An Autonomous Override System to Prevent Airborne Loss of Control

Sweewarman Balachandran and **Ella M. Atkins**
Department of Aerospace Engineering
University of Michigan
Ann Arbor, MI 48109
{swee,ematkins}@umich.edu

Abstract

Loss of Control (LOC) is the most common precursor to aircraft accidents. This paper presents a Flight Safety Assessment and Management (FSAM) decision system to reduce in-flight LOC risk. FSAM nominally serves as a monitor to detect conditions that pose LOC risk, automatically activating the appropriate control authority if necessary to prevent LOC and restore a safe operational state. This paper contributes an efficient Markov Decision Process (MDP) formulation for FSAM. The state features capture risk associated with aircraft dynamics, configuration, health, pilot behavior and weather. The reward function trades cost of inaction against the cost of overriding the current control authority. A sparse sampling algorithm obtains a near-optimal solution for the MDP online. This approach enables the FSAM MDP to incorporate dynamically changing flight envelope and environment constraints into decision-making. Case studies based on real-world aviation incidents are presented.

Introduction

Fly-by-wire systems have improved aviation safety. The flight control computer reliably maintains a trimmed flight condition, navigates using instruments, and makes required corrections to stay on course. Triply-redundant avionics provide statistical guarantees that avionics and power systems will function reliably. Most flight segments are flown by onboard automation with flight crew monitoring instruments and making high-level decisions. Despite automation aids, Loss Of Control (LOC) accidents still occur. LOC is a condition where an unusual attitude, rate of change of attitude, or aerodynamic state violate normal operating constraints causing deviation outside the *normal flight envelope* (Belcastro and Jacobson 2010). Most LOC accidents are a result of inappropriate pilot inputs, bad weather, and/or onboard system failures (Belcastro and Jacobson 2010).

Envelope protection systems are available today, preventing specific constraint violations including minimum airspeed and maximum bank angle (Traverse 2015). However, modern aircraft accidents typically involve multiple factors such as bad weather and aircraft health issues that result in envelope protection features disengaging leaving the crew to suddenly make difficult decisions in a high-stress, high-workload environment. The likelihood of inappropriate crew inputs increases as stress and workload increase and situational awareness degrades.

Adaptive control (Gregory et al. 2009), system identification (Yu et al. 2014), envelope estimation (McDonough, Kolmanovsky, and Atkins 2014) and path planning (Meuleau et al. 2009) can augment a conventional flight management system with control authorities capable of handling many LOC risk scenarios by adapting to on-board failures, constructing realistic emergency landing plans, and navigating the airplane to a safe landing. Such capabilities have not yet migrated into commercial flight management systems. Challenges in certifying adaptive systems have motivated use of such technologies as a backup to nominal automation rather than a replacement for it.

This paper investigates an efficient formulation of a Flight Safety Assessment and Management (FSAM) capability to rapidly recognize an imminent LOC scenario and automatically switch to an alternate control authority that can handle the high-risk scenario without experiencing a LOC event. FSAM is formulated as an MDP that integrates a novel state, action and reward formulation enabling broad consideration across the suite of factors that might contribute to LOC, an advance over the subsystem-level envelope protection capabilities now available. A sparse sampling MDP solver is applied to enable FSAM to build or modify policies online to account for unanticipated degradations in aircraft flight envelope constraints due to weather (e.g. winds, icing), damage and failures, etc. Accident-motivated case studies are presented to illustrate FSAM operation.

Related Work

Temporal logic planning has contributed to correct-by-construction software for safety-critical decision making (Liu et al. 2013). These techniques facilitate verification, validation and certification. However, the resulting systems are deterministic and difficult to scale making them difficult to extend across all possible anomalous and emergency situations that increase LOC risk. Geometric and search-based methods have been widely used for emergency flight or landing planning (Meuleau et al. 2009). Markov Decision Processes (MDP) or Partially Observable Markov Decision Processes (POMDP) have been used to design alerting systems that could warn the flight crew about imminent collision conflicts with other aircraft and issue conflict resolution advisories (Kochenderfer and Chryssanthacopoulos 2010). These systems always leave decision-making authority with the crew regardless of their responses. Our previous work (Balachandran and Atkins 2015) used a constrained MDP framework to address takeoff related LOC events. However, this formulation did not provide a comprehensive state-space or reward formulation nor did it consider computational complexity tradeoffs.

