2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Reachability-based abstraction for an aircraft landing under shared control

Nikolai Matni, Meeko Oishi

Abstract— **We extend techniques for a reachability-based abstraction to hybrid systems under shared control with application to pilot-autopilot interaction during an aircraft landing. A simple hybrid model of longitudinal aircraft dynamics and mode-logic is developed based on publicly available data. As the pilot and autopilot share control over some of the same inputs, it is possible for the pilot to "fight" the autopilot. New types of safety are proposed to identify regions in the state-space in which pilot-autopilot conflict can occur, and then computed using level set methods. The results partition the state-space into different levels of safety. Cells of this partition form discrete modes in an abstraction of the reachability result, which can inform the design of a pilot display. Our results show how shared control contributed to violations of "safe" pilot interaction with the automation in the Nagoya 1994 A300 accident.**

I. Introduction

Modern civil jet aircraft have many layers of automation, usually with complex mode structure corresponding to different maneuvers (e.g., level flight, constant-rate descent, constant-rate turn), different types of control protections (saturation, envelope protection), and different types of pilot input (manual, shared, supervisory). Flight management systems are designed mostly through ad-hoc rules to avoid known problems, then extensively tested through costly simulations to help identify unanticipated problems. However, as it is physically impossible to test all possible initial conditions and inputs, problems in human-automation interaction can still occur [1], [2], [3]. "Automation surprises," in which the pilot becomes confused about the current mode or cannot anticipate the next mode in the automation [4], [5], and other problematic behaviors in actual aircraft operation, have been implicated in aircraft incidents and accidents.

Computational techniques for verification can create a new level of confidence and reliability in safety-critical systems such as aircraft autopilots, by predicting where failures might occur, and how human operators can predict them [6], [7], [8], [9]. Verification provides a mathematical guarantee of safety, where safety is defined as the ability to remain within a desired set in the state-space, despite bounded control authority. Through standard reachability analysis and controller synthesis, we can compute the subset of the desired set (e.g., an aerodynamic flight envelope) in which we can guarantee the system can always remain: this is the reachable set, which is synonymous with the "safe" region of operation [10], [11]. We draw on the Hamilton-Jacobi techniques implemented in the Level Set Toolbox [12] because of their subgrid accuracy and success in previous aircraft applications, although other techniques [13], [14], [15], [16], [17] could be used.

Verification of *semi-automated systems* introduces further complexity because it involves not only the automation, but also the way in which the user interacts with the automation [18]. The user-interface both provides information to the user about the underlying automation, and allows the user to issue input commands to the system. Formal methods have been used to verify user-interfaces modeled as discrete event systems [7], [8], [19], [20], [21]. Estimation has been used to anticipate the human's actions [22] through particle filters. We consider hybrid systems with which a human interacts – this includes not only the discrete mode-logic, but also the continuous dynamics arising from forces acting on the physical system. Since we cannot guarantee what actions the human will take, we focus on guarantees that the correct information has been provided to the human, in order to achieve a desired task. While *how* this information is displayed is vitally important to effective human-automation interaction, we restrict ourselves to the portion of this problem we can quantify: *what* information is displayed.

In previous work, we used an abstraction method based on a reachability computation, to determine the relevant information content for a pilot's display during an aborted landing of a large civil jet aircraft [23], [24]. One of our key assumptions was that the pilot's input was limited to pre-planned, discrete, supervisory actions. This simple structure, in which the automation controlled the continuous dynamics within each mode, and the pilot simply selected modes as appropriate, allowed us to take advantage of the well-developed verification algorithms for standard, fully-automated hybrid systems.

In this paper, we extend these techniques to a semi-automated system under *shared control*, in which both the automation and the human have control over the same continuous input. Shared control scenarios are common in aircraft "manual" modes, in which the pilot's input is filtered by the automation's algorithms for saturation or envelope-protection. While treating the human's input as a continuous disturbance will ensure safety of the automated system, the reachability results may be far too conservative to be useful: in the aborted go-around scenario [23], this analysis would have recommended that the aircraft never land, in order to ensure a safe go-around! This paper proposes a model

This work was supported by M. Oishi's NSERC Discovery Grant and N. Matni's NSERC USRA.

N. Matni is a student with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada matni@interchange.ubc.ca.

M. Oishi (corresponding author) is an Assistant Professor with the Department of Electrical and Computer Engineering, Vancouver, BC, Canada moishi@ece.ubc.ca.

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

with shared continuous input, which still allows for useful verification results. By completing multiple reachability calculations, each under different assumptions about the pilot's input, we can examine guarantees of safety under shared control by broadly capturing pilot intent.

Our motivation arises from the 1994 Nagoya Airbus A300 incident [25]. During a manual approach to landing, an automatic go-around maneuver was accidentally triggered. While a landing maneuver requires slow descent, a go-around maneuver requires the aircraft to gain speed and ascend rapidly. Unaware that they had initiated a go-around maneuver, the flight crew applied a pitch-down command to make the aircraft descend, while the automation applied a pitch-up command to make the aircraft ascend. The flight crew's input commands overrode the automation's input commands to the elevators, but not to the trimmable horizontal stabilizers (THS). This resulted in elevators positioned to drive the aircraft into a nose-down orientation for descent, and the THS positioned to drive the aircraft into a nose-up orientation for ascent. The aircraft was physically unable to follow the descending glideslope, so the flight crew decided to abandon the landing. They returned to manual mode, and attempted a manual go-around maneuver, sending a pitch-up command to the elevators. This command, combined with the THS oriented to achieve a full nose-up position, caused the aircraft to climb too quickly and to stall. An unanticipated combination of mode-logic and aircraft dynamics under shared control led to a scenario in which the pilots inadvertently "fought" the aircraft automation.

