7-12-2014

Observability of user-interfaces for linear hybrid systems under collaborative control

Tasha Hammond

Tasha Hammond
Candidate

Electrical and Computer Engineering
Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Meeko Oishi , Chairperson

Chaouki Abdallah

Rafael Fierro

Observability of user-interfaces for linear hybrid systems under collaborative control

by

Tasha Hammond

B.S., Engineering Science, Trinity University, 2011

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Electrical Engineering

The University of New Mexico

Albuquerque, New Mexico

May, 2014

Dedication

To my mom and husband for their support, encouragement, and patience as I strove toward completion of this work.

Acknowledgments

I would like to thank my advisor, Dr. Meeko Oishi, for her thorough involvement and support through this process. Because of you, I have been introduced to and contributed to quality hybrid systems research. I would also like to thank my thesis committee for taking time out of their busy schedules to evaluate my work. I would like to thank the Department of Electrical and Computer Engineering for high level of education they provide and for preparing me to complete this work. Also, I greatly appreciate the work and advice given to me by the Electrical and Computer Engineering Graduate Student Coordinator Elmyra Grelle.

Observability of user-interfaces for linear hybrid systems under collaborative control

by

Tasha Hammond

B.S., Engineering Science, Trinity University, 2011

M.S., Electrical Engineering, University of New Mexico, 2014

Abstract

Human interaction with automation is ubiquitous, occurring in many cyberphysical systems such as cell phones, automobiles, and commercial aircraft. When interacting with such systems, human users are only exposed to a simplified representation the complex system structure in the form of an interface. The human can observe system outputs and make control inputs via this interface. Problems with human-automation interaction occur when the interface does not provide enough information or provides misinformation about the underlying system, such that the human cannot determine the current state of the automation. The user's knowledge of the current system state and prediction of the next system state is required for effective operation of an automated system. In this work, formal methods are employed to analyze user-interfaces of such cyberphysical systems in order to reveal state observability problems. The cyberphysical systems are modeled as hybrid systems, for which continuous behavior emerges from the laws of physics and discrete behavior results from logical conditions and rules governing the automation. Hybrid systems with LTI continuous dynamics under collaborative control are considered, where collaborative control indicates that some events and inputs are controlled by a human operator while other events and inputs are controlled by the automation. The human user is assumed to be a special type of state observer, with additional requirements beyond a standard (automated) state observer. To reflect these additional requirements, sufficient conditions for *user-observability* and *user-predictability* of linear hybrid systems under collaborative control are developed. Algorithms are generated to evaluate a user-interface based on these conditions for user-observability and user-predictability. Then, the algorithms are applied to a hybrid system model abstraction of the longitudinal dynamics of an aircraft flight management system.

Contents

List of Figures

List of Tables

List of Algorithms

Glossary

AOA	Angle of attack
AP	Autopilot
CDU	Control and display unit
DES	Discrete event system
FCPC	Flight control primary computer
FCU	Flight control unit
FD	Flight director
FMGEC	Flight management guidance and envelope computer
FMS	Flight management system
FPA	Flight path angle
GPS	Global positioning system
HAI	Human-automation interaction
IC	Integrated circuit
ILS	Instrument landing system

Glossary

LTI	Linear time-invariant
PAHS	Piecewise-affine hybrid system
PFD	Primary flight display
SLS	Switched linear system

Chapter 1

Introduction

1.1 Background

Before the word "computer" was ever used to describe a device, it was used to refer to a person who completed long calculations. These people, or computers, were employed during World War II to calculate the proper firing angle for artillery under a variety of conditions. Three thousand firing angle calculations were compiled into a single firing table, which would take a team of computers a month to finish [11]. In an attempt to reduce this computing time, John Mauchly proposed a design for an electronic computing device, and in 1945 Mauchly and J. Presper Eckert finished the construction of an electronic computer they named Electronic Numerical Integrator and Calculator (ENIAC). This computer took up an entire room and required 18,000 vacuum tubes [11].

Since then, a technological explosion has taken place, aided by the invention of the transistor in 1947 and the invention of the integrated circuit (IC) ten years later [8]. The constant improvement in size and performance of the transistor and IC has made powerful handheld computers, like smartphones, possible. With this technology, computers can not only be used in place of human labor to complete tedious calculations as in 1945, but they can now be used to complete much more complex tasks that were once performed strictly by humans. Examples include autopilot systems in aircraft, which perform many stabilization and navigation maneuvers in place of a human user, and the Therac-25, a medical device which administers radiation therapy to cancer patients automatically [30]. Relieving humans of tasks such as these and entrusting these tasks to devices with sufficient computing capabilities is hereafter referred to as *automation*.