Several architectures for automation aids that minimize LOC risk have been introduced (Belcastro and Jacobson 2010; Balachandran and Atkins 2014). These architectures integrate various modules to identify and adapt to changes in aircraft dynamics and flight envelopes, dynamically generate safe landing plans, and issue control authority overrides to avoid LOC. The decision making modules in these architectures can benefit from the formulation presented in this paper.

Background

Markov Decision Processes

A Markov Decision Process (MDP) (Puterman 1994) is defined by tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ represents states, $\mathcal{A}$ represents actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ represents transition probabilities, and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ represents a reward function for each state-action pair. Actions are chosen to maximize an expected cumulative discounted reward function

$$\mathcal{V}(s_n) = \mathbb{E}\Big[\sum_{n=0}^{\infty} \gamma^n \mathcal{R}(s_n, a_n)\Big] \tag{1}$$

Here, $a_n \in \mathcal{A}$ is the action selected for current state $s_n \in \mathcal{S}$ and $\gamma \in (0,1]$ is a discount factor. A policy is defined as the mapping $\pi : \mathcal{S} \to \mathcal{A}$. The optimal policy ($\pi^*$) is given by:

$$\pi^* = \arg\max_a \Big\{\mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{P}(s'|s,a)\mathcal{V}^*(s')\Big\} \tag{2}$$

$\mathcal{V}^*(s)$ is the optimal value of state s. The optimal policy can be obtained using algorithms such as value iteration, policy iteration or linear programming (Puterman 1994). Such algorithms explicitly enumerate all MDP states. This can be prohibitively expensive for large state-spaces, so approximate methods to find near-optimal solutions have been developed, including value function approximation, policy search methods and Monte Carlo tree search methods (Sutton and Barto 1998). In this work, we apply a sparsely sampled Monte Carlo tree search algorithm (Kearns, Mansour, and Ng 2002) to construct an approximate solution for the FSAM MDP.

Online Sparse Sampling for Large MDPs

Sparse sampling for large state-space MDPs was originally introduced by Kearns et al. (Kearns, Mansour, and Ng 2002). Given a generative model $\mathcal{G}$ of an MDP[1], the sparse sampling algorithm executes the following steps:

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 1).
3. The estimate of optimal value $\mathcal{V}^*(s)$ is given by:

   $$\hat{\mathcal{V}}^*_H(s) = \max_a \Big\{R(s,a) + \gamma \frac{1}{C} \sum_{s' \in S_a} \hat{\mathcal{V}}^*_{H-1}(s)\Big\} \tag{3}$$

   Note that Eqn (3) computes $\hat{\mathcal{V}}^*_H(s)$ recursively from $\hat{\mathcal{V}}^*_0(s) = 0$.
4. The optimal action is then given by:

   $$\arg\max_a \Big\{\mathcal{R}(s,a) + \gamma \frac{1}{C} \sum_{s' \in S_a} \hat{\mathcal{V}}^*_{H-1}(s)\Big\} \tag{4}$$

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 (Kearns, Mansour, and Ng 2002). Note that this algorithm does not require enumeration of all MDP states, and it can be applied when the MDP state-space is discrete, continuous, or mixed. Computation time can be reduced by independently evaluating each branch at the root node using multi-core processors or GPUs.

FSAM MDP Formulation

State features

The FSAM state must contain all attributes required to appropriately assess LOC risk and in turn make appropriate crew override decisions. Each state s of the MDP formulation is represented by the composition of four main features:

$$s = [F_1, F_2, F_3, F_4] \tag{5}$$

F_1, F_2, F_3, F_4 describe aircraft dynamics and controls, aircraft health, flight crew characteristics and environmental characteristics respectively. This diverse set of state features has never before been integrated for manned or unmanned flight control authority decision making.

