Guidance, Navigation, and Control and Co-located Conferences
August 19-22, 2013, Boston, MA
AIAA Infotech@Aerospace (I@A) Conference

Flight Safety Assessment and Management during Takeoff

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

The goal of a safety management system is to monitor sensors, identify hazards and mitigate risk to the extent possible. In this paper, we present a novel approach called Flight Safety Assessment and Management (FSAM) to assess key quantitative and qualitative feedback parameters for loss of control (LOC) risk. Decision making logic is specified as a set of hierarchical timed automata models. Logic is customized to phase of flight as well as the control authority and modes. States are classified as nominal, moderate risk and high risk, enabling FSAM to issue warnings and override actions as necessary. FSAM logic for the takeoff phase of flight is presented in this paper.

Nomenclature

FSAM	= Flight Safety Assessment and Management
LOC	= Loss of Control
FMS	= Flight Management System
FMC	= Flight Management Computer
AFDS	= Autopilot and Flight Directory System
AT	= Auto Throttle
EA-FMS	= Envelope-Aware Flight Management System
DFSA	= Deterministic Finite State Automaton
QLC	= Quantitative Loss of Control
RTO	= Rejected Takeoff
TOCW	= Takeoff Configuration Warning
PIC	= Pilot in Command
ETA	= Estimated Time of Arrival
Σ	= Alphabets
S	= States
S_0	= Initial States
G	= Timers
B	= Transitions
V_{mcg}	= Minimum ground control speed with one engine in-operative
V_1	= Takeoff decision speed
V_R	= Rotation speed
V_{LOF}	= Lift off speed
V_2	= Takeoff safety speed
V_{FP}	= Minimum flap retraction speed
V_{EF1}	= Airspeed at single engine failure
V_{EF2}	= Airspeed at multiple engine failure

*Graduate Student, Aerospace Engineering, University of Michigan, Ann Arbor, MI, 48109, Student Member
†Associate Professor, Aerospace Engineering, University of Michigan, Ann Arbor, MI, 48109, Associate Fellow

I. Introduction

Loss of control (LOC) is the leading cause of aviation accidents. LOC can be attributed to factors such as system and component failures, structural damage, inappropriate maneuvers (stall), adverse weather (wind shear, thunderstorms, icing), etc. Due to the wide variety of precursors to a LOC event, no single intervention strategy can be formulated to prevent all LOC situations. To effectively prevent LOC-related accidents, it is necessary to analyze how LOC events unfold. A typical LOC sequence progresses from an off-nominal condition such as vehicle impairment or an external disturbance, to inappropriate crew response, to a vehicle upset condition.[1] The conventional Flight Management System (FMS) used in modern airliners is an integral part of the aircraft's avionics. It automates a wide variety of tasks and hence reduces the pilot's workload. The Autopilot and Flight Director System (AFDS) and Auto Throttle (AT) support automatic flight plan following from takeoff through landing. When active, the AFDS commands the roll attitude to track flight plan heading and sets pitch to hold a specific airspeed and flight path angle or altitude. AT manages the thrust settings to adjust the airspeed according to the current flight plan. The AFDS also can accurately track the glide slope and localizer during approach to landing. The crew may disengage the FMS completely or partially although automation logic may still guide low-level control functions. The FMS systems are triply redundant to ensure backup in case of system failures. Stall and load factor envelope protection are available during normal operations. Despite the excellent safety records of AFDS equipped aircraft, LOC events still occur. LOC events are even more common in aircraft without AFDS as well as in cases where the AFDS has been manually disengaged. Current software logic deactivates the flight director systems when serious off-nominal conditions are detected, leaving the crew to manage the situation. In such situations envelope protection may or may not be available. Inappropriate control inputs from the crew during such off-nominal conditions tend to aggravate the situation,[2] particularly in cases where other factors, e.g., sensor failure, limit situational awareness for the crew and automation.

The Envelope-Aware Flight Management System (EA-FMS), shown in Figure 1, is proposed to prevent LOC by improving capabilities in identifying/updating dynamics, envelope boundaries, and ultimately control authority switching. The EA-FMS can augment the current AFDS to facilitate crew training and fleet transitions as well as certification. EA-FMS monitors the existing AFDS "silently" except in cases of moderate to high LOC risk. EA-FMS is consistent with the Aircraft Integrated Resilient Safety Assurance and Fail Safe Enhancement (AIRSAFE) concept.[3] EA-FMS has six core modules as shown in Figure 1. Resilient control enables robust recovery from LOC events including stall and unusual attitude whether induced by environment, crew, on-board failure or multiple factors.[1,4] Resiliency can be achieved in cases that can be anticipated and modeled a prior in an envelope or model (trim) database.[5,6] Envelope estimation and system identification monitor incoming data and model parameter updates provide data for LOC risk assessment and enable EA-FMS to respect new envelope constraints. The flight planning/guidance task is inactive during nominal flight, engaged only to assist with online envelope discovery or guide the aircraft to a safe state or safe emergency landing. This paper focuses on the Flight Safety Assessment and Management (FSAM) module.

II. Flight Safety Assessment and Management

FSAM, shown in Figure 2, is responsible for real-time assessment of LOC risk, activation of LOC warnings, and resilient control override of the crew. Logic models used by FSAM are intuitive and verifiable to facilitate validation and flight crew understanding. Flight safety assessment begins by computing qualitative and quantitative metrics used to assess loss of control risk and recoverability using the conventional ADFS and EA-FMS adaptive modules. The five quantitative loss-of-control (QLC) envelopes[4] include parameters enabling identification of conditions representing adverse aerodynamics and unusual attitude. Envelopes aimed at ensuring structural integrity as well as dynamic pitch and roll control are also defined. These serve as baseline metrics supplemented by LOC indicators developed by the EA-FMS Envelope Estimation module. Qualitative metrics indicate the presence of sensor and actuator faults, failures, and errors; these values are generated by a combination of component-level diagnostics and the data provided by the Sensor Diagnostic module. Deterministic logic models applicable to a suite of LOC precursor scenarios are developed. While parameters may be estimated with some uncertainty, use of a deterministic model facilitates understanding by the flight crew and analysis of system behavior for certification. The phases of flight ultimately to be considered in logic models include takeoff, climb, cruise, loiter, approach, and landing. These will be