We aim to identify such scenarios before implementation and operation of shared control systems. The main contributions of this paper are: 1) an example of human-automation interaction under shared control, 2) a novel application of reachability tools for verification under shared control, and 3) an extension of abstraction techniques for a human-automation system under shared control, as opposed to supervisory control. The paper is organized as follows: In Section II, we first introduce the aircraft model, a hybrid system with both longitudinal continuous dynamics as well as realistic mode-logic. Section III describes the use of reachability tools to compute three different levels of safety. Section IV presents the results of an abstraction based on the techniques introduced in [24], but extended to accommodate these new levels of safety. We discuss our results in the context of our previous work, and lastly, Section V provides conclusions and directions for future work.

II. Model Formulation

We model the longitudinal dynamics of the aircraft as a switched system $H = (X, Q, R, \Sigma, f, \delta)$ with continuous state $x \in X$, discrete modes Q, continuous reference inputs $r \in R$, discrete inputs $\sigma \in \Sigma$, continuous dynamics $f_q : X \times R \to X$ indexed by mode $q \in Q$, and transition function $\delta : X \times Q \times R \times \Sigma \to X \times Q$. Figure 1 depicts modes $Q = \{$TRIM ADJ, MAN, GA, ALT, GA/PO, MAN/OOT, PERT ACQ $\}$, discrete inputs $\Sigma = \{\sigma_{\mathrm{MAN}}, \sigma_{\mathrm{P/O}}, \sigma_{\mathrm{TRIM}}, \sigma_{\mathrm{GO}}\}$, the transition function δ, and the initial mode $Q_0 = \{\mathrm{MAN}\}$. The

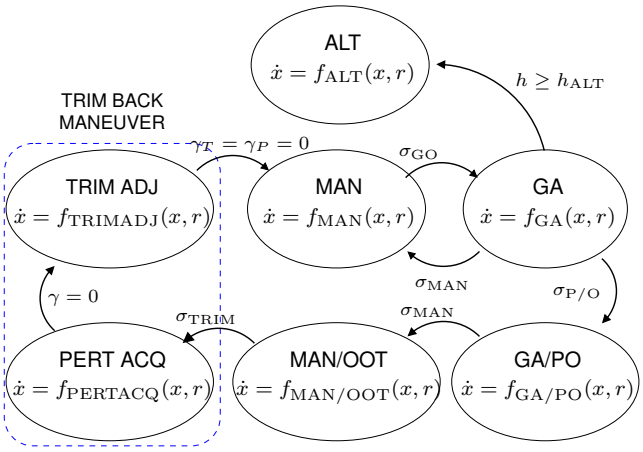

Fig. 1. Hybrid dynamical system model of manual landing scenario.

continuous dynamics $\dot{x} = f_q(x, r)$ are described below. In all modes, the aircraft automation controls low-level dynamics to track a desired reference trajectory. However, in some modes, the reference values are chosen by the pilot, while in other modes, the reference values are chosen by the automation. The mode-logic is based on publicly available data for the A-300, a mid-size civil jet aircraft [25].

In a standard manual landing, the pilot selects MAN mode, and guides the aircraft along a desired descending flight path. To abort the landing, the pilot selects GA mode (σ_{GO}), which causes the aircraft to automatically climb at a constant rate. Normally, after reaching a desired altitude h_{ALT}, the aircraft automatically transitions to level flight. From GA mode, the go-around can be aborted by re-selecting MAN mode (σ_{MAN}). However, if the pilots attempt to physically override the automation ($\sigma_{\mathrm{P/O}}$) and then re-select MAN mode (σ_{MAN}), the aircraft enters a different mode sequence (GA/PO, MAN/OOT). These modes are governed by significantly different control laws than are GA, MAN.

The modes in H differ in how the reference inputs r_P and r_T are achieved, and in the constraints on the reference inputs, as shown in Table I. Modes in Figure 1 fall into two categories: fully automated (GA, ALT, PERT ACQ, TRIM ADJ) and shared control (MAN, MAN/OOT, GA/PO). In fully automated modes, the automation selects the reference inputs. In shared control modes, the user selects the reference input r_P, and the automation selects the reference input r_T as applicable.

A. Aircraft Equations of Motion

The longitudinal dynamics of the aircraft arise from a standard linearization of rigid-body dynamics about the aircraft body frame under quasi-steady flow [26] with small perturbations. Using the short period approximation, the state $x_P = [\alpha, \dot{\theta}, \gamma_P]$ consists of angle of attack α, pitch rate $\dot{\theta}$, and flight path angle γ_P. The equations of motion are obtained by linearizing around the trim condition, $x_T = [0, 0, \gamma_T]$ with aircraft trim speed $u_T = 243.5$ ft/s and trim flight path angle γ_T. The total flight path angle is $\gamma = \gamma_P + \gamma_T$, with γ_P a perturbation from the trim flight path

Mode	Input constraints	Control authority	Reach set	Function
MAN	$r_P \in [-13.3^\circ, 13.3^\circ]$	Shared control	Fig. 5	Manual mode for descent
GA	$r_P = 6.65^\circ$	Fully automated	Fig. 2	Go-around maneuver
ALT	$r_P = 0^\circ$	Fully automated	Fig. 3	Altitude hold
GA/PO	$r_P = -13.3^\circ, r_T = 10.0^\circ$	Shared control	Fig. 6	Go-around with pilot override
MAN/OOT	$r_P \in [-13.3^\circ, 13.3^\circ]$	Shared control	Fig. 5	Manual out-of-trim mode
TRIM ADJ	$r_T = 0^\circ, r_P = -\gamma_T$	Fully automated	Fig. 4	Trim adjust (trim back maneuver)
PERT ACQ	$r_T = 0^\circ, r_P = -\frac{1}{2}\beta\gamma_T$	Fully automated	Fig. 4	Perturbation acquire (trim back maneuver)

TABLE I

AIRCRAFT LANDING MODES, REFERENCE INPUT BOUNDS, CONTROL AUTHORITY, AND CORRESPONDING REACHABLE SET.

angle γ_T. The reference input $r = [r_P, r_T] \in \mathbb{R}^2$ consists of the reference flight path angle for γ_P and γ_T, respectively. We extend the state-space to $x = [x_P, \gamma_T] \in \mathbb{R}^4$. The input δ_e due to elevator deflection affects the dynamics of x_P, while the THS affects the dynamics of γ_T. We describe the synthesis of these controllers in terms of the reference flight path angles r_P, r_T below.