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

$$\begin{aligned} F_1 &= [F_{11}, F_{12}, F_{13}, F_{14}] \\ F_{11} &= [u, v, w, p, q, r, \phi, \theta, \psi, x, y, h] \\ F_{12} &= [\delta_e, \delta_a, \delta_r, \delta_t] \\ F_{13} &= [c_g, c_f, c_p] \\ F_{14} &= [M, N, T] \end{aligned} \tag{6}$$

Here F_{11} describes traditional aircraft *physical* state (Stevens and Lewis 2003). 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, h 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

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

landing gears (c_g). F_{14} specifies current control mode M, the number of override directives previously issued N, and time elapsed in the current control mode T. F_{11} and F_{12} contain continuous-valued variables, F_{13} takes discrete values. M, N in F_{14} are discrete, and T is continuous. All F_1 parameters are observable from onboard sensors.

Aircraft health state F_2 describes the health status of various subsystems onboard the aircraft:

$$F_2 = [f_{sys}, h_{eng}, h_{act}, h_{sys}] \tag{7}$$

$f_{sys} = (f_{s_1}, \ldots, f_{s_n})$ are flags that denote the on/off status of various flight systems with potential to influence LOC risk. $h_{eng} = (h_{e_1}, \ldots, h_{e_n})$ 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, anti-icing. All F_2 features are discrete and observable from sensor and health monitoring subsystems.

Pilot characteristics Flight crew state can play an important factor in an FSAM override decision. A simple pilot state abstraction F_3 is proposed:

$$F_3 = [F_{31}, F_{32}, F_{33}] \tag{8}$$

$F_{31} \subset \{CP, FO\}$ 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 paper does not claim progress in translating sensor observations to "human state estimates", research has shown that such observers are feasible (Busso et al. 2004).

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

$$F_4 = [f_{winds}, f_{visibility}, f_{temp}, f_{precip}] \tag{9}$$

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 $f_{precip} \in \{none, rain, snow, hail, storm\}$ denotes precipitation. $f_{wind}, f_{visibility}, f_{temp}$ are continuous while f_{precip} is discrete in this formulation. Wind, visibility and temperature can be estimated from onboard sensors including weather radar plus meteorological reports.

Actions

FSAM is a high-level supervisory module that passively monitors MDP state for elevated LOC risk. If the flight crew mitigates the LOC risk, FSAM continues to passively monitor. FSAM only overrides when LOC risk is substantial and switching to an alternate control authority statistically ensures LOC prevention or recovery. The FSAM MDP therefore has two fundamental actions, i.e. NOOP (no operation) and TOGL (toggle). FSAM remains passive and continues to monitor if NOOP is chosen. TOGL is chosen to switch to a different control authority. Note that there could be multiple toggle actions if there are many control authorities.

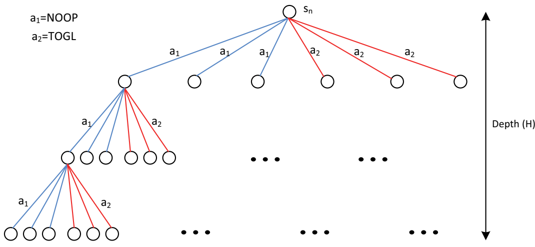

Figure 1: Sparse sampled look-ahead tree with two actions and a branching factor of three

$$a \in \{NOOP, TOGL\} \tag{10}$$

Reward formulation

The FSAM reward function is a "cost" (negative reward) function that penalizes unsafe aircraft states but discourages routine override directives. A weighted additive reward function is proposed for FSAM:

$$\mathcal{R} = \sum_{i=0}^{n} w_i \mathcal{R}_i \tag{11}$$

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]