While automation has been implemented in a broad range of industries including aviation, nuclear power, manufacturing, and medicine to relieve humans of tedious tasks, automation has also been implemented to reduce the opportunity for human error as a contributing cause of accidents. This approach to automation is especially prevalent in the aviation industry, where accidents can be fatal. FAA investigations have determined that over half of aircraft accidents are the result of human error [18]. To reduce or perhaps, eliminate this statistic, many aircraft functions were automated. However, aircraft accident rates have remained relatively constant since the 1970's despite increasingly automated aircraft. This suggests that automation designers are missing some key information about aircraft accidents, and perhaps accidents associated with automated systems overall.

This missing information may be found by noting that automation has not entirely replaced humans in the previously mentioned industries but rather aided humans. Thus, humans must interact with the automation frequently to accomplish a joint goal. This human-automation interaction (HAI) would then appear to play a large role in system functionality and take priority during the design process. But it seems likely that many automation designers overlooked HAI, thereby designing the automation as an independent system element rather than a system element inextricably linked to the human operator.

As a result of automation design without regard for HAI, automated systems are often complex and seem to behave in a counterintuitive manner. The complexity of modern automated systems often arises from the abundance of modes, or types of system behavior. While only one mode may be active at a given time, the ability of an automated system to execute many different behaviors via modes allows for great flexibility because a single task may be accomplished in a number of different ways [30]. But such flexibility can generate situations in which the user experiences "non-determinism," where different outcomes manifest after pushing the same button under apparently identical circumstances [13], [26], [15]. that are perceived as non-determinism by a human user. wealth of modes also creates the opportunity for much confusion.

Non-determinism can make it impossible for the user to ascertain the current mode of the system. Such occurrences are termed "mode confusion" in the literature [30]. Furthermore, automated systems often present the user with too much information, such that all or most of the information becomes useless [5].

These examples of poor HAI threaten the productivity and functionality of automated systems as well as the welfare of the human stakeholders. Aircraft are particularly sensitive to poor HAI because they are highly complex and dynamic systems. Thus, aircraft are especially important in the study of HAI.

Furthermore, aircraft, like other automated systems, are sometimes referred to as cyberphysical systems, which are physical processes controlled using computational elements. Cyberphysical systems exhibit both continuous dynamics, which arise from the laws of physics, and discrete dynamics, which result from digital logic. Due to the high complexity of such systems, human users often interact with the system via a simplified realization, called the *interface.* Since interfaces represent the underlying system behavior, they must capture both the continuous and discrete dynamics of cyberphysical systems. Therefore, cyberphysical system interfaces can be modeled mathematically as hybrid systems, which provide a rigorous framework to represent systems with both continuous and discrete dynamics. In this work, interfaces for cyberphysical systems are modeled as linear hybrid systems in order to analyze the interfaces for HAI problems.

1.2 Motivation

As mentioned previously, incidents and accidents caused by HAI problems, especially in aircraft, can be dangerous [30], [33]. A specific incident of interest is the 2009 Air France Flight 447 from Rio de Janeiro to Paris [1]. On this A-330 aircraft, the Pitot tubes measuring air pressure became obstructed by ice crystals and gave erroneous readings. These air pressure measurements were also used to determine the aircraft speed, which also became erroneous. The abnormally low speed reading due to the faulty pressure measurement caused the autopilot and autothrust to disengage, thereby abruptly relinquishing control of the aircraft to the human operators. Loss of the autopilot and autothrust capabilities cause the aircraft control law to change from “normal” to “alternate,” which entailed loss of flight envelope protections for pitch and angle of attack.

A lack of sufficient communication between the automation and the flight crew prevented the flight crew from quickly taking notice of the control law change. This lack of communication continued as the crew struggled to diagnose the root cause of the problem: the iced Pitot tubes and faulty pressure measurements. Without this indication or an indication of the loss of flight envelope protections, the pilots sent excessive pitch-up commands to the elevators. These continued pitch-up commands caused the aircraft’s angle of attack to become too steep, eventually causing the engines to stall and the aircraft to enter an unrecoverable dive [1].

This incident raises concerns about the information available to pilots during unusual flight conditions, such as erroneous speed readings. Considering the flight displays and flight control panels as a user-interface through which the pilots can interact with the automation, it is paramount that relevant information is properly conveyed to the flight crew through such a user-interface. The purpose of this work is to analyze the observability of the user-interface from the pilot's perspective to reveal design flaws and prevent later HAI-related accidents.

