10.2514/6.2015-0115

AIAA SciTech Forum
5-9 January 2015, Kissimmee, Florida
AIAA Infotech @ Aerospace

A Constrained Markov Decision Process Framework for Flight Safety Assessment and Management

Sweewarman Balachandran* and Ella M. Atkins†

University of Michigan, Ann Arbor, MI 48109

Loss of Control is the most common contributing factor to aviation accidents. Flight Safety Assessment and Management (FSAM) is a high level automation aid to further reduce risk due to loss of control. Nominally, FSAM serves as a loss-of-control watchdog. When off-nominal conditions conducive to loss of control are encountered, FSAM issues appropriate warnings and resilient control overrides to ensure safe operation of the aircraft. This paper describes a framework for modeling FSAM as a Constrained Markov Decision Process where constraints represent flight envelope boundaries and decisions represent control mode overrides or no-operation (continue monitoring without action). The decisions made by FSAM are based on an optimal policy that minimizes a cumulative cost that penalizes high risk flight conditions which contribute to loss of control. Using this CMDP framework we develop policies that prevent loss of control during takeoff. We also illustrate the advantages of using the CMDP approach over a more conventional (unconstrained) MDP approach.
[*Errata: 05/17/2016 - New equation numbers added. Typological errors in (15) - (22) were fixed*]

Nomenclature

EA	**Envelope-Aware**
EA-FMS	**Envelope-Aware Flight Management System**
FSAM	**Flight Safety Assessment and Management**
LOC	**Loss of Control**
MPD	**Markov Decision Process**
CMDP	**Constrained Markov Decision Process**
α, β	**Angle of attack, side slip angle**
ϕ, θ, ψ	**Roll, pitch and yaw angles**
X	**Longitudinal position on the runway**
H	**Altitude**
V	**True airspeed**
$\delta_e, \delta_a, \delta_r, T$	**Elevator, aileron, rudder and thrust control inputs**
P	**Pilot control mode**
AP	**Safety autopilot control mode**
Θ	**Dynamic pitch**
$\mathcal{S}$	**MDP states**
$\mathcal{A}$	**MDP actions**
$\mathcal{R}$	**MDP rewards**
$\mathcal{P}$	**MDP transition probability tensor**
λ	**MDP discount factor**
$\mathcal{T}$	**Abstraction map**

I. Introduction

Loss of Control (LOC) is one of the fundamental causes of aviation accidents. LOC can be attributed to factors such as inappropriate pilot response, hazardous weather conditions, malfunctioning systems and more.[1–3] Often, LOC is a result of complex interactions between two or more of these contributing factors.

In our previous publications,[4–6] we proposed the Envelope Aware Flight Management System (EA-FMS) that supports LOC prediction, prevention, and recovery through an integrated suite of adaptive algorithms[7,8]. The Flight Safety Assessment and Management (FSAM) module of the EA-FMS is responsible for real time assessment of LOC risk and activation of LOC warnings and resilient control override of the flight crew or nominal automation in each

*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

phase of flight (i.e. takeoff, climb, cruise, descent and landing). We developed an FSAM module for the takeoff phase of flight. A manually constructed resilient control override strategy prevented LOC during takeoff and was implemented using a deterministic Moore Machine framework.

A deterministic Moore Machine FSAM formulation requires a domain expert to manually construct the finite state machines that enable resilient control overrides to prevent LOC. Subsequently, hardware and software systems would undergo a rigorous verification and validation process to ensure they satisfy functional, performance, and safety requirements.[9–11] The FSAM module developed using the deterministic approach must also be verified and validated to ensure that it satisfies requirements and enables the aircraft to remain within the safe operating envelope using traditional software certification methods[10]. However, if the system enters an unsafe state that violates a given requirement, the initial system design has to be modified accordingly. This iterative process is called model checking[11–13] and is carried out until all requirements imposed on the system are satisfied. The model checking process is tractable when the system under consideration has a small number of states. The complexity of the model checking process increases with the increase in the dimensionality of the system state space, making it increasingly difficult to test all possible executions of the system to ensure that the probability of entering an unsafe state remains below a designated threshold.

In this paper, we present a novel approach to resilient control that ensures that the probability of entering an unsafe state remains below an acceptable threshold. In the context of FSAM, we use a Constrained Markov Decision Process (CMDP)[14] to construct an optimal policy that selects the appropriate control authority in situations where unacceptable risk is encountered with a default operating mode. The policy generated using a CMDP is less prone to error when compared to a manually constructed strategy for resilient control override. Furthermore, the policy generation process is less labor-intensive. For example, in a conventional MDP formulation, the decision maker would have to verify the resulting MDP policy to ensure that it prevented the system from reaching unsafe states. However, these unsafe states can be specified as constraints in the CMDP framework and hence the resulting policy is guaranteed to satisfy the safety properties imposed on the system. The CMDP formulation applied to FSAM can be prohibitively complex. However, we manage this complexity by decomposing the overall MDP to smaller MDPs that address LOC with respect to each phase of flight; takeoff, climb, cruise, approach and landing.