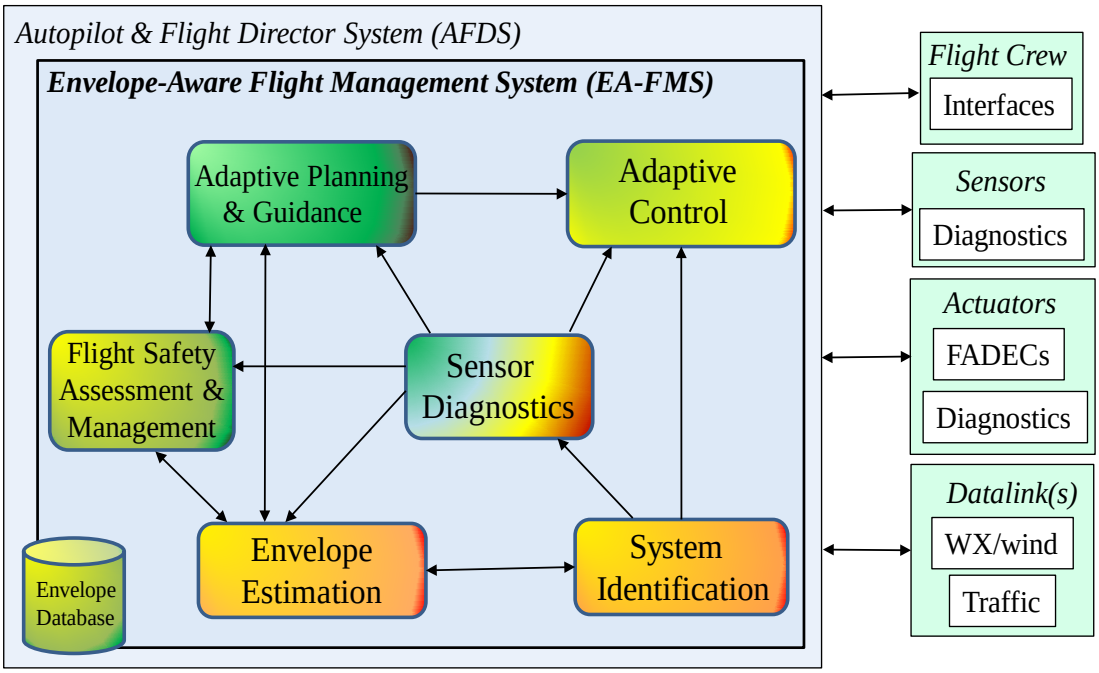

Figure 1. Envelope-Aware Flgiht Management System

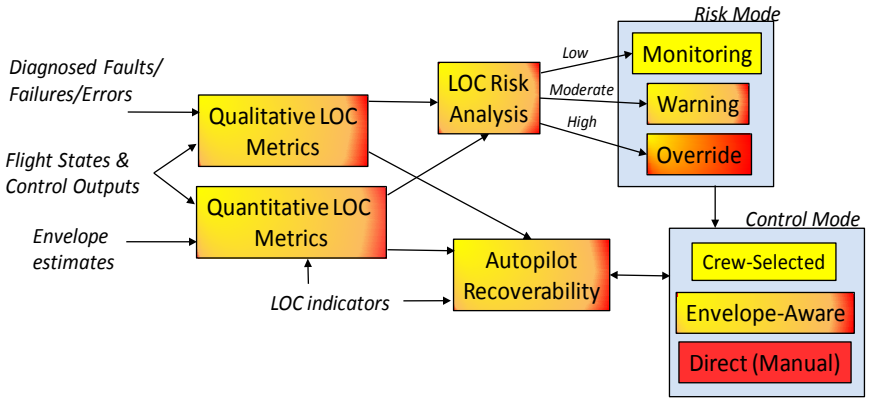

Figure 2. Flight Safety Assessment and Management Module

further divided to manage complexity of each machine. We seek readability of the state machine logic not just for software validation purposes but ultimately by a flight crew who should understand the underlying functionality to expect from the system. Development of logic models for each scenario is guided by operating procedures, checklists, by literature/media surveys and accident reports of various aviation incidents of the past. We employ timed automata[7] to model these deterministic logic models, providing a means to transition between discrete states based on elapsed time as well as observed events.

A. Timed Automaton background

The timed automaton[7] is an extension of the deterministic finite state automaton.[8] 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 set of states, finite alphabet, transition table, initial state, and final state set. The DFSA starts from an initial state and receives as input a sequence of symbols from the machine's alphabet. The alphabet symbols can have different interpretations based on the application of the DFSA. Each symbol or alphabet leads to a unique transition from the current state to another or the same state. The accepting states are distinguished as elements of a final state set. A set of all the accepted strings are termed as the language of the DFSA. The timed automata[7] is a extension of the DFSA where the transition between states are governed by timers along with incoming symbols from the defined alphabet. These timers serve as minimum/maximum state residence time constraints. Figure 3 illustrates a simple example timed automaton used to model a "turnstile" that will only remain open for ten seconds after inserting a coin.

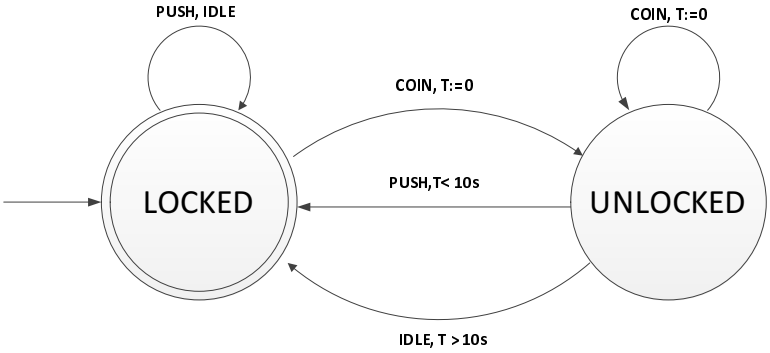

Figure 3. Timed automaton for a turnstile

The automaton in Figure 3 has two states, *locked* and *unlocked*. The *locked* state serves both as the initial and final state. A timer T is initiated and re-initiated whenever a new coin is inserted. Inserting a coin makes the state machine transition to the *unlocked* state. If the turnstile is pushed in the *unlocked* state, it will then transition to the *locked* state. One could also choose to insert coins in the *unlocked* state and hence the state machine will remain in the *unlocked* state. If no entry is attempted within 10 seconds, the turnstile reverts back to the *locked* state. Timed automata can be represented as a timed transition table.[7] A time transition table is a tuple (Σ, S, S_0, G, B) where

$$\Sigma = \{coin, push, idle\} \tag{1}$$

$$S = \{locked, unlocked\} \tag{2}$$

$$S_0 = \{locked\} \tag{3}$$

$$G = \{T\} \tag{4}$$

$$B = \begin{cases} < locked, unlocked, coin, \{T\}, \{\} >, < unlocked, unlocked, coin, \{T\}, \{\} >, \\ < unlocked, locked, push, \{T\}, \{T < 10s\} >, < locked, locked, push, \{T\}, \{\} >, \\ < unlocked, locked, idle, \{T\}, \{T > 10s\} >, < locked, locked, idle, \{T\}, \{\} > \end{cases} \tag{5}$$

The behavior of finite state automata and timed automata can be observed in many devices such as vending machines, elevators, and traffic lights. In the context of Flight Safety Assessment and Management

(FSAM) an aircraft can be viewed as transitioning through a sequence of states or modes, each of which are timed in accordance with a 4D flight plan. At the top layer, modes such as climb, cruise, and descent are appropriate abstractions. Lower-level logic is also important for FSAM. As will be illustrated below, during takeoff, the aircraft transitions through a variety of V-speeds before rotating to initiate the departure climb. The set of risk factors and appropriate reactions are dependent on the lower-level mode as well as phase of flight. In all modes or states, the flight crew/autopilot must initiate appropriate tasks (or actions) to safely follow the flight plan or handle anomalies as they are encountered. In the context of the timed automaton, the "state" must capture information pertinent to the determination of risk and the appropriate action set. "Events" must allow FSAM to transition between states based on observed sensor data as well as observed crew or automation actions. FSAM generates two types of outputs: warnings and control mode override actions. Each is transient, returning control to the crew-directed (pilot) mode when the high-risk situation (with LOC potential) is averted. Below, logic models applicable to a suite of LOC precursor scenarios are developed. First, the high-level FSAM mode switching logic in the form of a timed automaton is presented. Next, a "deeper dive" into takeoff is presented. Development of the logic models for each scenario is guided by standard operating procedures and checklists and by accident and incident reports found in the literature and online databases. The logic models are developed in a hierarchical order to manage complexity and facilitate understanding. The automata at the various levels of hierarchy are discussed in the following sections.