1.3 Related Work

The problem of human-automation interaction has been investigated by two general groups of researchers: the human factors, or engineering psychology, research community and the formal methods research community. The human factors research community is concerned with the design of devices and equipment that optimize the productivity, comfort, and safety of the human(s) that must interact with these devices and equipment [18]. This research community has adopted the philosophy that overall system function and performance will improve if "human-centered design" principles are implemented, where human-centered design caters to human needs and preferences. Justification for this design philosophy lies in the fact that humans ultimately bear the responsibility of safe and effective machine operation, so the human operator should be the first priority in the design process [5]. Much of the human factors research is based on historical data available in accident reports, empirical data gathered by surveying human users about their experience with automated systems, or by observing human reactions to abnormal automation conditions in a simulator [30], [33].

In reference to formal methods, we mean not only model checking and reachability analysis [7], [24], but also general mathematical techniques.The formal methods research community is concerned with developing mathematical representations of various automated systems in order to analyze and evaluate them for certain properties. An aim of formal methods research is also to develop analysis techniques that are applicable to many types of systems, regardless of context. An advantage of the formal methods approach to researching HAI is that many different physical systems can be represented by a single model with variable parameters. Once an acceptable model is constructed, appropriate design principles can be quantified, and systems can be evaluated based on those concrete principles. However, the formal methods approach has the disadvantage of being unable to fully capture many real phenomena since system models are not perfect representations.

1.3.1 Human Factors

The researchers in the human factors community were some of the first to identify and study the problems associated with HAI. In particular, HAI in aircraft flight-deck automation became a major focus of human factors research because aircraft are highly dynamic systems for which skilled operators are required, and aircraft accidents can be quite serious [5]. Although aircraft accidents are few, occurring at a rate of 2 to 3 accidents per million departures, and this rate has stayed relatively constant since the 1970s, aircraft traffic has more than doubled since that time [3]. Such traffic growth increases the absolute number of accidents over a given time period even though the accident rate remains constant. Since public perception of air travel is based on absolute number of aircraft accidents, it is imperative that the number of accidents be continually reduced in order to compete with the growing number of departures [3].

Furthermore, over half of those aircraft accidents are believed to be caused by human error [3], [33]. In an attempt to eliminate the opportunity for human error in aircraft, many aircraft control functions were automated. But research has shown that the possibility for error has not been reduced overall, but changed [30]. According to researchers, these remaining errors are also non-random, occurring as a result of traceable factors [22]. Human factors researchers believe that these factors stem from breakdowns in HAI and can be corrected by employing human-centered design principles.

Early human factors research that is still cited today established a basis for human-centered design. Fitts' list of tasks that "men are better at, machines are better at" (MABA-MABA), suggest that humans lack the skill to assimilate a large amount of information at one instant in time or perceive minuscule changes in data sequences or graphs of data points [16]. This indicates that humans are poor monitors of information that automated systems generate [33].

With the advanced flight management systems (FMS) of today's aircraft, such as the A-320, A-330, A-340, and the B-777, much of the flight control tasks are automated. As such, the aircraft operator is forced into a role of monitoring automation performance rather than actively controlling the aircraft. This role of the human as a monitor and supervisor opposes the suggested human role implied by Fitt's list, namely, an active controller.

However, Weiner and Curry have noted that it is not necessary to place the human in a supervisory role where he is inherently disadvantaged [33]. In fact, monitoring tasks and control tasks can be automated independently of one another [33], which means that the automation can be tasked with monitoring and the human can be tasked with active machine control without sacrificing system functionality. This automation configuration would improve system performance because each system component, namely, the human and the automation, is utilized according to its strengths.

The level of automation associated with each type of task—monitoring or

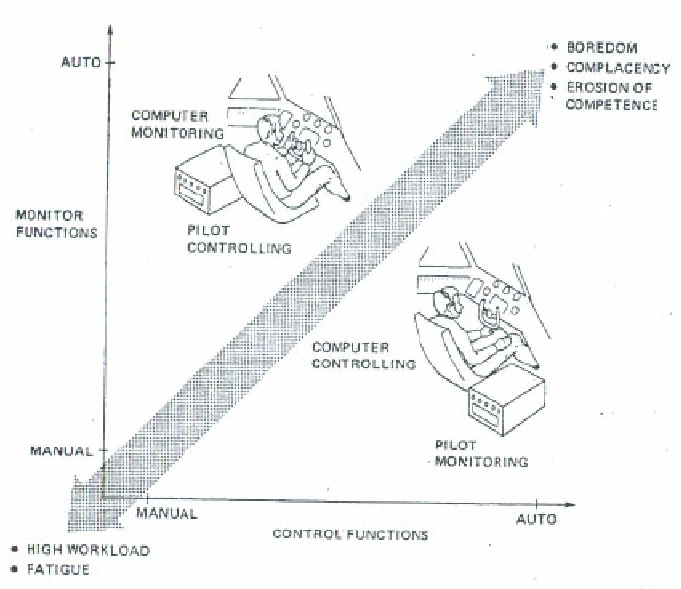

Figure 1.1: Graph describing independent control and monitoring tasks as well as levels of automation for each task type [33].

