 2010-8004, Toronto, Ontario, 2010, doi: 10.2514/6.2010-8004.

[8] 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," *Proc. AIAA Guidance Navigation, and Control Conference*, No. AIAA 2014-0612, National Harbor, MD, 2014, doi: 10.2514/6.2014-0612.

[9] Belcastro, C. M. and Jacobson, S. R., "Future Integrated System Concepts for Preventing Aircraft Loss-of-Control Accidents," *Proc. AIAA Guidance Navigation, and Control Conference*, No. AIAA 2010-8142, Toronto, Ontario, 2010, doi: 10.2514/6.2010-8142.

[10] Traverse, P., *The Avionics Handbook*, chap. 31, CRC press LLC, 2015, Airbus Electrical Flight Controls.

[11] Gregg, F. B., *The Avionics Handbook*, chap. 29, CRC press LLC, 2015, Boeing B-777: Fly-By-Wire Flight Controls.

[12] Breen, B., "Controlled Flight Into Terrain and the enhanced Ground Proximity Warning system," *Aerospace and Electronic Systems Magazine, IEEE*, Vol. 14, No. 1, 1999, pp. 19–24, doi: 10.1109/62.738350.

[13] Armand, J and Lignee, R and Villaume, F, "The Runway Overrun Prevention System," July 2009, Safety First-The Airbus Safety Magazine, Issue 8, Publisher: Airbus S.A.S.

[14] "Descent Below Visual Glidepath and Impact With Seawall, Asiana Airlines Flight 214, Boeing 777-200ER, HL7742, San Fransisco, California, July 6, 2013," Accident report NTSB/AAR-14/01, 2014.

[15] Chongvisal, N. T., Xargay, E., Talleur, D., Kirlik, A., and Hovakimyan, N., "Loss-of-control prediction and prevention for NASAs Transport Class Model," *Proceedings of AIAA Guidance, Navigation and Control Conference*, National Harbor, MD, 2014, doi: 10.2514/6.2014-0784.

[16] Ackerman, A., Pelech, S. T., Carbonari, R. S., Hovakimyan, N., Kirlik, A., and Gregory, I. M., "Pilot-in-the-loop flight simulator for NASAs Transport Class Model," *Proceedings of AIAA Guidance, Navigation and Control Conference*, National Harbor, MD, 2014, doi: 10.2514/6.2014-0613.

[17] Tekles, E. X., Choe, R., Hovakimyan, N., Gregory, I., and Holzapfel, F., "Flight envelope protection for NASAs Transport Class Model," *Proceedings of AIAA Guidance, Navigation and Control Conference*, National Harbor, MD, 2014, doi: 10.2514/6.2014-0269.

[18] Bak, S., Manamcheri, K., Mitra, S., and Caccamo, M., "Sandboxing controllers for cyber-physical systems," *Cyber-Physical Systems (ICCPS), 2011 IEEE/ACM International Conference on*, IEEE, 2011, pp. 3–12.

[19] 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.

[20] 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*, Orlando, FL, 2009, doi: 10.2514/6.2009-1458.

[21] Bragg, M. B., Basar, T., Perkins, W. R., Selig, M. S., Voulgaris, P. G., Melody, J. W., and Sarter, N. B., "Smart icing systems for aircraft icing safety," , No. AIAA-2002-0813, 2002, doi: 10.2514/6.2002-813.

[22] Idan, M., Johnson, M., Calise, A., and Kaneshige, J., "Intelligent aerodynamic/propulsion flight control for flight safety: a nonlinear adaptive approach," *Proceedings of the American Control Conference, 2001.*, Vol. 4, 2001, pp. 2918–2923 vol.4, doi: 10.1109/ACC.2001.946346.

[23] Rysdyk, R. T. and Calise, A. J., "Fault tolerant flight control via adaptive neural network augmentation," 1998, doi: 10.2514/6.1998-4483.

[24] Napolitano, M. R., An, Y., and Seanor, B. A., "A fault tolerant flight control system for sensor and actuator failures using neural networks," *Aircraft Design*, Vol. 3, No. 2, 2000, pp. 103 – 128, doi: 10.1016/S1369-8869(00)00009-4.