The open-loop dynamics are given by $\dot{x} = A_{ol}(\gamma_T)x + B_{ol}\delta_e$, with

$$\begin{aligned} A_{ol}(\gamma_T) &= \begin{bmatrix} -0.6277 & 1 & -0.1322\sin(\gamma_T) \\ -1.9552 & -1.0524 & 0.0344\sin(\gamma_T) \\ 0.6277 & 0 & 0.1322\sin(\gamma_T) \end{bmatrix} \\ B_{ol} &= \begin{bmatrix} -0.0418 & -1.3391 & 0.0418 \end{bmatrix}^T \end{aligned} \tag{1}$$

with numerical values for stability derivatives taken from data for the DC-8, a mid-size civil jet aircraft, on approach to landing [26] (App. A, Table A-5).

In most modes, the trim dynamics are $\dot{\gamma}_T = 0$. However, under unusual circumstances that require re-trimming (described in Section II-B), the THS effectively moves the trim conditions gradually from one set-point to another. Different controllers are used in each mode.

B. Aircraft Mode-Logic

In Altitude (ALT) and Go-Around (GA), $\dot{\gamma}_T = 0$ and r_T and r_P are constants, as described in Table I. In Manual (MAN) mode, $\dot{\gamma}_T = 0$, and the pilot chooses the reference input $r_P \in [-13.3^\circ, 13.3^\circ]$. In these modes, a static full-state feedback controller for tracking

$$\delta_e(x, r) = -K(\gamma_T)x_P + Nr_P \tag{2}$$

is chosen with $K(\gamma_T) \in \mathbb{R}^{1\times 3}$ such that the closed-loop systems $f_{\mathrm{GA}}(x, r) = f_{\mathrm{MAN}}(x, r) = f_{\mathrm{ALT}}(x, r)$,

$$f_{\mathrm{ALT}}(x, r) = \begin{bmatrix} A_{cl}x_P + B_{cl}r_P \\ 0 \end{bmatrix} \tag{3}$$

with $A_{cl} = A_{ol}(\gamma_T) - B_{ol}K(\gamma_T)$,

$$\begin{aligned} A_{cl} &= \begin{bmatrix} -0.6486 & 0.9376 & -0.0963 \\ -2.6226 & -3.0477 & -3.0803 \\ 0.6486 & 0.0624 & 0.0963 \end{bmatrix} \\ B_{cl} &= -2.3B_{ol} \end{aligned} \tag{4}$$

have eigenvalues at $-1.2, -1.2 \pm 0.12j$.

The Go-Around/Pitch Override (GA/PO) mode, in which $\dot{\gamma}_T \neq 0$, occurs when the pilot attempts to manually override the pitch during a GA maneuver. The pilot has control of the elevator and therefore r_P, while the automation retains control of the THS and therefore r_T. We assume that $r_P < 0$ and $r_T > 0$ since the pilot is attempting to land the aircraft, and the autopilot is attempting to ascend. With the control law (2) implemented, the dynamics

$$f_{\mathrm{GA/PO}}(x, r) = \begin{bmatrix} A_{cl}x_P + B_{cl}(r_P - \gamma_T) \\ -\beta\mathrm{sgn}(\gamma_T - (r_T - \gamma_P)) \end{bmatrix} \tag{5}$$

in x_P and γ_T are coupled, and reflect the fact that the pilot and the autopilot are "fighting" such that $\gamma \to r_P$ and $\gamma \to r_T$, respectively. In (5), $\beta = 0.5^\circ/\mathrm{sec}$, and sgn represents the signum function, with $\mathrm{sgn}(0) = 0$.

In Manual Out of Trim (MAN/OOT), $\dot{\gamma}_T = 0$ but $\gamma_T \neq 0$. This mode occurs when the pilot initiates manual operation from GA/PO. With the controller as in (2), the dynamics are

$$f_{\mathrm{MAN/OOT}}(x, r) = f_{\mathrm{MAN}}(x, r) \tag{6}$$

as in (3). However, note that the bounds on r_P are not adjusted to reflect the non-zero γ_T (Table I). It is therefore possible for the pilot to chose r_P such that γ becomes unsafe, as appears to have occurred in the description of the Nagoya 1994 accident.

Perturbation Acquire (PERT ACQ) mode, in which $\dot{\gamma}_T = 0$ and $r_P = -\gamma_T$, is the first part of the trim-back maneuver, which drives γ to 0. With control as in (2), the dynamics are

$$f_{\mathrm{PERTACQ}}(x, r) = \begin{bmatrix} A_{cl}x_P + B_{cl}(-\gamma_T) \\ 0 \end{bmatrix} \tag{7}$$

In TRIM ADJ mode, the second part of the trim back maneuver, we have $\dot{\gamma}_T \neq 0$, and $r_T = 0$, $r_P = -\frac{1}{2}\beta\gamma_T$. With control as in (2), the dynamics are

$$f_{\mathrm{TRIMADJ}}(x, r) = \begin{bmatrix} A_{cl}x_P + B_{cl}(-\frac{1}{2}\beta\gamma_T) \\ -\beta\mathrm{sgn}(\gamma_T) \end{bmatrix} \tag{8}$$

with $\beta/2$ chosen to regulate the transient response.

III. SAFETY UNDER SHARED CONTROL

An algorithm has been recently developed to create a discrete event system (DES) abstraction of a hybrid system with discrete user-initiated inputs [24]. The algorithm involves three steps: 1) separation of the hybrid system into subsystems which contain no human-initiated discrete inputs, 2) calculation of the reachable set for each subsystem, and

3) abstraction to a DES based on the reachability result. The reachability result is used to partition the state-space into intersections of "safe" or "unsafe" regions in each subsystem. Since we cannot predict when or even if the pilot will initiate a discrete event, we complete reachability analysis only over those portions of the system which are fully automated. The intersections of reachable sets reveal the effect of discrete pilot inputs (e.g., transitioning the system from a safe region in one subsystem to an unsafe region in the next subsystem) without making unrealistic assumptions about the pilot (e.g., by treating pilot-initiated events as either standard controlled inputs or disturbance inputs).