The rest of the paper is organized as follows. Section II provides the necessary background for MDPs. In Section III we illustrate a conventional MDP framework for developing an optimal policy for FSAM for the takeoff phase. In Section IV, we develop a mechanism that enables us to incorporate hard constraints into the MDP framework. Section V illustrates the optimal policy generation for FSAM using the CMDP framework. We also discuss the benefits of using a CMDP framework over a conventional MDP. Section VI provides conclusions and describes potential future research directions.

II. Background

A discrete-time fully observable MDP[15,16] is represented as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ represents a finite set of all possible states of the system. $\mathcal{A}$ represents a finite set of actions that can be executed. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ represents the transition probabilities associated with transitions from a given state to another state by executing an action. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ represents a reward function that assigns a finite number to each state-action pair. The actions $a_n \in \mathcal{A}$ at each decision epoch are chosen such that they maximize the expected cumulative discounted reward function of the form

$$\mathcal{V}(\mathcal{S}_n) = \mathbb{E}\Big[\sum_{n=0}^{\infty} \lambda^n \mathcal{R}(\mathcal{S}_n, \mathcal{A}_n)\Big] \tag{1}$$

Here, $\mathcal{S}_n$ is the current state, $\mathcal{A}_n$ is the action selected at the current state. $\lambda \in (0, 1]$ is a discount factor that emphasizes short term rewards. The optimal policy is then defined as the mapping π where

$$\pi : \mathcal{S} \rightarrow \mathcal{A}$$

Eqn 1 can be maximized and subsequently the optimal policy can be found using value iteration, policy iteration or a linear programming algorithm[15,16].

III. An MDP Formulation for Safe Takeoff

In this section, we first present a simple MDP formulation that enables us to avoid LOC risk during takeoff. We then motivate the necessity for a constrained MDP formulation with the help of an example.

LOC is often a result of complex interactions between aircraft dynamics, inappropriate control inputs (both pilot and automation), improper configuration, and adverse weather phenomena. Thus, to address the problem of LOC, a decision maker must consider all contributing factors to LOC risk. An MDP to assess and act to preserve flight safety needs to consider the following features as part of its state formulation: aircraft dynamics and control, aircraft configuration and health, operator-related features and environmental features. The actions should represent override, warn and no-operation (monitor) directives.

A. State Representation

In this paper, we consider an FSAM capability focused on preventing LOC during takeoff. LOC during takeoff is attributed to several factors such as improper rejected takeoff procedures, poor directional control, inappropriate rotation techniques, runway overruns, etc. In this paper, we focus attention on a particular takeoff LOC factor: inappropriate rotation techniques. Consequently, we consider only the aircraft dynamics and control state feature as part of our MDP state formulation. This MDP state has the following form:

$$\mathcal{S} = \{\mathcal{S}_i\}\ i = 1, ..., n$$

where $\mathcal{S}_i$ is defined as

$$\mathcal{S}_i = [V, X, \Theta, H, M] \tag{2}$$

Here V is the airspeed, X is the longitudinal position of the aircraft on the runway and Θ is the dynamic pitch of the aircraft[17]. Θ is defined as $\theta + q$ where θ is the pitch attitude of the aircraft and q is pitch angular rate. H is the altitude of the aircraft. M is the control mode of the aircraft. State features V, X, Θ, H can each take a range of values in $\mathbb{R}$ and $M \in \{P, AP\}$ where P denotes that the pilot is in control and AP denotes that the autopilot is in control.

Since FSAM is primarily a passive system that issues override directives only when LOC conditions are encountered, the MDP formulation has two actions; no-operation (NOOP) and override (OVRD):

$$\mathcal{A} = \{NOOP, OVRD\}$$

Here $NOOP$ denotes no-operation where FSAM simply monitors the operation of the aircraft and does not interfere with ongoing actions of the flight crew and nominal autopilot. $OVRD$ denotes an action where FSAM overrides the current control authority with another control authority which can appropriately handle a situation with high LOC risk.

The state representation for the MDP consists of the continuous-valued states (V, X, Θ, H) and discrete valued states M. The MDP requires a discrete abstraction of continuous-valued variables. With knowledge of the takeoff dynamics and aircraft envelopes for the takeoff phase, we develop the following abstraction maps:

$$\mathcal{T}_1 := V \times X \rightarrow S \tag{3}$$

$$\mathcal{T}_2 := \Theta \times H \rightarrow G \tag{4}$$

$$\mathcal{T}_3 := S \times G \rightarrow Q \tag{5}$$

Here, abstraction map $\mathcal{T}_1$ transforms continuous states V and X to discrete states in set $S = \{s_1, ..., s_{17}\}$ as shown in Fig 1. Fig 1 indicates the performance envelopes for a given runway length (field length)[4]. For example, an aircraft operating at the maximum field limit weight will accelerate from rest along the solid green curve, lift off and reach V_2 speed will overshoot the runway. Consequently, any trajectory to the left of this green curve will end up overshooting the runway before reaching the V_2 speed. Similarly, the dashed pink lines indicate the points in $V - X$ space at which a rejected takeoff can be safely initiated and the aircraft can be stopped by the end of the runway. Consequently, any rejected takeoff initiated to the right of the dashed pink curve will overshoot the runway before the aircraft can fully stop.