This paper proposes ten reward/cost terms. $\mathcal{R}_1$ 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.

$$\mathcal{R}_1 = \begin{cases} -1 & if\ (V \leq V_{min}) \vee (V \geq V_{max}) \\ 0 & otherwise \end{cases} \tag{12}$$

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

$$\mathcal{R}_2 = \begin{cases} -1 & if\ (\phi \leq \phi_{min}) \vee (\phi \geq \phi_{max}) \\ 0 & otherwise \end{cases} \tag{13}$$

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

$$\mathcal{R}_3 = \begin{cases} -1 & if\ (h \leq h_{min}) \vee (h \geq h_{max}) \\ 0 & otherwise \end{cases} \tag{14}$$

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

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

$$\mathcal{R}_4 = \begin{cases} -1 & if\ ||X - X_0|| > 0 \\ 0 & otherwise \end{cases} \tag{15}$$

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

The following terms penalize unsafe *pilot* states. $\mathcal{R}_5$ penalizes the absence of the required crew in the cockpit:

$$\mathcal{R}_5 = \begin{cases} -1 & if\ (M = P) \wedge (F_{31} = \{\emptyset\}) \\ -c_1 & if\ (M = P) \wedge (F_{31} = \{CP\} \vee \{FO\}) \\ 0 & otherwise \end{cases} \tag{16}$$

where $0 < c_1 < 1$. $\mathcal{R}_6$ penalizes states in which the pilot may not be capable of flying the aircraft, thus encouraging the selection of a control authority that can maintain appropriate control of the aircraft.

$$\mathcal{R}_6 = \begin{cases} -1 & if\ (M = P) \wedge (F_{32} = unconscious) \\ -c_2 & if\ (M = P) \wedge (F_{32} = fatigued) \\ 0 & otherwise \end{cases} \tag{17}$$

where $0 < c_2 < 1$. $\mathcal{R}_7$ penalizes unusual cockpit activity.

$$\mathcal{R}_7 = \begin{cases} -1 & if\ (M = P) \wedge (F_{33} = abnormal) \\ 0 & otherwise \end{cases} \tag{18}$$

To ensure override actions are not issued by FSAM unnecessarily, $\mathcal{R}_8$ imposes a penalty on choosing an override action. $\mathcal{R}_9$ prevents repeated switching between control authorities by imposing a penalty for an override that is inversely proportional to the duration since the last override. $\mathcal{R}_{10}$ penalizes the total number of overrides.

$$\mathcal{R}_8 = \begin{cases} -1\ if\ (M = P) \wedge (a \neq NOOP) \\ \qquad\qquad \wedge (F_{32}, F_{33} = nominal) \\ 0\ otherwise \end{cases} \tag{19}$$

$$\mathcal{R}_9 = \begin{cases} -\frac{1}{T} & if (T > 0) \wedge (a = TOGL) \\ 0 & otherwise \end{cases} \tag{20}$$

$$\mathcal{R}_{10} = -N_{TOGL} \tag{21}$$

Generative model

The generative model is a function that takes as inputs the current state s^n, action a^n and outputs the reward $\mathcal{R}(s^n, a^n)$ and the next state s^{n+1} chosen according to the 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:

$$\mathcal{P}(s^{n+1}|s^n, a^n) = $$
$$\mathcal{P}(F_1^{n+1}, F_2^{n+1}, F_3^{n+1}, F_4^{n+1}|F_1^n, F_2^n, F_3^n, F_4^n, a^n) \tag{22}$$

The conditional independence among the state features can be exploited to simplify Eqn (22):

$$\begin{aligned} \mathcal{P}(s^{n+1}|s^n, a^n) = \mathcal{P}_1(F_1^{n+1}|F_1^n, F_2^n, F_3^n, F_4^n, a^n) \times \\ \mathcal{P}_2(F_2^{n+1}|F_1^n, F_2^n, F_3^n, F_4^n) \times \\ \mathcal{P}_3(F_3^{n+1}|F_1^n, F_2^n, F_3^n, F_4^n) \times \\ \mathcal{P}_4(F_4^{n+1}|F_4^n) \end{aligned} \tag{23}$$