[25] Wensley, J., Lamport, L., Goldberg, J., Green, M., Levitt, K., Melliar-Smith, P., Shostak, R., and Weinstock, C., "SIFT: Design and analysis of a fault-tolerant computer for aircraft control," *Proceedings of the IEEE*, Vol. 66, No. 10, 1978, pp. 1240–1255, doi: 10.1109/PROC.1978.11114.

[26] Burns, A., Harper, D., Barfield, A., Whitcomb, S., and Jurusik, B., "Auto GCAS for analog flight control system," *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th*, 2011, pp. 8C5–1–8C5–11, doi: 10.1109/DASC.2011.6096148.

[27] Alur, R., "The Theory of Timed Automata," *Theoretical Computer Science*, Vol. 126, 1999, pp. 183–225.

[28] Hopcroft, J. E., *Introduction to automata theory, languages, and computation*, chap. 2, Pearson Education India, 1979, pp. 38–83.

[29] Savage, J. E., *Models of computation, Exploring the Power of Computing*, chap. 4, Addison-Wesley, 1998, pp. 153–207.

[30] Moore, E. F., *Automata studies*, chap. Gedanken-experiments on sequential machines, Princeton University Press, 1956, pp. 129–153.

[31] Gigante, G. and Pascarella, D., "Formal methods in avionic software certification: the DO-178C perspective," *Leveraging Applications of Formal Methods, Verification and Validation. Applications and Case Studies*, Springer, 2012, pp. 205–215.

[32] "Northwest Airlines, INC; McDonnnell Douglas DC 9-82,N312RC, Detroit Metropolitan Wayne County Airport, Romulus, Michigan," Accident report NTSB/AAR-85/05, 1988, http://libraryonline.erau.edu/online-full-text/ntsb/aircraft-accident-reports/AAR88-05.pdf, Dec 2012.

[33] "Runway Side Excursion During Attempted Takeoff in Strong and Gusty Crosswind Conditions- Continental Airlines Flight 1404,Boeing 737-500, N18611," Accident report NTSB/AAR-10/04, 2008.

[34] Federal Aviation Authority, "Pilot Guide to Takeoff Safety," Online database, Accessed November 2012, http://www.faa.gov/other_visit/aviation_industry/airline_operators/training/media/takeoff_safety.pdf.

[35] 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.

[36] 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.

[37] 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.

[38] Verspay, J. and Khatwa, R., "A comparative evaluation of three take-off performance monitor display types," *Flight Simulation and Technologies*, Guidance, Navigation, and Control and Co-located Conferences, AIAA, Aug. 1993.

[39] Inagaki, T. and Itoh, M., "Situation-adaptive autonomy: the potential for improving takeoff safety," *Robot and Human Communication, 1997. RO-MAN '97. Proceedings., 6th IEEE International Workshop on*, Sep 1997, pp. 302–307.

[40] 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 Publications, 2000, pp. 13–16.

[41] Roskam, J. and Lan, C. T. E., *Airplane Aerodynamics and Performance*, chap. 10, DARcorporation, 1997, pp. 435–507.

[42] Tomlin, C. J., *Hybrid Control of Air Traffic Management Systems*, Ph.D. thesis, EECS Department, University of California, Berkeley, 1998.

[43] Balachandran, S. and Atkins, E. M., "Flight Safety Assessment and Management during Takeoff," *AIAA Infotech@Aerospace Conference*, No. AIAA 2013-4805, Boston, MA, 2013, doi: 10.2514/6.2013-4805.

[44] Baier, C. and Katoen, J. P., *Principles of model checking*, Vol. 26202649, chap. 2, MIT press Cambridge, 2008, pp. 19–82.

[45] 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*, No. AIAA 2014-0785, National Harbor, MD, 2014, doi: 10.2514/6.2014-0785.