Abstraction map $\mathcal{T}_2$ abstracts continuous states Θ and H to discrete states in set $G = \{g_1, \ldots, g_8\}$ (see Fig 2). Abstraction map $\mathcal{T}_3$ maps discrete state S and G to discrete states in set $Q = \{q_1, \ldots, q_{136}\}$. Thus, the state q_1 denotes the tuple (s_1, g_1) which in turn denotes the set of states in the velocity (V) - position (X) space represented by partition

1 in Fig 1 and the set of states with dynamic pitch (Θ) - altitude (H) space represented by partition 1 in Fig 2. Table 1 illustrates abstraction map $\mathcal{T}_3$.

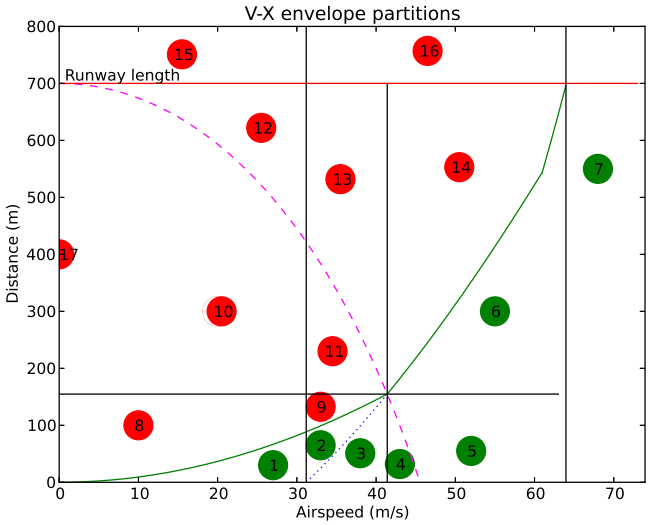

Figure 1. V-X envelope partitions

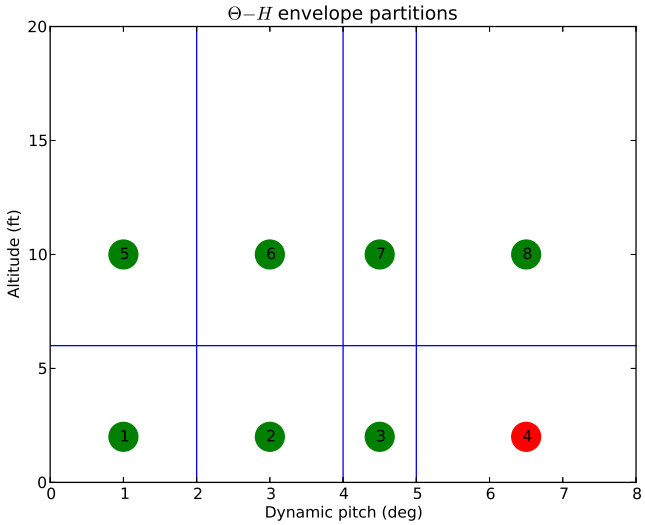

Figure 2. $\Theta - H$ envelope partitions

After applying the above abstraction maps, the MDP state space described by (2) reduces to

$$\mathcal{S}_i = [q, M] \tag{6}$$

where $q \in Q$.

B. Reward formulation

The above abstractions enable us to identify the safe and unsafe flight envelope regions. For example, the partitions indicated in green in Fig 1 are states that ensure that the aircraft has sufficient acceleration to lift off and reach the

Table 1. State space indexing

Q	(S, G)
q_1	(s_1, g_1)
q_2	(s_2, g_1)
q_3	(s_3, g_1)
$\vdots$	$\vdots$
q_8	(s_1, g_2)
$\vdots$	$\vdots$
q_{136}	(s_{17}, g_8)

appropriate performance speeds for the climb out phase. The partitions in red indicate states that characterize improper longitudinal acceleration during ground roll which could lead to runway overruns. Similarly, the red partition in Fig 2 characterize over-rotation which could result in a tail strike. Consequently, this compact representation of the state space enables us to formulate the reward function so that we can penalize states that are unsafe and reward states that are safe. The following MDP reward function is proposed:

$$\mathcal{R}(S_i, \mathcal{A}_j) = \alpha_1 \mathcal{R}_1(\mathcal{S}_i, \mathcal{A}_j) + \alpha_2 \mathcal{R}_2(\mathcal{S}_i, \mathcal{A}_j) + \alpha_3 \mathcal{R}_3(\mathcal{S}_i, \mathcal{A}_j) + \alpha_4 \mathcal{R}_4(\mathcal{S}_i, \mathcal{A}_j)$$

where

$$\mathcal{R}_1(\mathcal{S}_i, \mathcal{A}_j) = \begin{cases} -1 & if\ M = P\ and\ \mathcal{A}_j = OVRD \\ 0 & else \end{cases} \tag{7}$$

$$\mathcal{R}_2(\mathcal{S}_i, \mathcal{A}_j) = \begin{cases} -1 & if\ M = AP\ and\ \mathcal{A}_j = NOOP \\ 0 & else \end{cases} \tag{8}$$

$$\mathcal{R}_3(\mathcal{S}_i, \mathcal{A}_j) = \begin{cases} -1 & s \leq 3\ and\ g = 3 \\ 0 & else \end{cases} \tag{9}$$

$$\mathcal{R}_3(\mathcal{S}_i, \mathcal{A}_j) = \begin{cases} -1 & g = 4 \\ 0 & else \end{cases} \tag{10}$$