Note that the features at future time steps $(n + 1)$ depend only on the current features (n) due to the Markov assumption. Here P_1 represents the transition dynamics of the aircraft states and is modeled with aircraft equations of motion (Stevens and Lewis 2003). P_2, P_3 and P_4 represent the transition probabilities associated with the health, pilot characteristics and environment. Models representing these distributions may be constructed from flight data, weather reports, observations of pilot behaviors etc. The environmental feature F_4 is independent of other state features. Each feature F in Eqn (23) is composed of sub-features and hence the terms in Eqn (23) can be further expanded by considering the conditional independence relations among the sub-features.

Case Study

This section illustrates example policy construction for the special case where F_2, F_3 and F_4 remain constant, i.e. all systems and crew function nominally, the weather is clear, and winds are calm, such that distributions P_2, P_3 and P_4 need not be modeled. P_1 is simplified as follows:

$$\begin{aligned} \mathcal{P}_{11}(F_{11}^{n+1}|F_{11}^n, F_{12}^n, F_{13}^n)\mathcal{P}_{12}(F_{12}^{n+1}|F_{11}^{n+1}, F_{12}^n, F_{13}^n, a^n) \times \\ \mathcal{P}_{13}(F_{13}^{n+1}|F_{13}^n, F_{14}^n)\mathcal{P}_{14}(F_{14}^{n+1}|F_{14}^n, a^n) \end{aligned} \tag{24}$$

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

$$X_{n+1} = X_n + \mathcal{F}(X_n, U_n)\Delta t + W_n \tag{25}$$

where $\mathcal{F}$ represents the equations of motion. A Twin-Otter aircraft model (Grauer and Morelli 2014) is used in Eqn (25). $X = F_{11}$ is aircraft physical state, $U = F_{12}$ represents physical control inputs and W is a state disturbance vector with Gaussian noise. Δt is the discretization time.

$\mathcal{P}_{12}$ 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 (McRuer and Krendel 1974) with parameters chosen according to a user-specified distribution. The envelope-aware controller is modelled as a Linear Quadratic Regulator (LQR) control law (Kirk 1970) 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, gears 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 terms used are $\mathcal{R}_{i=1,2,3,8,9,10}$. The weights w_i on these reward terms are $1000, 1000, 1000, 10, 5, 0.8$ respectively. The discount factor γ is 0.7. 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

Table 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 ($15s$)	$3.460s$	$> 25s$	$> 25s$

Table 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 ($55s$)	$1.272s$	$12.672s$	$> 25s$