B. Top Level Flight Safety Assessment and Management Logic

The goal of the top level FSAM logic is twofold. First, it aims to break down FSAM's 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 provides an intuitive understanding of FSAM's switching behavior in a graphical context using transitions between flight modes with which crews are already familiar. Figure 4 shows a timed automaton that depicts the progression of flight phases from takeoff to landing. Each state represents a particular phase of flight with nominal phases marked in black. By nominal, we mean the progression from takeoff to landing that follows the original flight plan. The states with a perceptible to moderate level of risk are yellow while states with high risk are red. At this level, off-nominal states arise due to risk factors induced by conditions such as deviations from the flight plan caused by adverse weather, nearby air traffic, or inappropriate crew response. Specific risks due to failure or damage events are identified at lower layers of the automata hierarchy.

The alphabet symbols used by the top level automaton are summarized in Table 1. These symbols are provided to the automaton by a translator that converts incoming flight data to a stream of automaton "alphabet" input symbols. A timer (t) is initiated at the beginning of the takeoff phase to store elapsed time of flight. From takeoff to landing, the aircraft regularly receives status (state and event) updates. At this top layer, the automaton regularly receives a character that indicates current phase (mode) of flight (i.e $M_{climb}, M_{cruise}, M_{descent}$, etc.). Receipt of the same mode character multiple times in succession is handled with reflexive transitions not shown in Figure 4. Each state in Figure 4 has its own sub-machine hierarchy. 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 or within a window around the estimated time of arrival (ETA) (e.g: $M_{climb}, t \leq ETA_1$). The ETAs at each phase are calculated prior to departure as per the original flight plan and updated en route according to changes in flight conditions (e.g. wind, detours due to weather, air traffic etc). An off-nominal condition is flagged by the symbol O. Off-nominal conditions include one or more of the following events: failing to reach the next phase before/at the estimated time of arrival (e.g: $M_{takeoff}, t > ETA_1$), veering off course with respect to the original flight plan, discrepancies in fuel available vs. fuel required for the remainder of the flight, etc.

On receiving the symbol O, the automaton transitions to the corresponding *Flight Plan Alert* state. This is a warning state where the crew is made aware of the off-nominal conditions. No override action is performed at this top layer, although if the problem persists risk states at lower automaton levels can/will trigger related mitigation actions as needed. The symbol O' marks the return to the nominal flight plan. 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 a thunderstorm, other air traffic etc. Such changes are identified by comparing the original flight plan with the flight plan changes made to the navigation computers of the FMS en-route and flagged by the symbol C. Note that the state machine is able to distinguish between intentional (C) and unintentional (O) deviations from the

nominal flight plan. The state machine in Figure 4 has the same structure from takeoff through descent. The approach phase is modeled to take into account events such entering a holding pattern (M_{Hold}) and executing a missed approach ($M_{go-around}$). The automaton recognizes a hold or a missed approach state by monitoring the navigation modes on the Flight Management Computer (FMC) and AT systems. While executing a holding pattern or missed approach, the flight crew and automation must be particularly aware of factors such as remaining fuel and other traffic and must ensure that the aircraft does not experience high risk due to exhausting its reserve fuel or facing a loss-of-separation event.

This top level automaton focuses on modeling phase of flight and elapsed time of flight, with risk flags set when deviating from the flight plan or violating high level constraints. Monitoring the flight plan can help avoid flight plan deviations due to various factors such as lack of situational awareness, distraction in the cockpit, or automation errors. The "bookkeeping" strategy used by the timed automaton is analogous to tools used by the pilot community today to keep track of factors such as arrival times at key waypoints / decision points in the flight.[9]

Table 1. Symbols used in the top-level timed automaton

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 mode(phase) of flight
O	Flight plan fault (delay in flight leg/off course/fuel warning etc...)
O'	Return to nominal flight plan
RTO	Rejected takeoff
t	Elapsed time

The state machine in Figure 4 is formally represented by the tuple (Σ, S, S_0, G, B) where Σ represents the alphabet, S represents the states, S_0 represents the initial state, G represents the timer and B represents the edges in Figure 4. No final or failure states are distinguished.

$$\Sigma = \{M_{climb}, M_{cruise}, M_{descent}, M_{approach}, M_{landing}, M_{hold}, M_{go-around}, O, O', C, RTO\} \quad (6)$$

$$S = \begin{cases} Takeoff - \ldots, Climb - \ldots, Cruise - \ldots, Descent - \ldots, \\ Approach, Landing, Abort, \\ Hold, Missed\ Approach \end{cases} \quad (7)$$

$$S_0 = \{Takeoff - Nominal\} \quad (8)$$

$$G = \{t\} \quad (9)$$

$$(10)$$

III. Takeoff phase of Flight

Takeoff is one of the most safety-critical and difficult phases of flight, second only to final approach and landing. To build a credible FSAM capability for takeoff, we first examined causal factors in takeoff-related accidents. Ninety-seven rejected takeoff (RTO) runway overrun accidents and incidents have been reported from 1960-2000. These have given rise to over more than 400 fatalities.[1] A survey of causal factors from accidents that occurred during the takeoff phase is summarized in Figure 5.[10] These contributing factors are grouped into the following four categories:

1. Improper management of takeoff decision speeds

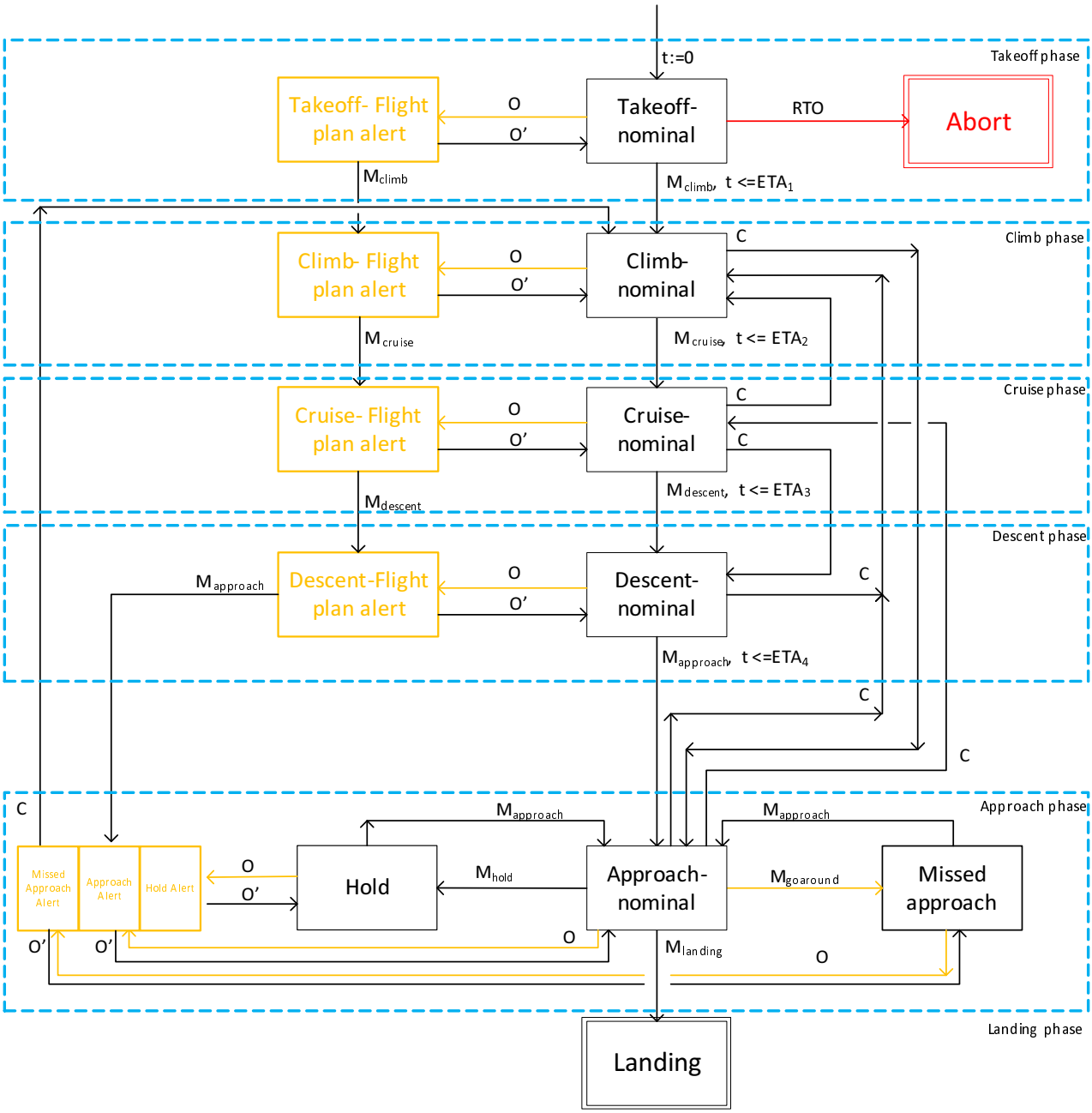

Figure 4. FSAM top level timed automaton

- RTO (rejected takeoff) procedure initiated after V_1.
- No time for the RTO before veering off the runway.
- Rotation above V_R.
- Premature rotation (below V_R).

2. Pilot error

- Non compliance with standard operating procedures.
- Improper crew resource management.
- Rotation not attempted.

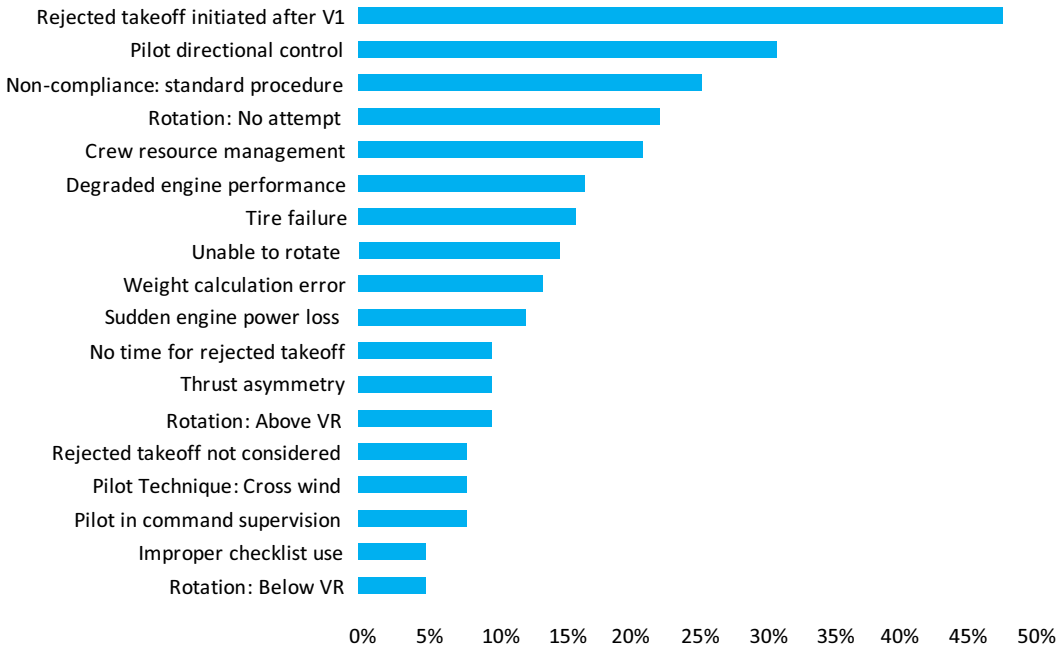

Figure 5. Factors leading to accidents during take-off[10]

- Improper checklist use.
- Inadequate supervision of the vital instruments.
- Aircraft weight calculation and entry errors.

3. Weather
   - Inadequate directional control due to severe cross winds.
   - Improper techniques to handle cross wind takeoffs.
4. System/Component Failures
   - Degraded engine performance.
   - Sudden engine power loss.
   - Thrust asymmetry.
   - Tire failure.
   - Unable to rotate due to degraded actuator performance.

Some contributing factors may be linked to each other and may belong to one or more categories. The goal of FSAM for the takeoff phase is to be able to identify the potential factors that could lead to LOC during takeoff, avoid them if possible or if not deal with risk states through issuance of informative warnings or through override in high risk cases.

In the FSAM takeoff state machines depicted in this paper, the crew is in command of the takeoff; an alternate FSAM formulation in which the automation is also feasible, with similar functionality except that either the crew or adaptive automation must be invoked should the nominal automation not be capable of avoiding (yellow or red) risk states. To develop the FSAM model for takeoff, the following assumptions were made. These will be relaxed as extensions to FSAM logic are made over the course of the project.

- Flight instruments, sensors, onboard computers and flight software (automation) are functional.
- Actuators are working properly.

- No icing (wing/tail contamination or engine icing), damage, or other performance-degrading condition is present.
- No other air traffic pose risk by entering the runway environment.

The ***Takeoff*** state at the top level (shown in Figure 4) expands into a set of states as shown in Figure 6. This forms the second level in the takeoff automaton hierarchy. The ***Pilot in Command*** state has a state machine which monitors the flight crew's inputs and helps avoid pilot error, aids the flight crew in managing the decision speeds appropriately and provides directional control augmentation. Machine specifications for these state machines are described in subsequent sections. The automaton in Figure 6 also has states that handle critical events such as engine failures and tire bursts at various airspeeds during the ground roll.

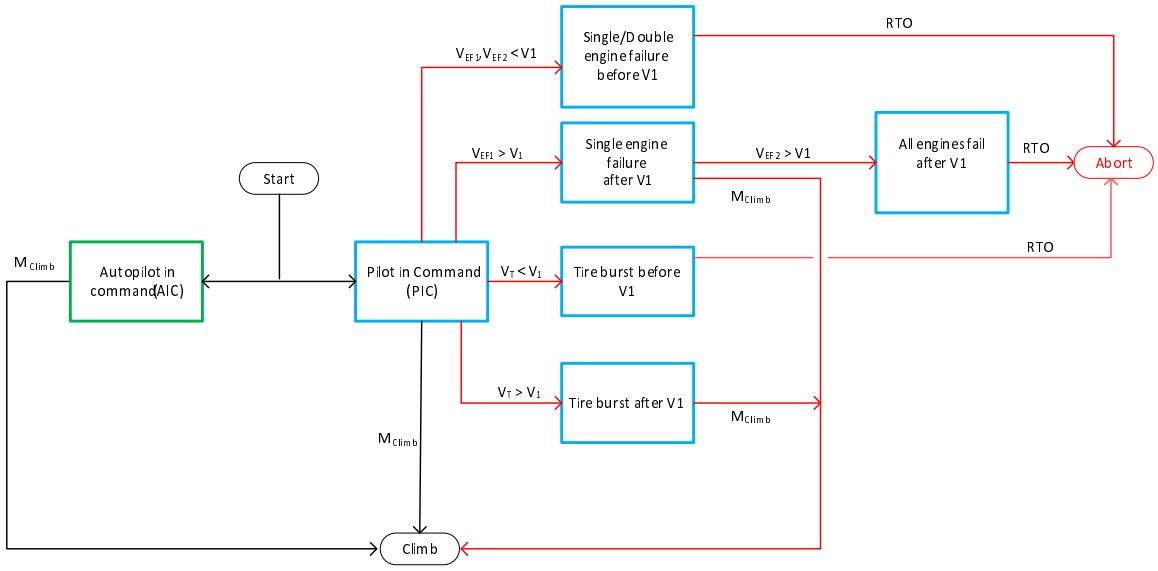

Figure 6. Top-level takeoff phase of flight

A. Machine specification for Pilot in Command State

The takeoff phase requires that the flight crew execute standard operating procedures to configure the aircraft appropriately for takeoff, coordinate with the air traffic controllers for various clearances, and monitor and react to situations encountered throughout the takeoff. In a commercial aviation aircraft, a typical takeoff ground roll lasts around 20 - 35 seconds. During this short time span, the crew must monitor flight instruments and maintain a state of high alertness to make the appropriate decisions in case an anomalous situation is encountered.