We use the same algorithm here, with modifications that incorporate the effect of shared control on the reachability analysis on each subsystem (described in Sections III-A and III-B). Our system (Figure 1) contains both discrete pilot inputs and *continuous* pilot inputs (in modes MAN, GA/PO, MAN/OOT). We therefore separate the system into six subsystems which contain no pilot-initiated events (since there are 6 modes which occur *after* a discrete pilot-initiated input), containing the modes {GA}, {GA/PO}, {MAN-OOT}, {PERT ACQ, TRIM ADJ, MAN}, {MAN}, {GA, ALT}. Hence all transitions within the six subsystems are state-based transitions, and each subsystem is amenable to standard reachability analysis.

Computing the reachable set involves representing all the states which have a path to a target set. We draw on level set methods [12] here because of their subgrid accuracy and success in previous aircraft applications [23], [27], however other techniques can be used [16], [17], [15], [13], [14]. For a continuous system, the reachable set $\mathcal{W}(t)$ is computed in backwards time: starting with the desired target, which is encoded implicitly as a level set function, the boundary of the target set is propagated backwards in time according to the system dynamics $\dot{x} = f(x, r)$. Define a continuous function $J_0 : X \to \mathbb{R}$ which encodes the target

$$\mathcal{W}_0 = \{x \in X \mid J_0(x) \geq 0\}. \tag{9}$$

Finding the backwards reachable set $\mathcal{W}(t)$ requires solving the terminal value time-dependent modified Hamilton-Jacobi (HJ) partial differential equation (PDE) [11]

$$\begin{array}{rcll} \frac{\partial J(x,t)}{\partial t} + \min\left[0, H\left(x, \frac{\partial J(x,t)}{\partial x}\right)\right] & = & 0 & \text{for } t < 0 \\ J(x,0) & = & J_0(x) & \text{for } t = 0 \end{array} \tag{10}$$

with

$$H\left(x, \frac{\partial J(x,t)}{\partial x}\right) = \max_{r \in R} \frac{\partial J(x,t)}{\partial x}^T f(x, r). \tag{11}$$

As shown in [11], we obtain an implicit representation of the reachable set $\mathcal{W}(t) = \{x \in \mathcal{X} \,|\, J(x,t) \geq 0\}$. The algorithm for hybrid systems is described in [11].

The initial cost function is determined by state constraints (due to the flight envelope) and control constraints (due to feedback under saturation). The aerodynamic envelope is defined by state bounds such that $x_{\min} \leq x \leq x_{\max}$, where $x_{\min} = [-11.5°, -15°, -13.3°, -13.3°]$ and $x_{\max} = -x_{\min}$. Additionally, since the elevators can deflect a maximum of $u_{\max} = 50°$, the feedback control (2) must lie within this range for any reference flight path angle r_P. Bounds on the reference inputs $r = [r_P, r_T]$ for each mode are indicated in Table I. The target consists of states outside of these bounds, as well as states which result in a saturated control input. For each mode $q \in Q$ we define the initial cost function:

$$\begin{array}{rcl} J_0(q,x) & = & \min_x\{J_0^{\text{state}}(q,x), J_0^{\text{sat}}(q,x)\}, \text{ with} \\ J_0^{\text{state}}(q,x) & = & \min_x\{x - x_{\min}, x_{\max} - x\} \\ J_0^{\text{sat}}(q,x) & = & \min_x\{u_{\max} - \max_{r\in\mathcal{R}(q)} \delta_e(x,r), \\ & & \quad \min_{r\in\mathcal{R}(q)} \delta_e(x,r) - u_{\max}\} \end{array} \tag{12}$$

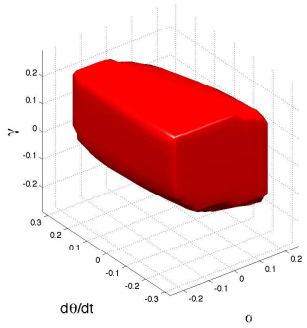

Fig. 2. Safe set $\mathcal{W}_{\text{safe}}^{\text{GA}}$ in $x_P = [\alpha, \dot{\theta}, \gamma]$ of fully automated mode GA; note $\dot{\gamma}_T = 0$. States inside the shaded region are deemed "safe".

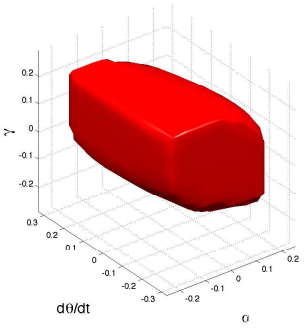

Fig. 3. Safe set $\mathcal{W}_{\text{safe}}^{\text{ALT}}$ in $x_P = [\alpha, \dot{\theta}, \gamma]$ of fully automated mode ALT; note $\dot{\gamma}_T = 0$. States inside the shaded region are deemed "safe".

A. Fully automated modes

For modes which are fully automated in their continuous control input (GA, ALT, PERT ACQ, TRIM ADJ), we compute the reachable set using the Hamiltonian (10). The result of the reachability computation in each mode is the set of states which are guaranteed to remain within their flight envelope and not saturate the elevators. Figures 2 and 3 show the reachable sets for modes GA and ALT. (We assume that all continuous states inside the reachable set in GA will be driven into the reachable set in ALT.) Figure 4 shows the reachable sets for the modes PERT ACQ and TRIM ADJ in subsystem {PERT ACQ, TRIM ADJ, MAN}. Since MAN is a shared control mode, we address it in a separate computation in Section III-B. The computed sets in Figure 4 are 4D, so we plot a series of 3D sets to provide 3D snapshots of the 4D object at specific values of γ_T.