We note here that the $\mathcal{R}_i$'s are normalized. The term $\mathcal{R}_1$ penalizes an override action. This prevents FSAM from issuing unnecessary override directives. The term $\mathcal{R}_2$ penalizes staying in autopilot control over multiple state transitions. This encourages an MDP policy that transfers control back to the pilot when the aircraft is deemed to be inside the safe envelope. The term $\mathcal{R}_3$ penalizes premature rotations and $\mathcal{R}_4$ penalizes over-rotations that can lead to tail strikes during the rotation stage. $\alpha_i > 0$ are weights that enable us to emphasize different components of the reward function.

C. Estimation of Transition Probabilities

Equations of motion for the takeoff phase developed in a previous publication[5] are used to simulate trajectories of the takeoff phase. Mathematical models of human pilot behavior are used to model the inputs of the pilot during takeoff[10,18]. An autopilot controller that is capable of preventing a tail strike during takeoff is used as the override controller. A transition probability model that describes the probability of transition between the various states under different control authorities (pilot/autopilot) is obtained using Monte Carlo simulations with the above takeoff simulation framework.

We first obtain the optimal policy for the above unconstrained MDP using a simple value iteration algorithm. For this illustration, the following values of α_i were chosen; $\alpha_1 = 5$, $\alpha_2 = 50$, $\alpha_3 = 1$ and $\alpha_4 = 10$. The directed graph in Fig 3 illustrates the optimal policy and the evolution of the states as a result of applying this policy. Each node represents a state (as described by Eqn (6)) and the edges represent the transitions as a result of selecting the optimal policy action ($NOOP/OVRD$) . The probability of transition is also indicated on the edges.

In the presented tail strike case study, the chosen weights penalized staying in autopilot control more than penalizing an override action. Also, a tail strike state was penalized more than a premature rotation state. Consequently, FSAM chooses to override the pilot on reaching a state beyond which the probability of entering into a tail strike is

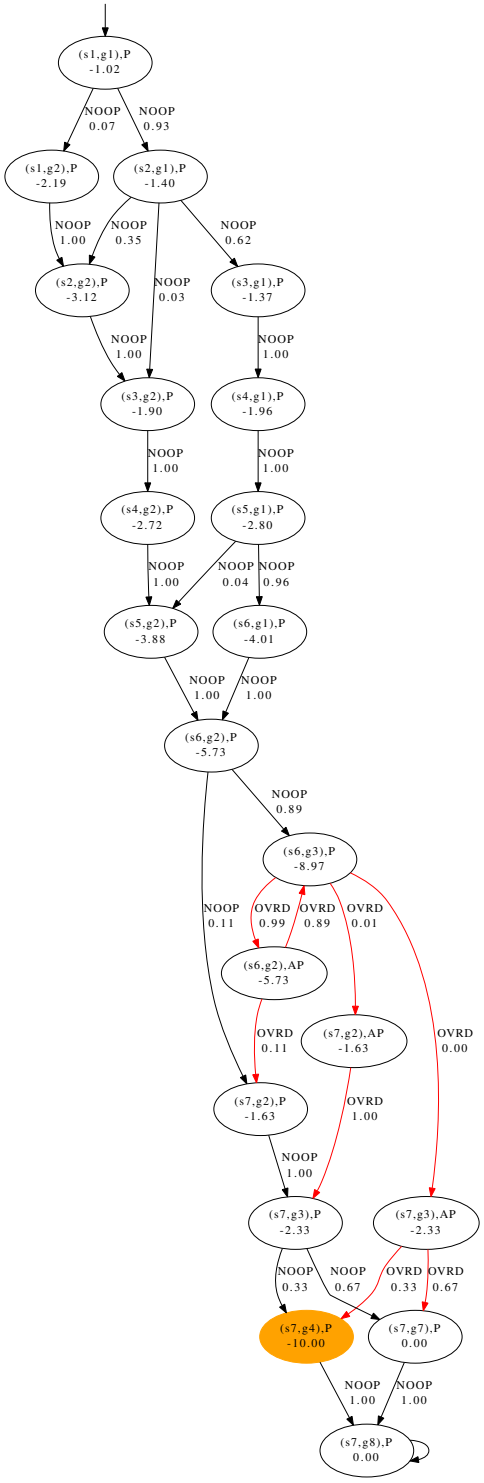

Figure 3. MDP without constraints

very high (s_6, g_3). This prevents an aircraft tail strike at (s_6, g_4). However, as a result of the heavy penalty for staying in autopilot control, FSAM then transfers control back to the pilot. This leads to the possibility of a tail strike at (s_7, g_4) (shown in yellow). The obtained policy did not fully eliminate the possibility of a tail strike due to competing reward terms. This can be avoided by placing a significantly larger penalty on the tail strike state, making FSAM stay in the autopilot mode until the aircraft was free from entering a tail strike state. This method of analyzing the policy to identify high risk states and recomputing the policy with different weighting factors to obtain the desired system behavior can be cumbersome especially if the state space is very large.