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

[47] Kochenderfer, M. J., Chryssanthacopoulos, J. P., Kaelbling, L. P., and Lozano-Perez, T., "Model-Based Optimization of Airborne Collision Avoidance Logic," MIT, Lincoln Laboratory Project report, 2010, Accessed September 2015, http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA513420.

[48] Temizer, S., *Planning Under Uncertainty for Dynamic Collision Avoidance*, Ph.D. thesis, Massachusetts Institute of Technology, 2011, Accessed Septermber 2015, http://dspace.mit.edu/handle/1721.1/64487.

[49] Winder, L. F., *Hazard Avoidance Alerting with Markov Decision Processes*, Ph.D. thesis, Massachusetts Institute of Technology, 2004, Accessed February 2013, http://hdl.handle.net/1721.1/28860.

[50] 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.

[51] 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," *In Proc. American Control Conference*, 2014, doi: 10.1109/ACC.2014.6859428.

[52] 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.

[53] 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 & Theoretical Artificial Intelligence*, Vol. 9, No. 2-3, 1997, pp. 237–256, doi: 10.1080/095281397147103.

[54] 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.

[55] 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*, No. AIAA 2010-8142, 2009.

[56] 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," *Proc. AIAA Guidance Navigation, and Control Conference*, National Harbor, Maryland, 2014.

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

[58] Meuleau, N., Plaunt, C., Smith, D. E., and Smith, T. B., "An Emergency Landing Planner for Damaged Aircraft." *Proceedings of the 21st Innovative Applications of Artificial Intelligence Conference*, Pasadena, California, 2009.

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

[60] 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*, Kissimmee, FL, 2015, doi: 10.2514/6.2015-1798.

[61] Russell, S. J. and Norvig, P., *Artificial intelligence: A Modern Approach*, chap. 17, Pearson Education Limited, 2014.

[62] Puterman, M. L., *Markov Decision Process: Discrete Stochastic Dynamic Programming*, chap. 2-6, John Wiley & Sons, Inc, 1994.

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

[64] Abbeel, P., *Apprenticeship learning and reinforcement learning with application to robotic control*, Ph.D. thesis, Stanford University, 2008.

[65] Tsitsiklis, J. N. and Van Roy, B., "Feature-based methods for large scale dynamic programming," *Machine Learning*, Vol. 22, No. 1-3, 1996, pp. 59–94.

[66] Ng, A. Y. and Jordan, M., "PEGASUS: A policy search method for large MDPs and POMDPs," *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., 2000, pp. 406–415.

[67] Deisenroth, M. and Rasmussen, C. E., "PILCO: A model-based and data-efficient approach to policy search," *Proceedings of the 28th International Conference on machine learning (ICML-11)*, 2011, pp. 465–472.

[68] Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y., "An application of reinforcement learning to aerobatic helicopter flight," *Advances in neural information processing systems*, Vol. 19, 2007, pp. 1.

[69] Kearns, M., Mansour, Y., and Ng, A. Y., "A sparse sampling algorithm for near-optimal planning in large Markov decision processes," *Machine Learning*, Vol. 49, No. 2-3, 2002, pp. 193–208.

[70] Stevens, L. B. and Lewis, L. F., *Aircraft Control and Simulation*, chap. 1-2, Hoboken, NJ: Wiley, 2003, pp. 1–137.

[71] Busso, C., Deng, Z., Yildirim, S., Bulut, M., Lee, C. M., Kazemzadeh, A., Lee, S., Neumann, U., and Narayanan, S., "Analysis of Emotion Recognition Using Facial Expressions, Speech and Multimodal Information," *Proceedings of the 6th international conference on Multimodal interfaces*, ACM, 2004, pp. 205–211.

[72] 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.

[73] Altman, E., *Constrained Markov decision processes*, Vol. 7, CRC Press, 1999.

[74] McRuer, D. T. and Krendel, E. S., "Mathematical Models of Human Pilot Behavior," Published by Advisory Group for Aerospace Research and Development (North Atlantic Treaty Organisation), 1974, Accessed September 2015, https://www.cso.nato.int/Pubs/rdp.asp?RDP=AGARD-AG-188.