B. Shared control modes

For modes that are not fully automated (MAN, GA/PO, MAN-OOT), the pilot has control over the continuous input. Our strategy to address verification under shared continuous control is to use multiple reachability calculations, each with different assumptions about pilot intent. We have chosen three sets of assumptions here, although in general, this number could be chosen arbitrarily. We choose increasingly looser assumptions about the pilot's behavior to create multiple levels of safety: *safe*, *marginally safe*, and *recoverably*

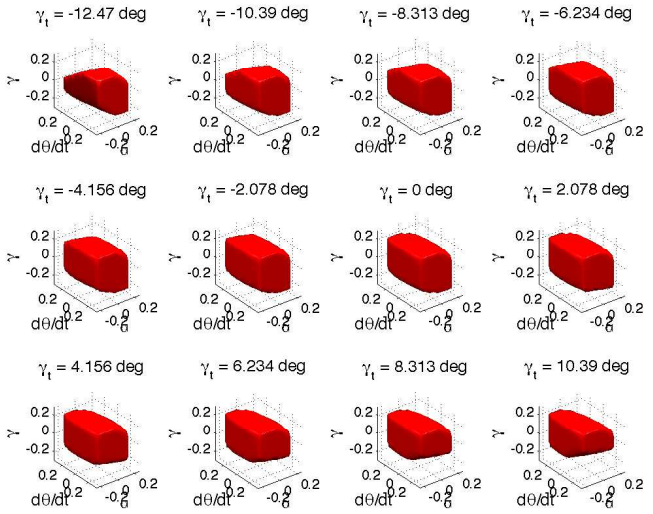

Fig. 4. Safe sets $\mathcal{W}_{\text{safe}}^{\text{TRIMADJ}}$, $\mathcal{W}_{\text{safe}}^{\text{PERTACQ}}$ in $x = [\alpha, \dot{\theta}, \gamma, \gamma_T]$ of fully automated modes TRIM ADJ, PERT ACQ. States inside the shaded region are deemed "safe". Since $\dot{\gamma}_T \neq 0$, we compute the reachable set in 4D, and plot 3D snapshots for various values of γ_T as labeled above.

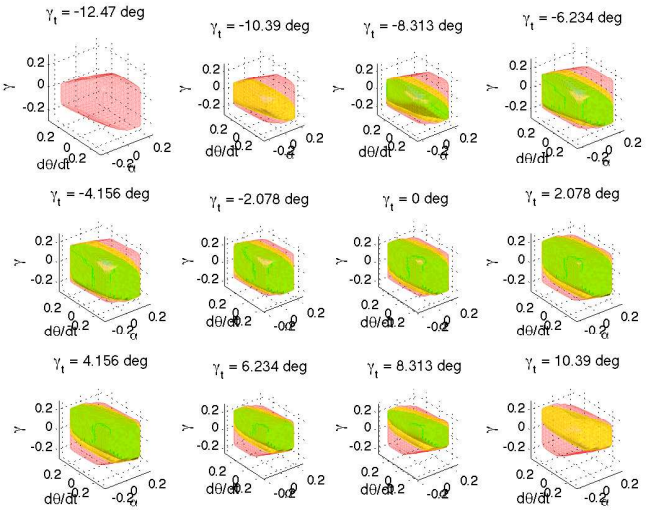

Fig. 5. The green, yellow and red sets represent, respectively, the safe set $\mathcal{W}_{\text{safe}}^{\text{MAN/OOT}}$, the marginally safe set $\mathcal{W}_{\text{marg}}^{\text{MAN/OOT}}$ and the recoverably set $\mathcal{W}_{\text{recov}}^{\text{MAN}}$ of the shared control mode MAN/OOT. The respective sets of MAN, $\mathcal{W}_{\text{safe}}^{\text{MAN}}$, $\mathcal{W}_{\text{marg}}^{\text{MAN}}$, $\mathcal{W}_{\text{recov}}^{\text{MAN}}$, are shown in the 3D snapshots at $\gamma_T = 0$.

safe. To find regions that are safe, marginally safe, and recoverably safe, respectively, we 1) treat the pilot's reference input as a disturbance input, 2) assume the pilot provides no reference input ($r_P = 0$), and 3) treat the pilot's reference input as a controlled input. Note that we do *not* enforce the computed invariance-preserving control law along the boundary of the reachable set in shared control modes.

The Hamiltonians for these calculations, respectively, are

$$\begin{aligned} H\left(x, \frac{\partial J(x,t)}{\partial x}\right) &= \min_{r\in\hat{R}} \frac{\partial J(x,t)}{\partial x}^T f(x,r) \\ H\left(x, \frac{\partial J(x,t)}{\partial x}\right) &= \frac{\partial J(x,t)}{\partial x}^T f(x,0) \\ H\left(x, \frac{\partial J(x,t)}{\partial x}\right) &= \max_{r\in R} \frac{\partial J(x,t)}{\partial x}^T f(x,r) \end{aligned} \tag{13}$$

with $\hat{R} \subset R$. The resulting reachable sets $\mathcal{W}_{\text{safe}}$, $\mathcal{W}_{\text{marg}}$, $\mathcal{W}_{\text{recov}}$ are interpreted as providing corresponding levels of safety: 1) safe, 2) marginally safe, 3) recoverably safe. Note that regions of higher safety levels are always fully contained within regions of lower safety levels.

$$\mathcal{W}_{\text{safe}} \subseteq \mathcal{W}_{\text{marg}} \subseteq \mathcal{W}_{\text{recov}} \tag{14}$$

The reachable set $\mathcal{W}_{\text{safe}}$ is computed under the assumption that the pilot is actively driving the aircraft out of its flight envelope. It represents the worst-case scenario, and therefore provides the most conservative result. Those states in $\mathcal{W}_{\text{safe}}$ are "safe" because for those states the aircraft will remain in its flight envelope regardless of the pilot's continuous input. We assume that the bounds on the reference input r are 25% of the values given in Table I, a reasonable estimate of a pilot's actions under standard operation. Larger bounds will produce more conservative results, by allowing the disturbance input (e.g., the pilot) more control authority.