IV. Constrained MDP formulation

The goal of this section is to construct a constrained MDP[16,19] 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 (1) subjected to constraints of the form

$$\begin{aligned} p(\mathcal{S}_1^*|\mathcal{S}_0) &\leq p_1 \\ p(\mathcal{S}_2^*|\mathcal{S}_0) &\leq p_2 \\ &\vdots \\ p(\mathcal{S}_m^*|\mathcal{S}_0) &\leq p_m \end{aligned} \tag{11}$$

Here $p(\mathcal{S}_i^*|\mathcal{S}_0)$ is the conditional probability of entering state $\mathcal{S}_i^*$ from a given initial state $\mathcal{S}_0$.

The expected value or utility of state $\mathcal{S}_0$ when acting according to policy π is given by

$$\mathcal{V}(\mathcal{S}_0)_\pi = \mathbb{E}\Big[\sum_{n=0}^{\infty} \lambda^n \mathcal{R}(\mathcal{S}_n, \mathcal{A}_n)\Big]_{\mathcal{S}_0} \tag{12}$$

For a Markov process, Eqn (12) can be expressed as

$$\begin{aligned} \mathcal{V}(\mathcal{S}_0) &= \sum_{n=0}^{\infty} \sum_{\mathcal{S}_i \in \mathcal{S}} \sum_{\mathcal{A}_j \in \mathcal{A}} \lambda^n p(\mathcal{S}_n = \mathcal{S}_i, \mathcal{A}_n = \mathcal{A}_j|\mathcal{S}_0) \mathcal{R}(\mathcal{S}_n = \mathcal{S}_i, \mathcal{A}_n = \mathcal{A}_j) \\ &= \sum_{\mathcal{S}_i \in \mathcal{S}} \sum_{\mathcal{A}_j \in A} \rho(\mathcal{S}_i, \mathcal{A}_j)_{\mathcal{S}_0}^{\pi} \mathcal{R}(\mathcal{S}_n = \mathcal{S}_i, \mathcal{A}_n = \mathcal{A}_j) \end{aligned} \tag{13}$$

Here $\rho(\mathcal{S}_i, \mathcal{A}_j)_{\mathcal{S}_0}^{\pi}$ is defined as the occupational measure of the state-action pair $(\mathcal{S}_i, \mathcal{A}_j)$.

$$\rho(\mathcal{S}_i, \mathcal{A}_j)_{\mathcal{S}_0}^{\pi} := \sum_{n=0}^{\infty} \lambda^n p(\mathcal{S}_n = \mathcal{S}_i, \mathcal{A}_n = \mathcal{A}_j|\mathcal{S}_0) \tag{14}$$

The occupational measure is the discounted total probability of reaching a state $\mathcal{S}_i$ and executing an action $\mathcal{A}_j$ as a result of starting in state $\mathcal{S}_o$ and acting according to policy π. The sum of the occupational measure of state $\mathcal{S}_i$ over all possible actions $\mathcal{A}_j \in \mathcal{A}$ is obtained from Eqn (14) as follows

$$\begin{aligned} \sum_{\mathcal{A}_j \in \mathcal{A}} \rho(\mathcal{S}_i, \mathcal{A}_j) &= \sum_{\mathcal{A}_j \in \mathcal{A}} \sum_{n=0}^{\infty} \lambda^n p(\mathcal{S}_i, \mathcal{A}_j|\mathcal{S}_0) \\ &= p(\mathcal{S}_0) + \sum_{\mathcal{S}_x \in \mathcal{S}} \sum_{\mathcal{A}_y \in \mathcal{A}} \sum_{n=1}^{\infty} \lambda^{n-1} p(\mathcal{S}_x, \mathcal{A}_y|\mathcal{S}_0) p(\mathcal{S}_i|\mathcal{S}_x, \mathcal{A}_y) \\ &= p(\mathcal{S}_0) + \sum_{\mathcal{S}_x \in \mathcal{S}} \sum_{\mathcal{A}_y \in \mathcal{A}} \rho(\mathcal{S}_x, \mathcal{A}_y)_{\mathcal{S}_0}^{\pi} p(\mathcal{S}_i|\mathcal{S}_x, \mathcal{A}_y) \end{aligned} \tag{15}$$