The Federal Aviation Regulations (FAR) define several airspeed checkpoints called V-speeds[11] to guide the flight crew in making the appropriate decisions during the takeoff ground roll. The most important V speed is V_1, the decision speed by which the flight crew must decide to continue or reject a takeoff. The flight crew may need to reject a takeoff due to several factors such as single/multiple engine failure(s), tire burst(s), runway incursion, etc. A rejected takeoff initiated after V_1 will leave the aircraft with insufficient runway length remaining to stop safely.

Analogously, rotation initiated before the appropriate V speed can result in an early departure stall.[12,13] Thus, just as V speeds guide flight crews through steps in the takeoff sequence, the V speeds also form an integral part of the decision-making logic in FSAM. Figure 7 graphically represents the various V speeds involved in the takeoff phase. A description of the V speeds are provided in Table 3.

The lower level takeoff logic modules are split into longitudinal and lateral logic, mirroring how the control system is typically decomposed for a fixed-wing aircraft. To emphasize the role played by FSAM in the decision making process, we first introduce a typical manually piloted takeoff phase with no augmentation from FSAM (Figure 8).

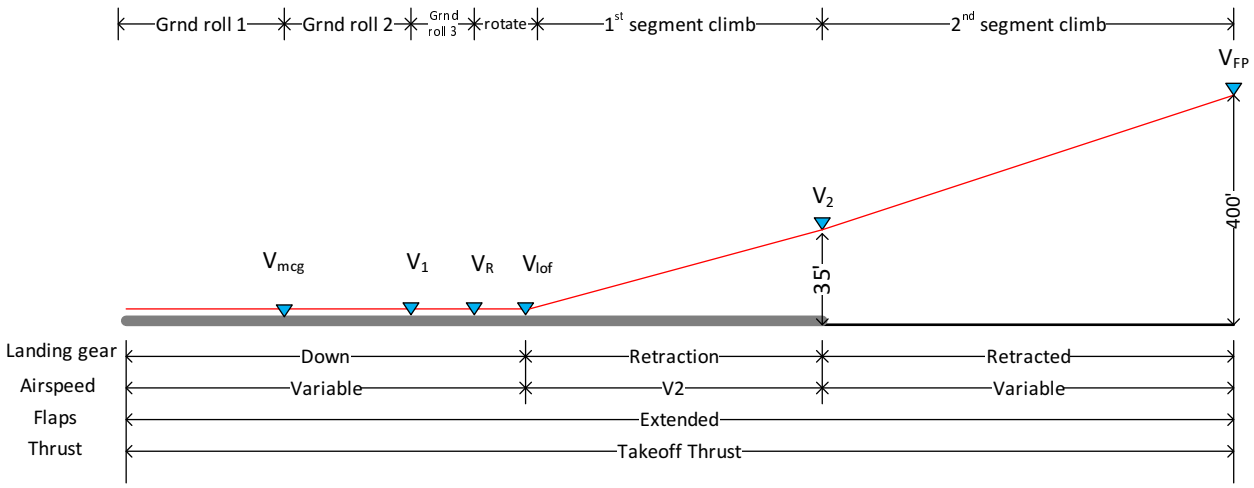

Figure 7. V speeds for takeoff

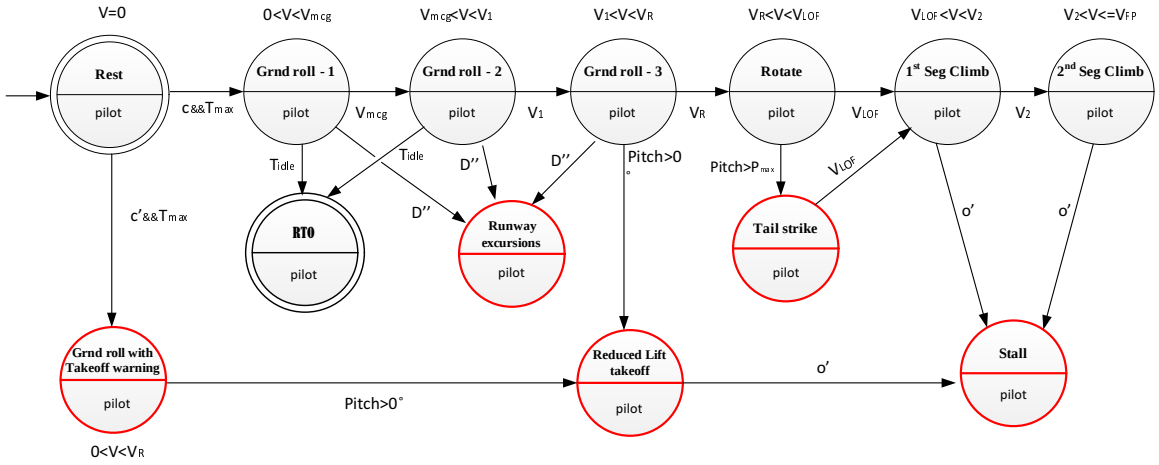

Figure 8. The Takeoff phase with no FSAM augmentation

The takeoff begins with the aircraft in a state of rest (*Rest*) at the end of the runway. The takeoff checklists are completed prior to arriving at the runway and the aircraft is configured for takeoff (c). The pilot flying then sets the required takeoff thrust (T_{max}) and the aircraft transitions from the state of rest to the ground roll state (*Grnd roll*). During this state, the aircraft begins to accelerate down the runway. It passes through a series of airspeeds that partition performance characteristics and takeoff abort constraints.[11,14] The states and alphabet used in the logic model are described in Table 2 and Table 3. As shown in figure 9 each circle in the logic model represents the state of the aircraft during the takeoff phase and the control authority in-charge of the aircraft. Each state is velocity-dependent. The invariants/constraints on velocity are described along with each state. The upper half of each circle represents the current stage of the takeoff phase. The lower half represents the controlling authority (pilot/autopilot/EA-FMS). The double circle represents a final state. The logic model can safely stop executing only in a final state.