The reachable set $\mathcal{W}_{\text{marg}}$ is computed under the assumption that the pilot has no active continuous input. Those states in $\mathcal{W}_{\text{marg}}$ are "marginally safe" because for those states the aircraft will remain in its flight envelope assuming that the pilot does not interfere with the aircraft automation.

The reachable set $\mathcal{W}_{\text{recov}}$ is computed under the assumption that the pilot is acting as precisely as any automaton to keep the aircraft within its flight envelope. It represents the best-case scenario, and therefore provides the least conservative result. Those states in $\mathcal{W}_{\text{recov}}$ are "recoverably safe" because for those states the aircraft can remain in its flight envelope as long as the pilot applies the continuous input specifically computed through the above reachability calculation.

We complete this three-part calculation for subsystems {MAN} and {MAN/OOT}. Figure 5 shows $\mathcal{W}_{\text{safe}}$, $\mathcal{W}_{\text{marg}}$, and $\mathcal{W}_{\text{recov}}$, respectively, for MAN/OOT. The result of the three calculations in MAN are equivalent to the snapshot in Figure 5 at $\gamma_T = 0$. Notice that for some values of γ_T in Figure 5, there are no combinations of x_P which are safe or marginally safe.

While GA/PO is a shared control mode, we assume that in the worst case, $r_P = -13.3°$ and $r_T = 10.0°$. With these assumptions, the reachability calculation proceeds as in Section III-A. Figure 6 shows $\mathcal{W}_{\text{safe}}$ for GA/PO.

IV. Reachability-Based Abstraction

As described in [23], [24], the result of the reachability calculations in semi-automated systems can be abstracted to form a discrete event system. This abstraction is particularly useful because it is a simple way of encoding behaviors possible in a semi-automated system under shared control. In addition, information in the abstraction can be used to inform the design of user-interfaces [19], so that the user can be fully informed about the effect of user-initiated events on system safety. We use the algorithm as described in [24], applied to the reachability calculations performed in Section III:

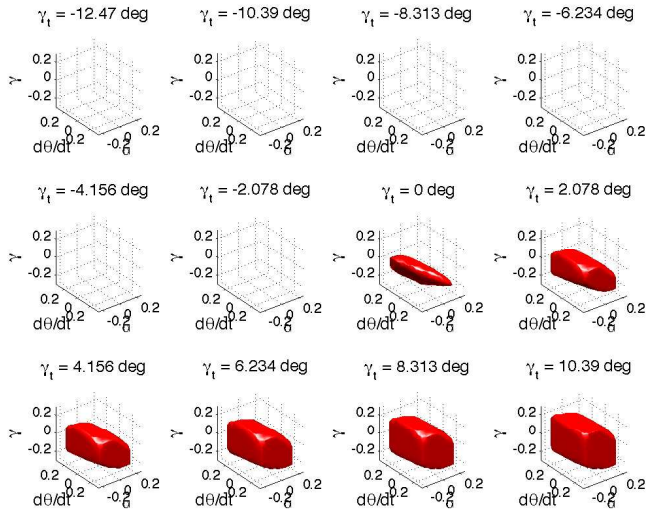

Fig. 6. Safe set $\mathcal{W}_{\text{safe}}^{\text{GA/PO}}$ of shared control GA/PO. Although functionally a shared control mode, we assume in the worst-case scenario that $r_P = -13.3°, r_T = 10°$, and hence pilot/autopilot interaction is predictable (and treated as a fully automated mode).

The state-space is partitioned according to the reachability result, and each cell of the partition is represented by a single discrete mode. The resulting DES is assured to be deterministic through further partitioning the state-space in modes from which the user can initiate a discrete event. Details of the abstraction (and proof of its determinism) are in [24].

However, the reconstruction of a DES occurs in a slightly different manner since we have multiple reachable sets to account for in modes under shared continuous control. In the fully automated modes of Figure 1, we use the standard abstraction of "safe" and "unsafe" sets into separate modes. For shared control modes, the new levels of safety defined in the previous section, and their corresponding sets $(\mathcal{W}_{\text{safe}}, \mathcal{W}_{\text{marg}}, \mathcal{W}_{\text{recov}})$, are used in creating the abstraction. In order for the DES to be deterministic, additional modes must be created when it is possible to transition from a fully automated mode to a shared control mode by a pilot initiated discrete event. For example, the mode corresponding to the "safe" set of GA, $\mathcal{W}_{\text{safe}}^{\text{GA}}$, must be further subdivided into modes representative of the intersection between $\mathcal{W}_{\text{safe}}^{\text{GA}}$ and the three levels of safety of MAN $(\mathcal{W}_{\text{safe}}^{\text{MAN}}, \mathcal{W}_{\text{marg}}^{\text{MAN}}, \mathcal{W}_{\text{recov}}^{\text{MAN}})$, since it is possible to return to MAN by a pilot-initiated input σ_{MAN} at anytime during the GA maneuver.

In addition, we take advantage of the relationship in (14) to reduce the complexity of state-based transitions within a given mode (such as MAN) – e.g., it is only possible to transition to neighboring levels of safety. One advantage of this relationship is that a user interacting with such abstractions is given warning, in effect, that their actions will lead to unsafety. For example, if the aircraft transitions from a safe region to a marginally safe region, the recoverably safe region is essentially a buffer to allow the user to "recover" to a higher level of safety before the aircraft enters the unsafe region. Once the complete DES has been created, modes with empty sets can be removed, as the system will never enter them. Finally, finite state machine reduction is used, and the result is a deterministic, reduced DES, shown in Figure 7.