Here $p(\mathcal{S}_0) = 1$ is the probability of starting in the initial state $\mathcal{S}_0$. This leads to the following expression:

$$\sum_{\mathcal{A}_j \in \mathcal{A}} \rho(\mathcal{S}_i, \mathcal{A}_j) - \sum_{\mathcal{S}_x \in \mathcal{S}} \sum_{\mathcal{A}_y \in \mathcal{A}} \rho(\mathcal{S}_x, \mathcal{A}_y)_{s_0}^{\pi} p(\mathcal{S}_i|\mathcal{S}_x, \mathcal{A}_y) = P(\mathcal{S}_0) \tag{16}$$

Eqns (13) and (16) can be expressed in their respective matrix forms as follows

$$\mathcal{V} = \mathcal{R}^T \rho \tag{17}$$

$$([I\ I \ldots I] - [p_{\mathcal{A}_1}^T p_{\mathcal{A}_2}^T \ldots p_{\mathcal{A}_n}^T])\rho = \beta \tag{18}$$

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 $p_{\mathcal{A}_i} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ is the transition probability matrix for each action $\mathcal{A}_i \in \mathcal{A}$. $\beta \in \mathbb{R}^{|\mathcal{S}|}$ is the initial state distribution with $\beta(\mathcal{S}_0) = 1$ and all other states $\beta(\mathcal{S}_i)$ are zeros. Using Eqn (17) and (18), the problem of maximizing the cumulative reward (Eqn (1)) is formulated as a linear program (LP) as follows

$$\max \mathcal{R}^T \rho \tag{19}$$

subject to the constraints

$$\begin{aligned} ([I\ I \ldots I] - [p_{\mathcal{A}_1}^T p_{\mathcal{A}_2}^T \ldots p_{\mathcal{A}_n}^T])\rho &= \beta \\ \rho &\geq 0 \end{aligned} \tag{20}$$

Note that the solution to Eqn (19) and (20) corresponds to the MDP without constraints (Eqn (1)). The additional constraints imposed by Eqn (11) are expressed as constraints on the occupational measures. For example, consider the constraint

$$p(\mathcal{S}_i|\mathcal{S}_0) \leq p_i$$

The above constraint can be expressed as

$$\sum_{\mathcal{A}_j\in\mathcal{A}} p(\mathcal{S}_i, \mathcal{A}_j|\mathcal{S}_0) \leq p_i$$

$$\sum_{n=0}^{\infty} \lambda^n \sum_{\mathcal{A}_j\in\mathcal{A}} p(\mathcal{S}_n = \mathcal{S}_i, \mathcal{A}_n = \mathcal{A}_j|\mathcal{S}_0) \leq \sum_{n=0}^{\infty} \lambda^n p_i \tag{21}$$

$$\sum_{\mathcal{A}_j\in\mathcal{A}} \rho(\mathcal{S}_i, \mathcal{A}_j) \leq \sum_{n=0}^{\infty} \lambda^n p_i \tag{22}$$

$$\sum_{\mathcal{A}_j\in\mathcal{A}} \rho(\mathcal{S}_i, \mathcal{A}_j) \leq \frac{1}{1-\lambda} p_i$$

$$\bar{e}^T \rho \leq \frac{1}{1-\lambda} p_i \tag{23}$$

Here $\bar{e}$ is a vector of zeros with ones in the positions corresponding to the occupational measures of state $\mathcal{S}_i$. Eqn (19), (20) and (23) comprise the LP formulation for the constrained MDP or CMDP[14,16]. The optimal action for each state $\mathcal{S}_i$ is obtained from the occupational measures as follows

$$p(\mathcal{A}_j|\mathcal{S}_i) = \frac{\rho(\mathcal{S}_i, \mathcal{A}_j)^{\pi}_{\mathcal{S}_0}}{\sum_{\mathcal{A}_j} \rho(\mathcal{S}_i, \mathcal{A}_j)^{\pi}_{\mathcal{S}_0}}$$

V. CMDP for Takeoff

In this section we apply the above CMDP formulation to re-construct a resilient control override strategy for the LOC case illustrated in Section III and illustrate that the CMDP enables 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 $[(s_7, g_4), P]$

$$p([(s_7, g_4), P]|[(s_1, g_1), P]) = 0 \tag{24}$$

i.e. the probability of entering the tail strike state (s_7, g_4), starting from the initial state (s_1, g_1), with the pilot in control (P) is zero. Eqn (24) can be expressed as constraints on the occupational measures of state $[(s_7, g_4), P]$ as illustrated in Eqn (23);

$$\bar{e}^T \rho = 0 \tag{25}$$

We can now solve the constrained MDP using the linear program described by Eqns (19)-(23). The resulting policy is shown in Fig 4. It can be seen that the new policy has no risk of tail strike (i.e. no (s_7, g_4) 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.