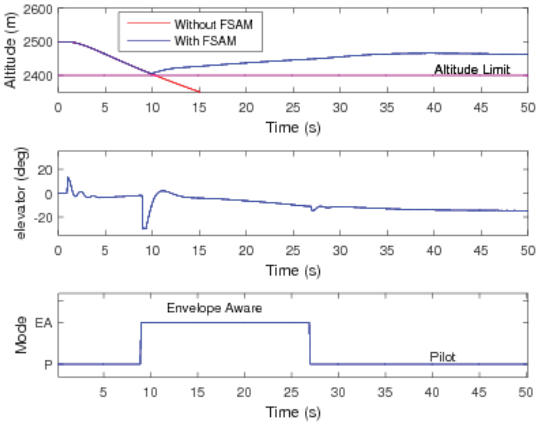

Figure 2: Altitude recovery

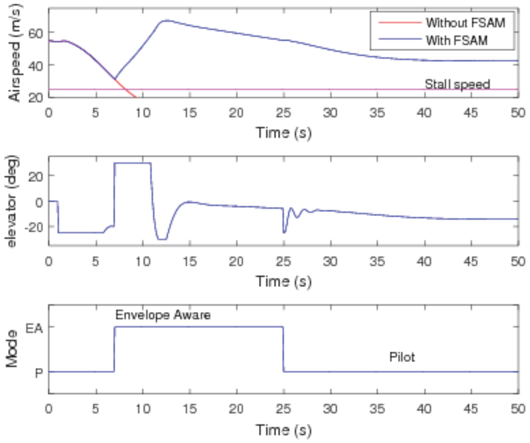

Figure 3: Stall recovery

(Kearns, Mansour, and Ng 2002). Parameter values and their effects on computation time are shown in Tables 1 and 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 2 indicates the aircraft's response without 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 therefore the policy would not override the pilot unless collision with surrounding terrain was imminent.

Policy behavior can be changed by tuning Eqn (11) weights. For example, varying the weight on $\mathcal{R}_9$ in Eqn (11) controls the duration the envelope-aware controller stays active after overriding the pilot. Similarly, the weight on $\mathcal{R}_{10}$ 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 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 (Chaslot, Winands, and van Den Herik 2008)). In this work, the computations were distributed across 8 cores. Table 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 (25). 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 1 shows results for a fixed decision epoch $\Delta T = 1s$. As expected, computation time decreases as the complexity of generative model decreases (i.e. Δt approaches ΔT). 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 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.

Conclusions and Discussion

This paper contributes a 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 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 a phase of flight (takeoff, climb, cruise, descent, landing). State feature time scale separation can also be exploited to decompose the MDP into several simpler MDPs. Control authority switching might be specified with finite state machines but manually generating state machines can be cumbersome and error-prone.

This paper's 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 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.

References

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

Balachandran, S., and Atkins, E. M. 2015. A constrained markov decision process framework for flight safety assessment and management. In *AIAA Infotech@Aerospace Conference*.

Belcastro, C. M., and Jacobson, S. R. 2010. Future integrated system concepts for preventing aircraft loss-of-control accidents. In *Proc. of AIAA Guidance Navigation, and Control Conference*.

Busso, C.; Deng, Z.; Yildirim, S.; Bulut, M.; Lee, C. M.; Kazemzadeh, A.; Lee, S.; Neumann, U.; and Narayanan, S. 2004. Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proceedings of the 6th international conference on Multimodal interfaces*, 205–211. ACM.

Chaslot, G. M.-B.; Winands, M. H.; and van Den Herik, H. J. 2008. Parallel monte-carlo tree search. In *Computers and Games*. Springer. 60–71.

Grauer, J. A., and Morelli, E. A. 2014. A generic nonlinear aerodynamic model for aircraft. In *Proc. AIAA Atmospheric Flight Mechanics Conference*.

Gregory, I. M.; Cao, C.; Xargay, E.; Hovakimyan, N.; and Zou, X. 2009. L1 adaptive control design for nasa airstar flight test vehicle. In *Proc. of AIAA Guidance, Navigation, and Control Conference*.

Kearns, M.; Mansour, Y.; and Ng, A. Y. 2002. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine Learning* 49(2-3):193–208.

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

Kochenderfer, M. J., and Chryssanthacopoulos, J. P. 2010. A decision-theoretic approach to developing robust collision avoidance logic. In *Annual conference on Intelligent Transportation Systems*. IEEE.

Liu, J.; Ozay, N.; Topcu, U.; and Murray, R. M. 2013. Synthesis of reactive switching protocols from temporal logic specifications. *IEEE Transactions on Automatic Control* 1771 – 1785.

McDonough, K.; Kolmanovsky, I.; and Atkins, E. M. 2014. 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*.

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

Meuleau, N.; Plaunt, C.; Smith, D. E.; and Smith, T. B. 2009. An emergency landing planner for damaged aircraft. In *IAAI*.

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

Stevens, L. B., and Lewis, L. F. 2003. *Aircraft control and Simulation*. Hoboken, NJ: Wiley. chapter 1-2, 1–137.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Traverse, P. 2015. *The Avionics Handbook*. CRC press LLC. Chapter 31, Airbus Electrical Flight Controls.

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