Our aircraft landing system generates behavior qualitatively similar to descriptions of the Nagoya 1994 accident. We focus in particular on the transition from GA/PO $\overset{\sigma_{\text{MAN}}}{\rightarrow}$ MAN/OOT. The steady state value $x^* = [0°, 0°, -3.3°, 10°]$ lies within the safe set $\mathcal{W}_{\text{safe}}^{\text{GA/PO}}$ in GA/PO mode, but only within the *marginally safe* set $\mathcal{W}_{\text{marg}}^{\text{MAN/OOT}}$ in MAN/OOT mode. When σ_{MAN} is initiated, the system moves from being "safe" to "marginally safe". Once in MAN/OOT, $\gamma_T = 10°$ and assuming the flight crew maintains $r_P = -13.3°$, the aircraft will remain in the marginally safe region $\mathcal{W}_{\text{marg}}^{\text{MAN/OOT}}$. Figure 5 shows that the safe set is empty at this value of γ_T, hence the highest level of safety that is possible at this stage is marginal safety, which is preserved so long as the pilots are effectively hands-off of the elevator controls with $r_P = 0°$.

However, once the flight crew decides to abandon the landing and attempt a manual go-around, the pilots choose r_P to make the aircraft climb in altitude. We assume $r_P = 6.65°$ as in the reachability calculation of Section III. With this value of r_P, the aircraft will first enter the recoverably safe set $\mathcal{W}_{\text{recov}}^{\text{MAN/OOT}}$, then eventually will be driven out of $\mathcal{W}_{\text{recov}}^{\text{MAN/OOT}}$ into a region which is unsafe, and corresponds to a stall condition. Although the flight crew was aware that the aircraft was in an out of trim configuration (i.e. that it was in MAN/OOT mode), they were not aware of the limitations that this imposed on their actions, and were therefore unable to reliably ascertain the safety of their maneuver.

V. Conclusion

We extend techniques for a reachability-based abstraction to a hybrid systems under shared control. We model pilot-autopilot interaction during an aircraft landing as a hybrid system with longitudinal aircraft dynamics. Our model is capable of mimicking a situation in which the pilot "fights" the autopilot. For modes under shared control, three sets corresponding to three levels of safety are computed: 1) safe, 2) marginally safe, and 3) recoverably safe. Using level set techniques, these sets are computed by choosing different Hamiltonians to reflect different roles for the pilot input. We use the result of the reachability computations to develop a DES abstraction by partitioning the state-space. Our initial results are consistent with behavior that was implicated in the Nagoya 1994 Airbus A-300 accident.

We aim to develop a general theory for techniques to identify problems in human-automation interaction early in the design process. In future work, we plan to extend the method presented for the aircraft landing system to generic hybrid systems with all possible combinations of continuous and discrete inputs, controlled by the user or by the automation.

References

[1] C. Billings, *Aviation Automation: The Search for a Human-Centered Approach*. Hillsdale, NJ: Erlbaum, 1997.

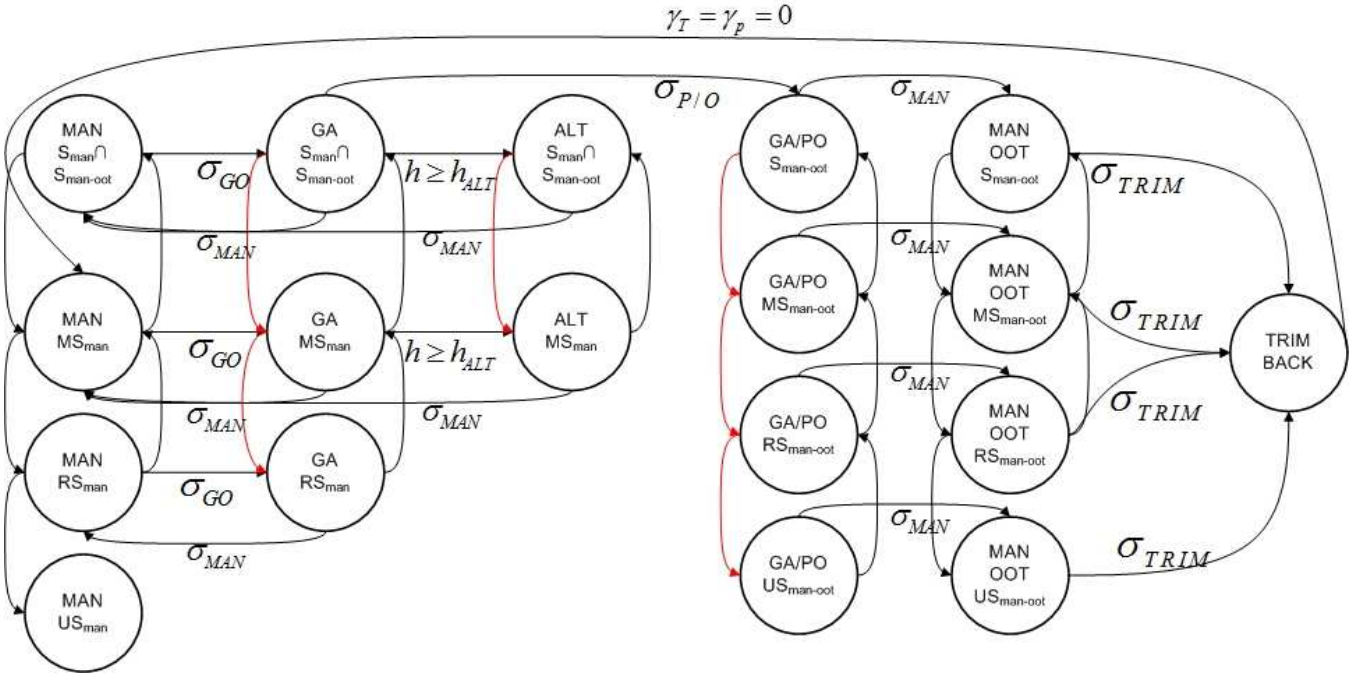

Fig. 7. User interface (reduced DES). For clarity, modes associated with fully automated "unsafe" sets, and modes with empty sets, are omitted. In the diagram the following abbreviations are used: Safe (S), Marginally Safe (MS), Recoverably Safe (RS), Unsafe (US). Unlabeled transitions are state-based transitions which correspond to crossing cell boundaries in a reachability-based partition of the state-space, described in [24].