[75] "Tailstrike and runway overrun, Melbourne Airport, Victoria," Australian Transportation Safety Bureau accident Report AO-2009-012, 2009, Australian Capital Territory, Australia http://www.atsb.gov.au/publications/investigation_reports/2009/aair/ao-2009-012.aspx.

[76] Munos, R. and Moore, A. W., "Variable resolution discretization for high-accuracy solutions of optimal control problems," *Robotics Institute*, 1999, pp. 256.

[77] Munos, R. and Moore, A., "Influence and variance of a Markov chain: Application to adaptive discretization in optimal control," *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, Vol. 2, IEEE, 1999, pp. 1464–1469.

[78] Uther, W. T. and Veloso, M. M., "Tree based discretization for continuous state space reinforcement learning," *Aaai/iaai*, 1998, pp. 769–774.

[79] Filev, D. P. and Kolmanovsky, I., "Generalized markov models for real-time modeling of continuous systems," *Fuzzy Systems, IEEE Transactions on*, Vol. 22, No. 4, 2014, pp. 983–998.

[80] Sutton, R. S. and Barto, A. G., *Reinforcement Learning: An Introduction*, chap. 6-10, MIT press Cambridge, 1998.

[81] Sand, W. R., Cooper, W. A., Politovich, M. K., and Veal, D. L., "Icing conditions encountered by a research aircraft," *Journal of climate and applied meteorology*, Vol. 23, No. 10, 1984, pp. 1427–1440.

[82] Van Dyke, D., "Tailplane icing: survival knowledge for pilots," *Professional Pilot*, Vol. 43, No. 9, 2009.

[83] Bureau d'Enquetes et d'Analyses, "Final report on the accident on 1st June 2009 to the Airbus A330-203 registered F-GZCP operated by Air France flight AF 447 Rio de Janeiro–Paris," *Ministère de l'Écologie. du Dévéloppement durable, des Transports et du Logement, Paris*, 2012.

[84] Dutch Safety Board, "Crashed during approach, Boeing 737-800, near Amsterdam Schiphol airport," Tech. rep., Technical report, Dutch Safety Board. Available at http://www.onderzoeksraad.nl/docs/rapporten/Rapport_TA_ENG_web.pdf, 2010.

[85] 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.

[86] Di Donato, P. F. A., Balachandran, S., McDonough, K., Atkins, E., and Kolmanovsky, I., "Envelope Aware Flight Management for Loss of Control Prevention given Rudder Jam," *Journal of Guidance, Navigation and Control (Accepted)*, 2016.

[87] Schuchard, E. A., Melody, J. W., Basar, T., Perkins, W. R., and Voulgaris, P., "Detection and classification of aircraft icing using neural networks," *38th AIAA Aerospace Sciences Meeting and Exhibit*, 2000.

[88] Sharma, V., Voulgaris, P. G., and Frazzoli, E., "Aircraft autopilot analysis and envelope protection for operation under icing conditions," *Journal of guidance, control, and dynamics*, Vol. 27, No. 3, 2004, pp. 454–465.

[89] Sarter, N. B. and Schroeder, B., "Supporting decision making and action selection under time pressure and uncertainty: The case of in-flight icing," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Vol. 43, No. 4, 2001, pp. 573–583.

[90] Lombaerts, T., Schuet, S., Acosta, D., Kaneshige, J., Shish, K., and Martin, L., "Piloted Simulator Evaluation of Maneuvering Envelope Information for Flight Crew Awareness," *AIAA Guidance, Navigation, and Control Conference*, Kissimmee, FL, Jan 2015.

[91] Lombaerts, T., Schuet, S. R., Wheeler, K. R., Acosta, D. M., and Kaneshige, J. T., "Safe maneuvering envelope estimation based on a physical approach," *AIAA Guidance, Navigation, and Control Conference*, Boston, MA, August 2013.

[92] Balachandran, S. and Atkins, E. M., "A Markov Decision Process Framework for Flight Safety Assessment and Management," *Submitted to the Journal of Guidance, Navigation and Control*, 2015.