As shown in Figure 7 and Figure 8, ground roll is split into a sequence of three nominal takeoff states to match appropriate risk state and action logic with decision-making speeds. It is standard procedure for the pilot not flying to call out the important decision speeds during the takeoff roll. V_1 is the most important decision speed. If required, as in cases of critical engine failures, a decision to reject or abort the takeoff (*RTO*) must be made before V_1. At V_R the aircraft transitions to a state where the pilot flying pulls back on the control column (*Rotation*) to increase the pitch of the aircraft. At V_{LOF} the aircraft is airborne and transitions to the 1st segment climb state ($1^{st}SegClimb$). In this state, once a positive rate of climb is

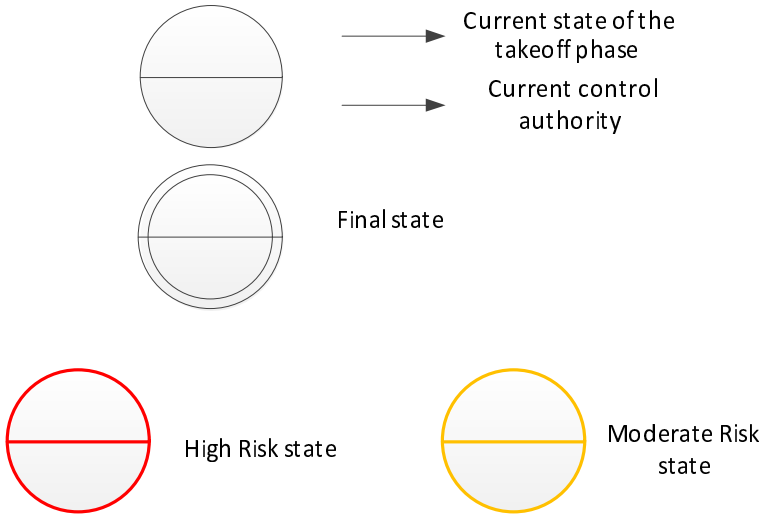

Figure 9. Automaton states

established, the gear is retracted. The aircraft then accelerates to V_2. At V_2 the aircraft is typically 35 ft above the runway threshold. On reaching V_2, a transition to the 2nd segment climb state ($2^{nd}SegClimb$) is made. During this state, the pilot establishes an optimal climb attitude and the aircraft accelerates to V_{FP} at which the flaps/slats are retracted. The takeoff logic disengages at V_{FP} and the next logic model for the climb phase engages.

The above paragraph describes a nominal takeoff sequence. Figure 8 also shows high risk conditions (marked in red) that could lead to accidents during takeoff. These include events such as inappropriate aircraft configuration, rotation initiated before achieving the required speed, excessive pitching during rotation, etc. The augmented logic model in Figure 10 illustrates how the FSAM logic responds to high risk situations. Note that FSAM takeoff logic is split into longitudinal and lateral modes to simplify the state machines. The yellow states are warning states with moderate risk levels. Control authority is transferred to the EA-FMS in these states to execute the necessary actions or override inputs of the crew as needed to prevent LOC. Incident reports on takeoff[12,13] indicate that an inappropriate aircraft configuration (c') was a key factor contributing to takeoff accidents. Many of these incidents occurred despite the presence of warning systems to alert the crew.[13] Thus, if maximum thrust were to be applied with the aircraft inappropriately configured for takeoff (required flaps/slats not deployed, weight calculation errors, mis-trimmed elevator), increased risk is present during the takeoff (T_{max} and c'). The FSAM logic then transitions to the $TOCW$ state. At this point there are two options. The flaps/slats could be deployed before the appropriate rotation speed is achieved or the takeoff could be aborted to prevent accidents. This gives the crew a chance to manually reconfigure the aircraft for takeoff before it is too late. If the crew fails to properly reconfigure the aircraft for takeoff before reaching V_{mcg}, the logic model would transition to the Abort state where EA-FMS would then override the crew to safely abort the takeoff.

During the 3rd ground roll state in Figure 10, if a pre-mature rotation is attempted, the logic model would transition to the $Pre-rotation$ state where FSAM ensures that rotation is made at the appropriate airspeed. During rotation, if the pilot increases the pitch attitude beyond a certain limit (P_{max}), the logic would transition to the $Over-rotation$ state where logic is activated to prevent pitch from exceeding that limit to avoid tail strikes. During the $1^{st}SegClimb$ and $2^{nd}SegClimb$ states, the logic model provides envelope protections similar to envelope protections available on commercial aircraft today. If the aircraft attempts to exit the safe operating envelope for that flight condition (E'), the control authority would be transferred to the EA-FMS that would bring the aircraft back into the safe envelope.

The lateral-plane logic in Figure 11 has the same structure as the longitudinal logic and aims to prevent directional control loss during the initial ground roll. Each state imposes constraints on the lateral position of the aircraft from runway center line as well as the heading, roll angle and the lateral acceleration. Figure 12 illustrates the thresholds on the lateral position of the aircraft from the center line of the runway. If the inner threshold is violated (D'), FSAM logic transfers control to a different authority (e.g: envelope-

aware controller) which then attempts to bring the aircraft within the specified bounds. If the new control authority is still not able to provide directional control (D''), FSAM aborts the takeoff. Of course, numerous issues in validation, verification, and pilot acceptance must be addressed before the "simple" override logic presented here will be certified; the goal in this work is to present an intuitive, deterministic logic model that ultimately can be certified once accepted as reducing rather than increasing risk.

The state machines described in (Figure 10 and Figure 11) form the third level of the takeoff hierarchy (see Figure 13). These state machines alert the crew and provide appropriate response as needed in case of anomalies in the operating procedures. For our current takeoff formulation such response corresponds to managing the decision speeds appropriately and providing directional control augmentation.

Table 2. States in the takeoff automata

States	Description
Rest	Aircraft at rest
Grnd roll - 1	1st ground roll segment
Grnd roll - 2	2nd ground roll segment
Grnd roll - 3	3rd ground roll segment
Rotate	Pilot initiates rotation by pulling back on the control column
1st Seg Climb	1st segment of initial departure climb
2nd Seg Climb	2nd segment of initial departure climb
TOCW	Takeoff configuration warning
Abort	Rejected takeoff initiated by EA
Pre-rotation	Pre-mature rotation
Over-rotation	Excessive rotation
Reduced lift takeoff	Insufficient lift for takeoff
Tail strike	Tail strike due to excessive rotation
Directional control	Directional control provided by EA
Pilot	Pilot in command of the aircraft
EA	Envelope Aware (adaptive) control in command