[2] S. Vakil, A. Midkiff, T. Vaneck, and R. Hansman, "Mode awareness in advanced autoflight systems," in *Proc. IFAC Symp. on Analysis, Design, and Evaluation of Man-Machine Sys.*, Cambridge, MA, 1995.

[3] L. Sherry and R. Feary, "Task design and verification testing for certification of avionics equipment," in *Proc. of the AIAA/IEEE Digital Avionics Systems Conf.*, Sept. 2004, pp. 10.A.3–10.A.10.

[4] N. Sarter and D. Woods, "How in the world did we get into that mode? Mode error and awareness in supervisory control," *Human Factors*, vol. 37, no. 1, pp. 5–19, 1995.

[5] A. Suzuki, T. Ushio, and M. Adachi, "Detection of automation surprises in discrete event systems operated by multiple users," in *SICE-ICASE Int'l Joint Conf*, Korea, Oct. 2006, pp. 1115–1119.

[6] S. Umeno and N. Lynch, "Safety verification of an aircraft landing protocol: A refinement approach," in *Hybrid Systems: Computation and Control*, LNCS 4416, A. Bemporad, A. Bicci, and G. Buttazzo, Eds. Springer Verlag, April 2007, pp. 557–572.

[7] J. Crow, D. Javaux, and J. Rushby, "Models and mechanized methods that integrate human factors into automation design," in *Int. Conference on HCI in Aeronautics*, Toulouse, France, Sept. 2000.

[8] A. Joshi, S. P. Miller, and M. P. Heimdahl, "Mode confusion analysis of a flight guidance system using formal methods," in *IEEE Digital Avionics Systems Conference (DASC 2003)*, Oct. 2003.

[9] R. Boyatt and J. Sinclair, "A lightweight formal methods perspective on investigating aspects of interactive systems," in *Proc. Int'l Workshop on Formal Methods for Interactive Systems*. Elsevier, Sep. 2007.

[10] C. Tomlin, J. Lygeros, and S. Sastry, "A game theoretic approach to controller design for hybrid systems," *Proc. of the IEEE*, vol. 88, no. 7, pp. 949–970, 2000.

[11] C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi, "Computational techniques for the verification of hybrid systems," *Proc. of the IEEE*, vol. 91, no. 7, pp. 986–1001, 2003.

[12] I. Mitchell, *A Toolbox of Level Set Methods*, Department of Computer Science, University of British Columbia, June 2004, www.cs.ubc.ca/∼mitchell/ToolboxLS.

[13] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo, *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, available from `http://www.cds.caltech.edu/sostools`, 2004.

[14] G. Frehse, "PHAVer: Algorithmic verification of hybrid systems past HyTech." in *Hybrid Systems: Computation and Control*, LNCS 34143, M. Morari and L. Thiele, Eds. Springer Verlag, 2005, pp. 258–273.

[15] M. Kvasnica, P. Grieder, and M. Baotic, "Multi-Parametric Toolbox (MPT)," 2004. [Online]. Available: http://control.ee.ethz.ch/∼mpt

[16] A. Chutinan and B. Krogh, "Computational techniques for hybrid system verification," *IEEE Trans. on Automatic Control*, vol. 48, no. 4, pp. 64–75, Jan. 2003.

[17] E. Asarin, T. Dang, and A. Girard, "Reachability analysis of nonlinear systems using conservative approximation," in *Hybrid Systems: Computation and Control*, LNCS 2623, O. Maler and A. Pneuli, Eds. Springer Verlag, March 2003, pp. 20–35.

[18] J. Bowen and S. Reeves, "Formal models of informal GUI designs," in *Int'l Conf. on Software Formal Methods for Interactive Systems*, Electronic Notes in Theor. Computer Science. July 2007, pp. 57–72.

[19] A. Degani and M. Heymann, "Formal verification of human automation interaction," *Human Factors*, vol. 44, no. 1, pp. 28–43, 2002.

[20] A. Cerone, P. Lindsay, and S. Connelly, "Formal analysis of human-computer interaction using model-checking," in *Proc. of the IEEE Int'l Conf. on Software En. and Formal Methods*. Sept. 2005, pp. 352–361.

[21] W. Hussak and S. Yang, "Formal development of remote interfaces for large scale real-time systems," in *IEEE Int'l Conference on Systems, Man, and Cybernetics*, 2004, pp. 124–129.

[22] C. Lesire and C. Tessier, "Estimation and conflict detection in human controlled systems," in *Hybrid Systems: Computation and Control*, LNCS 3927, J. Hespanha and A. Tiwari, Eds. Springer Verlag, March 2006, pp. 407–420.

[23] M. Oishi, I. Mitchell, A. Bayen, C. Tomlin, and A. Degani, "Hybrid verification of an interface for an automatic landing," in *Proc. IEEE Conf. on Decision and Control*, Las Vegas, NV, 2002, pp. 1607–1613.

[24] M. Oishi, I. Mitchell, A. M. Bayen, and C. J. Tomlin, "Invariance-preserving abstractions of hybrid systems: Application to user interface design," *IEEE Trans. on Control System Technology*, 2007, p. 229-244, Vol. 16, No. 2, March 2008.

[25] P. Ladkin and H. Sogame, "Aircraft accident investigation report 96-5," http://sunnyday.mit.edu/accidents/nag-contents.html, July 1996.

[26] D. McRuer, I. Ashkenas, and D. Graham, *Aircraft dynamics and automatic control*. Princeton University Press, 1973.

[27] A. Bayen, I. Mitchell, M. Oishi, and C. Tomlin, "Reachability analysis and controller synthesis for autopilot design," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 68–77, 2007.