[93] Wilborn, J. E. and Foster, J. V., "Defining Commercial Transport Loss-of-Control: A Quantitative Approach," *Proc. AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Providence, Rhode Island, 2004.

[94] Politovich, M. K., "Response of a research aircraft to icing and evaluation of severity indices," *Journal of aircraft*, Vol. 33, No. 2, 1996, pp. 291–297.

[95] "Loss of Control on Approach Colgan Air, Inc. Operating as Continental Connection Flight 3407 Bombardier DHC-8-400, N200WQ," Accident report NTSB/AAR-10/01, 2010.

[96] "In-Flight Icing Encounter And Loss Of Control, Simmons Airlines, D.B.A. American Eagle Flight 4184 Avions De Transport Regional (ATR) Model 72-212, N401am Roselawn, Indiana October 31,1994," Accident report NTSB/AAR-96/01, 1996, http://www.ntsb.gov/investigations/AccidentReports/Reports/AAR9601.pdf, April 2016.

[97] Grauer, J. A. and Morelli, E. A., "A Generic Nonlinear Aerodynamic Model for Aircraft," *Proc. AIAA Atmospheric Flight Mechanics Conference*, 2014.

[98] Kirk, D. E., *Optimal Control Theory, An Introduction*, Prentice-Hall, Inc., 1970.

[99] Balachandran, S. and Atkins, E. M., "An Autonomous Override System to Prevent Airborne Loss of Control," *Proceedings of the 28th Innovative Applications of Artificial Intelligence Conference*, Phoenix, AZ, 2016.

[100] Chaslot, G. M.-B., Winands, M. H., and van Den Herik, H. J., "Parallel monte-carlo tree search," *Computers and Games*, Springer, 2008, pp. 60–71.

[101] Sokolowski, J. A., Banks, C. M., and Petty, M. D., *Principles of Modeling and Simulation: A Multidisciplinary Approach*, Wiley online library, 2008.

[102] 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, 1996, pp. 415–418.

[103] Bochot, T., Virelizier, P., Waeselynck, H., and Wiels, V., "Model Checking Flight Control Systems: the Airbus Experience," *Proc. International Conference on Software Engineering*, Vancouver, Canada, 2009.

[104] 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.

[105] 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*, October 2003.

[106] Joshi, A., Heimdahl, M. P. E., Miller, S. P., and Whalen, M. W., "Model-Based Safety Analysis," May 2006, http://shemesh.larc.nasa.gov/fm/papers/Joshi-CR-2006-213953-Model-Based-SA.pdf.

[107] Lygeros, J. and Lynch, N., "On the formal verification of the TCAS conflict resolution algorithms," *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, Vol. 2, IEEE, 1997, pp. 1829–1834.

[108] 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.

[109] 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.

[110] Girard, A., "Reachability of uncertain linear systems using zonotopes," *Hybrid Systems: Computation and Control*, Springer, 2005, pp. 291–305.

[111] Asarin, E., Bournez, O., Dang, T., and Maler, O., "Approximate reachability analysis of piecewise-linear dynamical systems," *Hybrid Systems: Computation and Control*, Springer, 2000, pp. 20–31.

[112] 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.

[113] Kloetzer, M. and Belta, C., "Reachability analysis of multi-affine systems," *Hybrid Systems: Computation and Control*, Springer, 2006, pp. 348–362.

[114] Tabuada, P., *Verification and control of hybrid systems: a symbolic approach*, Springer, 2009.

[115] Lygeros, J., "On reachability and minimum cost optimal control," *Automatica*, Vol. 40, No. 6, 2004, pp. 917–927.

[116] Prajna, S. and Jadbabaie, A., "Safety verification of hybrid systems using barrier certificates," *Hybrid Systems: Computation and Control*, Springer, 2004, pp. 477–492.

[117] Tobenkin, M. M., Manchester, I. R., and Tedrake, R., "Invariant funnels around trajectories using sum-of-squares programming," *Proceedings of the 18th IFAC World Congress*, 2010, doi:10.3182/20110828-6-IT-1002.03098.