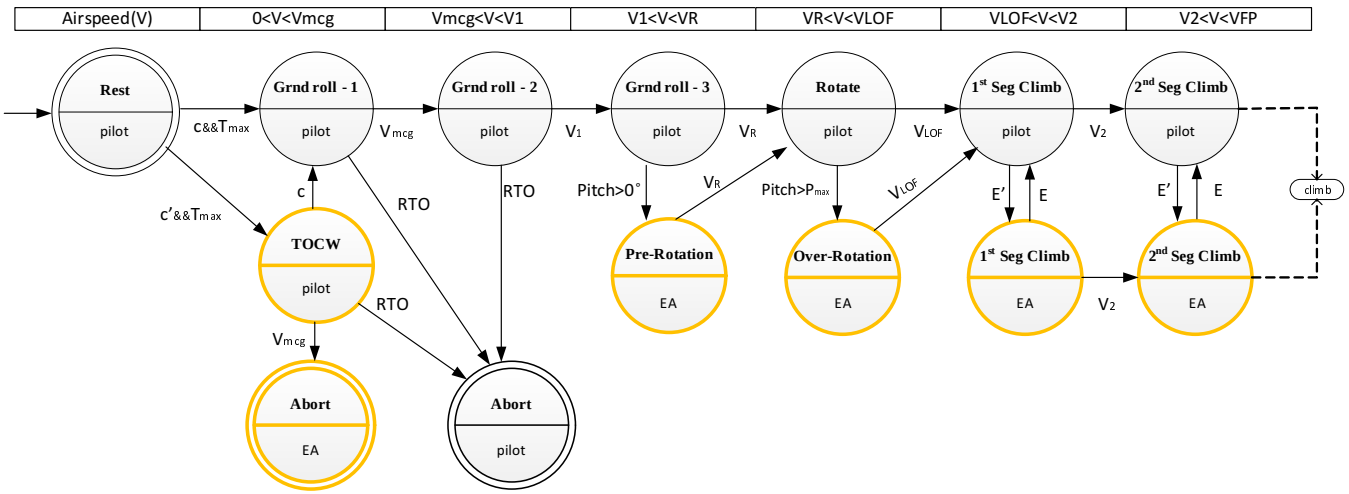

Figure 10. Longitudinal FSAM logic for takeoff

Table 3. Alphabet symbols for the takeoff automata

Alphabet(Σ)	Description
V_{mcg}	Minimum ground control speed with one engine in-operative
V_1	Takeoff decision speed (Go-No Go speed)
V_R	Rotation speed
V_{LOF}	Lift off speed
V_2	Takeoff safety speed
V_{FP}	Minimum flap retraction speed
V_{EF1}	Airspeed at which a single engine fails
V_{EF2}	Airspeed at which a all engine fails (twin engine aircraft)
V_T	Airspeed at which a tire burst occurs
T_{max}	Max/Takeoff thrust setting
RTO	Rejected Takeoff Procedure
c	Configured for takeoff
c'	Improper takeoff configuration (eg: flaps un-deployed)
E	Envelope protection de-activated
E'	Envelope protection activated
Pitch	Pitch attitude of the aircraft
o'	Aircraft flown out of safe envelope thresholds
o	Aircraft flown back into safe envelope
D	Aircraft within inner directional thresholds
D'	Aircraft out of inner directional thresholds (see figure 12 for example)
D"	Aircraft out of outer directional thresholds (see figure 12 for example)