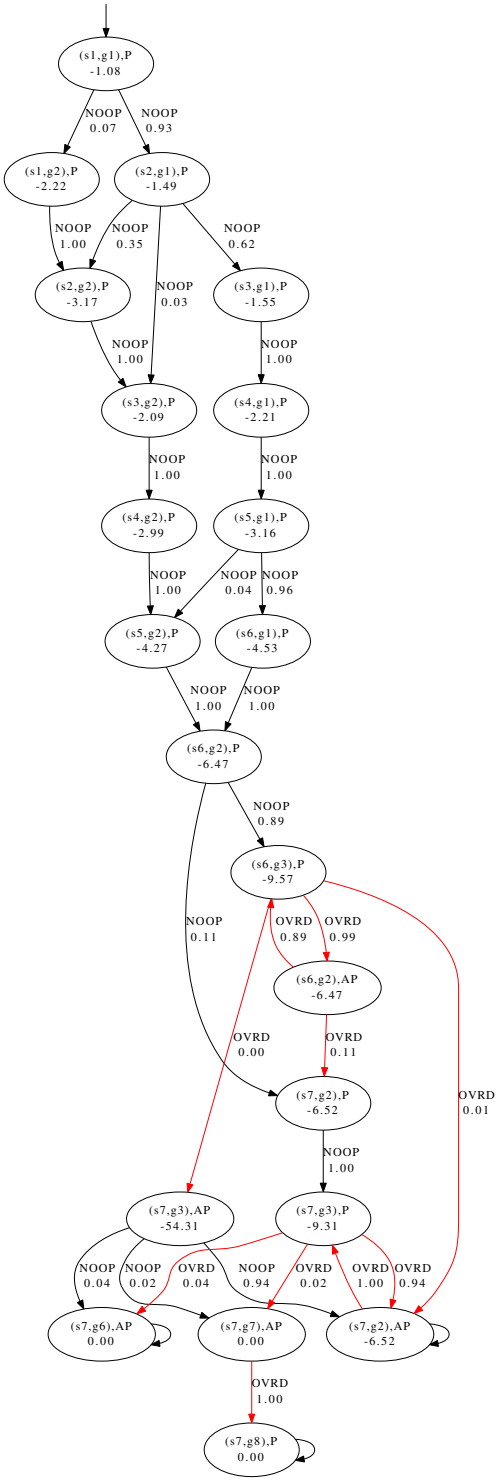

Figure 4. MDP with constraints

Fig 5 illustrates three takeoff scenarios with tail strike risk. We note here that the pilot is modeled as a human pilot transfer function[18] 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 momentary flattening of the 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 III using the unconstrained MDP (see Fig 3). Here, as illustrated previously (see Fig 3) , 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. Subsequently, the continued application of the excessive nose up elevator input results in a tail strike. The green lines indicate the aircraft's response to the MDP policy that was constructed using the CMDP approach described in Section V. Here, FSAM reverts control to the pilot only when there is no risk due to tail strikes (see Fig 4). Thus, from the pitch response (in green) it is evident that the aircraft does not encounter a tail strike.