[118] Kwiatkowska, M., Norman, G., and Parker, D., "Stochastic model checking," *Formal methods for performance evaluation*, Springer, 2007, pp. 220–270.

[119] 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*, ACM, 2010, pp. 243–252.

[120] Abbas, H., Fainekos, G., Sankaranarayanan, S., Ivančić, F., and Gupta, A., "Probabilistic temporal logic falsification of cyber-physical systems," *ACM Transactions on Embedded Computing Systems (TECS)*, Vol. 12, No. 2s, 2013, pp. 95.

[121] Holzmann, G. J., "The model checker SPIN," *IEEE Transactions on Software Engineering"*, Vol. 23, May 1997, pp. 279–295.

[122] Manna, Z. and Pnueli, A., *The temporal logic of reactive and concurrent systems: specifications*, Vol. 1, Springer Science & Business Media, 1992.

[123] Liu, J., Ozay, N., Topcu, U., and Murray, R. M., "Synthesis of reactive switching protocols from temporal logic specifications," *IEEE Transactions on Automatic Control*.

[124] Kress-Gazit, H., Fainekos, G. E., and Pappas, G. J., "Where's Waldo? Sensor-based temporal logic motion planning," *Robotics and Automation, 2007 IEEE International Conference on*, IEEE, 2007, pp. 3116–3121.

[125] 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*, ACM, 2014, pp. 293–302.

[126] Mamessier, S., Feigh, K., Pritchett, A., and Dickson, D., "Pilot Mental Models and Loss of Control," *AIAA Guidance, Navigation and Control Conference*, National Harbor, MD, 2014.

[127] Federal Aviation Administration, "PART 25 - AIRWORTHINESS STANDARDS: TRANSPORT CATEGORY AIRPLANES [online]," http://www.ecfr.gov/cgi-bin/text-idx?SID=5dc4d5058a9ad16943a2af08556801cd&tpl=/ecfrbrowse/Title14/14cfr25_main_02.tpl[cited May'15].

[128] 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.

[129] Girard, A. and Martin, S., "Control Synthesis for Constrained Nonlinear Systems using Hybridization and Robust Controllers on Simplices," *CoRR*, Vol. abs/1103.2612, 2011.

[130] Clarke, E., Grumberg, O., Jha, S., Lu, Y., and Veith, H., "Counterexample-guided abstraction refinement for symbolic model checking," *Journal of the ACM (JACM)*, Vol. 50, No. 5, 2003, pp. 752–794.

[131] Asarin, E., Dang, T., and Girard, A., "Reachability analysis of nonlinear systems using conservative approximation," *Hybrid Systems: Computation and Control*, Springer, 2003, pp. 20–35.

[132] 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, 2012, pp. 125–134, doi: 10.1145/2185632.2185653.

[133] Rubinstein, R. Y. and Kroese, D. P., *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*, Springer Science & Business Media, 2013.

[134] York, B. W. and Alaverdi, O., "A physically representative aircraft landing gear model for real time simulations," American Institue of Aeronautics and Astronautics, Inc., Jun 1996, Accessed March 2012http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA314062.

[135] Rankin, J., *Bifurcation Analysis of Nonlinear Ground Handling of Aircraft*, Ph.D. thesis, University of Bristol, Bristol,United Kingdom, 2010.

[136] Wong, J., *Theory of Ground Vehicles*, chap. 1, Wiley-Interscience, 2010, pp. 3–84.

[137] Zhang, Y. and Duan, H., "A directional control system for UCAV automatic takeoff roll," *Aircraft Engineering and Aerospace Technology: An International Journal*, Vol. 85, No. 1, 2013, pp. 48–61.

[138] Pacejka, H., *Tyre and Vehcile Dynamics*, chap. 4, Oxford: Elsevier, 2006, pp. 150–202.

[139] 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.

[140] Mitchell, I. M., "A toolbox of level set methods," *Dept. Comput. Sci., Univ. British Columbia, Vancouver, BC, Canada, http://www. cs. ubc. ca/˜ mitchell/ToolboxLS/toolboxLS. pdf, Tech. Rep. TR-2004-09*, 2004.