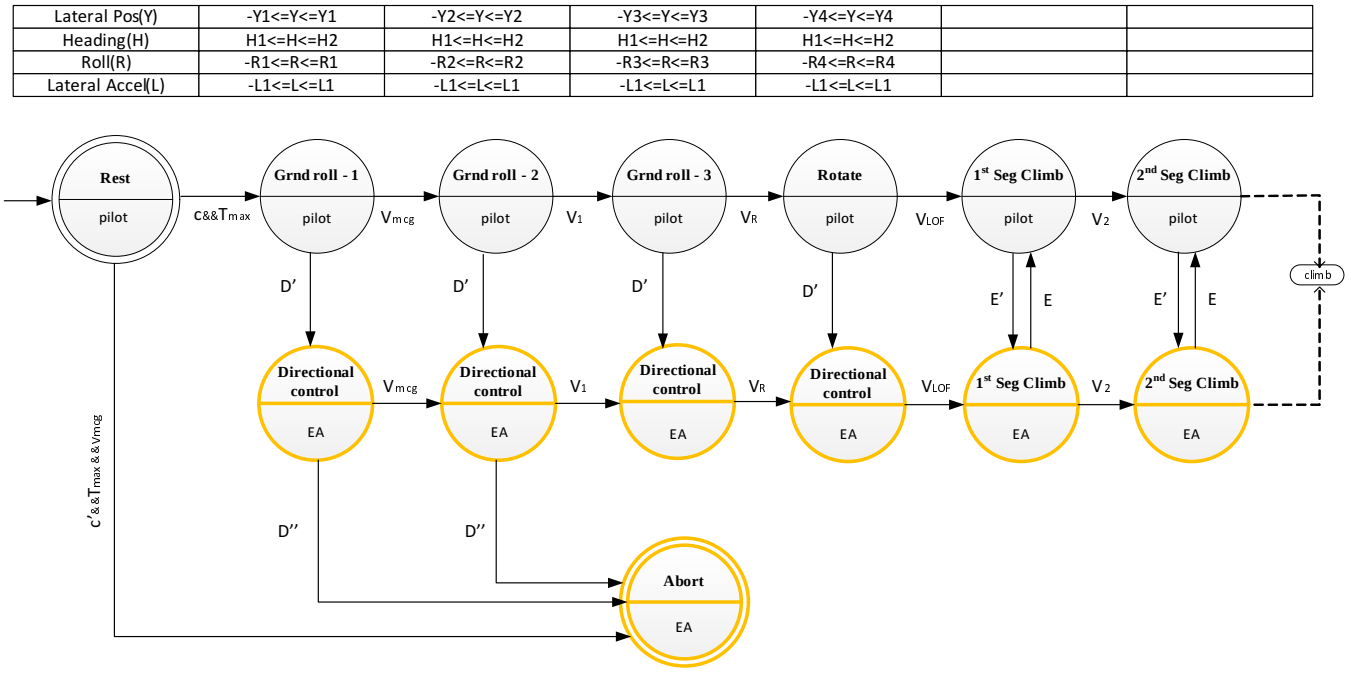

Figure 11. Lateral FSAM logic for takeoff

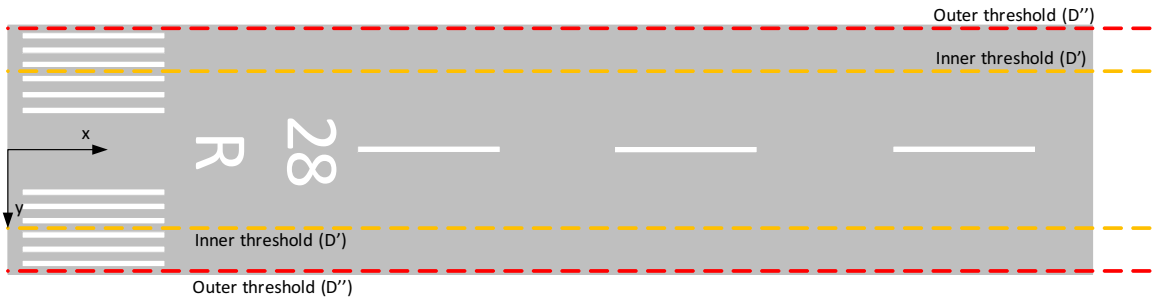

Figure 12. Lateral thresholds on the runway

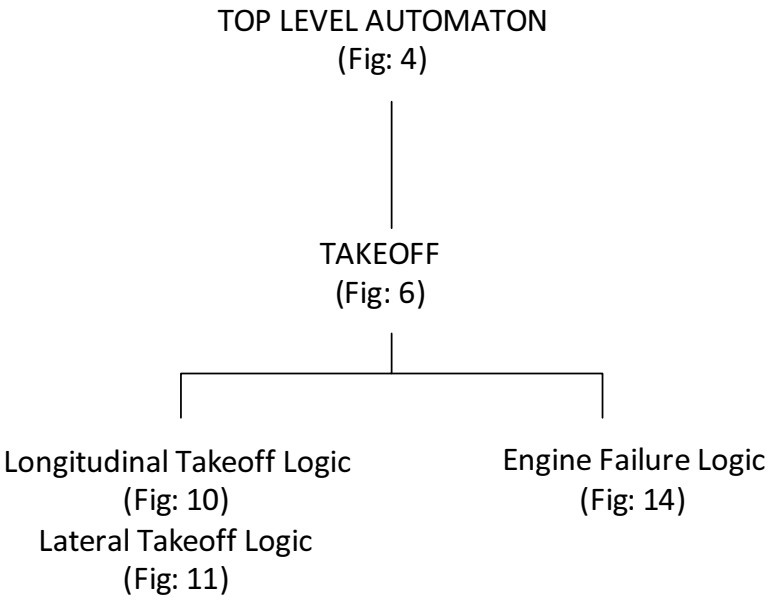

Figure 13. FSAM automata hierarchy

B. Logic for engine failure events

Figure 5 shows that RTO initiated after V_1 is the prime contributing factor for aviation accidents during the takeoff phase. 21 % of RTOs are caused due to engine failures.[14] If an engine fails after V_1, the aircraft must continue to accelerate with its remaining engine(s), lift off and reach V_2 at 35 ft.[14] For an engine failure before V_1, the takeoff must be rejected as soon as possible to stop safely within the available runway length. The earlier in the takeoff roll the failure occurs, the more time the flight crew has to initiate the RTO procedures. Failures occurring closer to the V_1 speed requires the pilots to react faster. The response time of the flight crew to such emergencies may not be fast enough. This delay in the response time may be attributed to the ability of the human brain to perceive the event, to respond to external stimuli, improper crew resource management, loss of situational awareness, etc. The FSAM automation can help the flight crew overcome such challenges. The response time of the automation is substantially faster that that of flight crew to such emergencies, particularly a benefit when immediate RTO is safe given remaining runway and stopping distance constraints. Figure 14 illustrates the FSAM logic for the *Single Double Engine Failure Before V*1 state of the second level in the takeoff hierarchy (Figure 6). The automaton in Figure 6 transitions from the *PIC* state to the *Engine Failure before* V_1 state when it recognizes an engine failure before the V_1 speed ($V_{EF} < V_1$). The goal of this engine failure state is to initiate the RTO procedure as soon as possible and thus avoid unnecessary delays.

Similar logic is also developed to deal with tire bursts during the takeoff ground roll (see Figure 6). Tire bursts can prove problematic since debris from the tires can be ingested into the engine or damage the airframe.

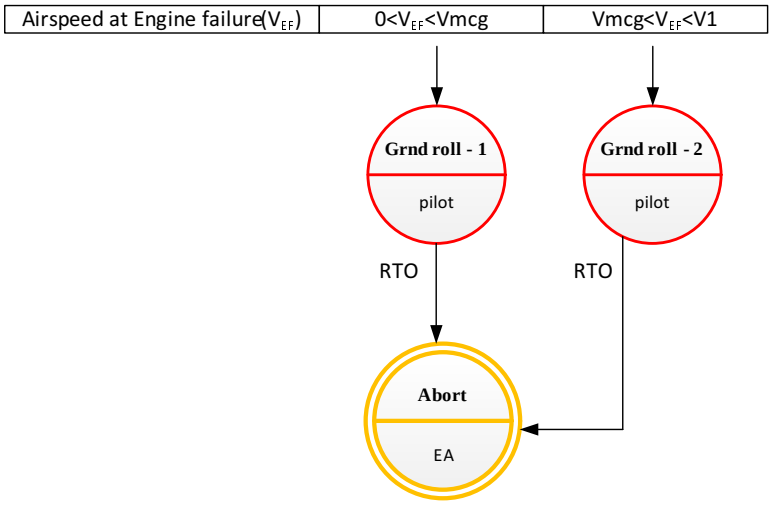

Figure 14. FSAM logic for engine failure before V_1

IV. Conclusion

In this paper, we have presented the concept of flight safety assessment and management and applied it to the takeoff phase of flight. Logic models that address key factors that contribute to LOC during the takeoff phase have been developed. The state machines described thus far collectively manage risk for flight plan following and takeoff given nominal functionality of most systems. Additional events introducing LOC risk such as control actuator failures, control surface jams, instrument failure, and software or communication failures are less probable but prone to introduce high risk of LOC. The hierarchical FSAM machine structure can be extended to incorporate the less probable events as well. Figure 13 illustrates the hierarchical order of all the state machines discussed so far. To analyze the interactions between the FSAM logic models and the flight control laws during the takeoff phase, a physics-based dynamics and control model of takeoff is under development, with aerodynamics and slip models used to establish realistic thresholds for warning and override state transitions in FSAM. The physics based model will serve as a platform to simulate various test scenarios. Aviation accidents that occurred during the takeoff phase will be used as case studies for the

initial validation of these state machines.

Acknowledgments

This work was supported in part by the National Aeronautics and Space Administration under Cooperative Agreement NNX12AM54A.

References

[1]C. M. Belcastro and J. V. Foster, "Aircraft Loss-of-Control Accident Analysis," in Proc. AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario, 2010.

[2]Bureau d'Enquêtes et d'Analyses pour la sécurité de l'aviation civile (BEA), "Aircraft Accident Report - Air France FL 447," 2012, [online database] URL: http://www.bea.aero/en/enquetes/flight.af.447/rapport.final.en.php [cited 25 July 1013]

[3]C. M. Belcastro and S. R. Jacobson, "Future Integrated Systems Concept for Preventing Aircraft Loss-of-Control Accidents," in Proc. AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario, 2010.

[4]J. E. Wilborn and J. V. Foster, "Defining Commercial Transport Loss-of-Control: A Quantitative Approach," in Proc. AIAA Guidance, Navigation, and Control Conference, Providence, RI, 2004.

[5]M. Strube, R. Sanner and E. Atkins, "Dynamic Flight Guidance Recalibration after Actuator Failure," in Proc. 1st AIAA Intelligent Systems Conference, Chicago, IL, 2004.

[6]Y. Tang, E. Atkins and R. Sanner, "Emergency Flight Planning for a Generalized Transport Aircraft with Left Wing Damage," in Proc. AIAA Guidance, Navigation, and Control Conference, Hilton Head, SC, 2007.

[7]R. Alur, "Timed Automata," Theoretical Computer Science, vol. 126, pp. 183-225, 1999.

[8]J. E. Savage, "Models of Computation: Exploring the Power of Computing," Addison-Wesley, pp. 153-207, 1998.

[9]Federal Aviation Administration, "Instrument Flying Handbook," [online database] URL: http://www.faa.gov/regulations_policies/handbooks_manuals/aviation/media/FAA-H-8083-15B.pdf [cited 25 July 1013]

[10]Flight Safety Foundation, "Reducing the risk of runway excursions," [online database] URL: http://flightsafety.org/files/RERR/fsf-runway-excursions-report.pdf [cited 25 July 1013]

[11]Electronic Code of Federal Regulations "Title 14: Aeronautics and Space," [online database] URL: http://www.ecfr.gov/cgi-bin/text-idx?c=ecfr;sid=c2d90aaaf3558c7b326df06c7823b1f5;rgn=div5;view=text;node=14%3A1.0.1.1.1;idno=14;cc=ecfr [cited 25 July 1013]

[12]National Transportation Safety Board, "Aircraft Accident Report - Northwest Airlines, Inc; McDonnnell Douglas DC 9-82,N312RC, Detroit Metropolitan Wayne County Airport, Romulus, Michigan," Washington D.C, 1988

[13]National Transportation Safety Board, "Aircraft accident report: Delta Airlines, Inc. Boeing 727-232, N473DA, Dallas-fort worth international airport," Texas, August, 1988

[14]Federal Aviation Administration, "Pilot guide to takeoff safety," [online database] URL: http://www.faa.gov/other_visit/aviation_industry/airline_operators/training/media/takeoff_safety.pdf [cited 25 July 1013]

[15]Federal Aviation Administration, "Advisory circular: stall and stick pusher training," [online database] URL: http://www.faa.gov/documentLibrary/media/Advisory_Circular/AC%20120-109.pdf [cited 25 July 1013]

[16]Federal Aviation Administration, "Pilot's Encyclopedia of Aeronautical Knowledge," New York, NY:Skyhorse Publishing, Inc., 2007