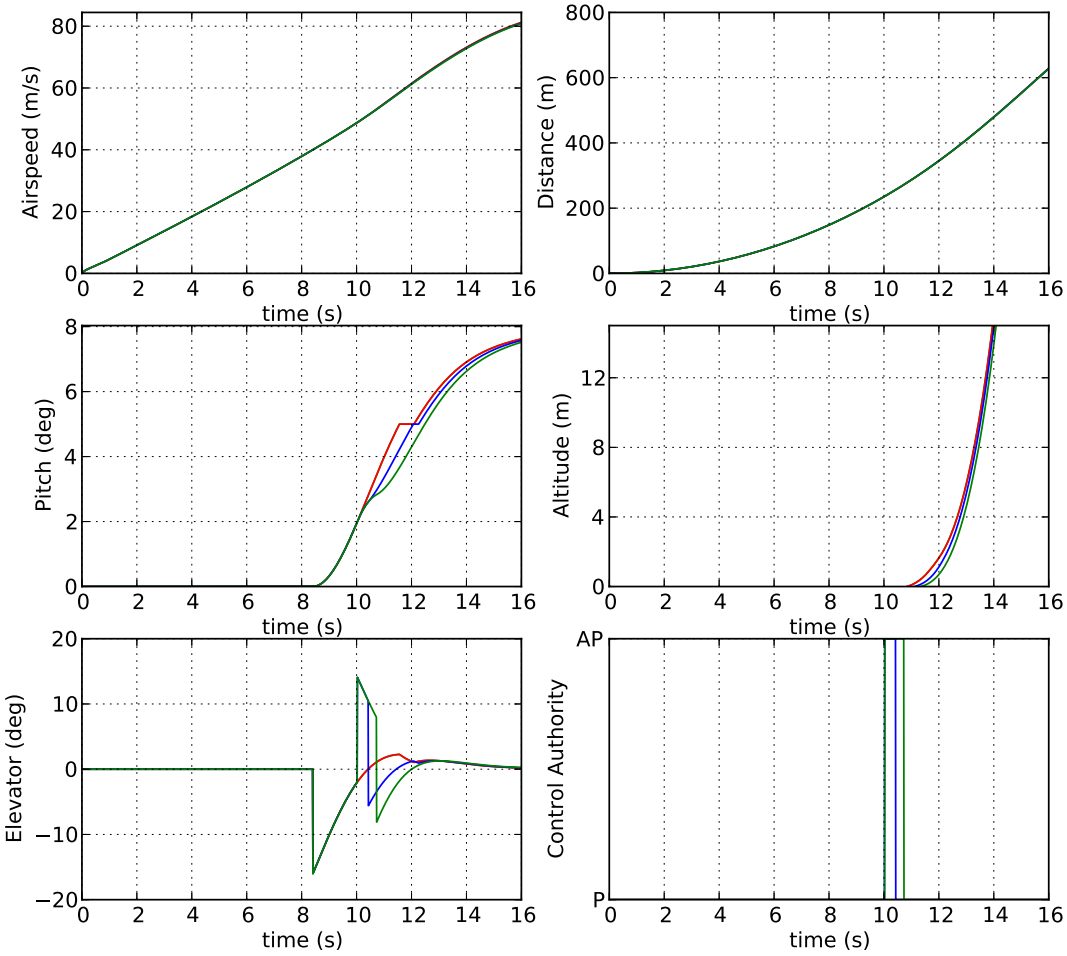

Figure 5. Tail strike scenarios with MDP and CMDP policies. No FSAM augmentation (red), FSAM with MDP policy (blue), FSAM with CMDP policy (green)

VI. Conclusions

FSAM is a decision making aid that can potentially reduce the risk due to LOC. In this paper, we have illustrated the application of a Constrained Markov Decision Process to construct resilient override strategies that ensure that the probability of entering an unsafe state is below an acceptable threshold. We have also illustrated a simple application of the CMDP to prevent LOC risk due to inappropriate rotation procedures that could lead to tail strikes. As a future research direction, we aim to analyze in detail the CMDP framework with additional case studies and results. Properties such as scalability of the CMDP approach and the worst case computation time as a function of state-space and action space size will also be characterized.

Acknowledgement

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

References

[1]C. M. Belcastro, R. L. Newman, D. A. Crider, L. Groff, J. V. Foster, D. H. Klyde, and A. M. Huston. Preliminary analysis of aircraft loss of control accidents: Worst case precursor combinations and temporal sequencing. In *Proc. AIAA Guidance Navigation, and Control Conference*, National Harbor, MD, 2014.

[2]C. M. Belcastro and J. V. Foster. Aircraft loss of control accident analysis. In *Proc. AIAA Guidance Navigation, and Control Conference*, Toronto, Ontario, 2010.

[3]Statistical Summary of Commercial Jet Airplane Accidents. Boeing technical issue, 2013. http://www.boeing.com/news/techissues/pdf/statsum.pdf.

[4]S. Balachandran and E. M. Atkins. Flight Safety Assessment and Management for Takeoff using Determinsitic Moore Machines. *Journal of Aerospace Information Systems (Revision Submitted, Mar, 2015)*.

[5]S. Balachandran and E. M. Atkins. An evaluation of flight safety assessment and management to avoid loss of control during takeoff. In *AIAA Guidance, Navigation and Control Conference*, National Harbor, MD, 2014.

[6]S. Balachandran and E. M. Atkins. Flight safety assessment and management during takeoff. In *AIAA Infotech@Aerospace Conference*, Boston, MA, 2013.

[7]K. McDonough, I. Kolmanovsky, and E. M. Atkins. Recoverable sets of initial conditions and their use for aircraft flight planning after a loss of control event. In *Proc. AIAA Guidance Navigation, and Control Conference*, National Harbor, Maryland, 2014.

[8]M. J. Yu, K. McDonough, D. S. Bernstein, and I. Kolmanovsky. Retrospective cost model refinement for fault signature detection. In *Proc. American Control Conference*, Portland, OR, 2014.

[9]V. Wiels, R. Delmas, D. Doose, P. L. Garoche, J. Cazin, and G. Durrieu. Formal verification of critical aerospace software. *AerospaceLab Journal*, May 2012.

[10]S. Balachandran, N. Ozay, and E. M. Atkins. Verification Guided Refinement of a Flight Safety Assessment and Management System. *(In preparation for submission to the Journal of Aerospace Information Systems - draft available)*.

[11]C. Baier and J. P. Katoen. *Principles of model checking*, volume 26202649. MIT press Cambridge, 2008.

[12]G. Gigante and D. Pascarella. Formal methods in avionic software certification: the do-178c perspective. In *Leveraging Applications of Formal Methods, Verification and Validation. Applications and Case Studies*, pages 205–215. Springer, 2012.

[13]C. M. Holloway. Making the implicit explicit: Towards an assurance case for do-178c. 2013.

[14]E. Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

[15]S. J Russell and P. Norvig. Artificial intelligence: a modern approachprentice hall. *Englewood cliffs, NJ*, 26, 1995.

[16]M. L. Puterman. *Markov Decision Process: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc, 1994.

[17]J. E. Wilborn and J. V. Foster. Defining commercial transport loss-of-control: A quantitative approach. In *Proc. AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Providence, Rhode Island, 2004.

[18]D. T. McRuer and E. S. Krendel. Mathematical models of human pilot behavior. Technical report, DTIC Document, 1974.

[19]C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 1999.