controlling—also affects the human's role in HAI. In particular, a high level of automation for control tasks produces the same result as that mentioned previously [33]. The human is forced into a supervisory position where he is weakest. On the other hand, if low-level control tasks, such as aircraft stability maintenance, were automated but high-level control tasks remained under the user's command, the user's workload would be reduced and his ability to use his strengths would be preserved. Figure 1.1 shows the relationship between type and level of automation in aircraft according to Weiner and Curry [33].

Parasuraman et al. also explained that automation can be separated into types and levels, but they developed a framework of four automation categories: information acquisition, information analysis, decision and action selection, and action implementation [27]. Similarly, each type of automation can be automated independently of the other types. These authors propose that with this framework, the appropriate level of automation can be determined for each automation type in a given application, thereby simplifying the automation designers job of creating a high-performance automated system that also promotes effective HAI [27].

We acknowledge that categorizing automation into types and levels can inform future design of flight-deck automation, but this method does not offer a solution to the problems associated with flight-deck automation currently in operation. Since the automation configuration which places the human in a supervisory position is currently in use in most commercial aircraft, we aim to develop techniques which will improve HAI for these systems.

First, we recognize that despite the human's disadvantageous position, the human must understand system behavior and know the state of the automation in order to be an effective supervisor. This requires that the automation convey correct and relevant information to the user [30]. The user receives information about the automation through a device called the *interface*. Therefore, any information that the user needs to reconstruct the system state must be available in the interface.

But an automation designer must determine which information is relevant. It is also likely that the relevant information will change depending on a given situation or mode of aircraft operation [20]. A test flight of the A-330 in Toulouse, France, demonstrated that designing interfaces to present relevant information is not a trivial problem. During the flight, the pilot tried a go-around (aborted landing) maneuver with a simulated engine failure. An unexpected mode change and subsequent "de-cluttering" scheme on the interface occurred during the attempted go-around, causing the flight envelope protections to disengage without the pilots knowledge. The decluttering scheme was a construct created to reduce the overwhelming amount of information contained in the interface that required the pilot's attention. But the decluttering scheme hid information about the safe flight envelope boundaries from the pilot, which ultimately led to a stall and fatal crash [29], [3]. The decluttering scheme clearly contributed to the accident because of the lack of relevant information

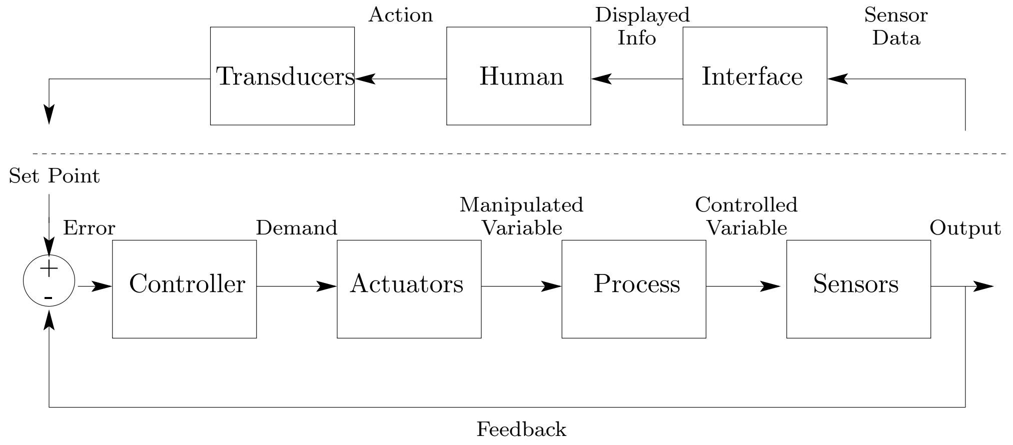

Figure 1.2: Simple negative feedback control system model, similar to that in [20].

it provided.

Other examples of the flight crew lacking enough information from the automation are cited in [33] and describe incidents during which the automation fails gradually. This automation tendency is not only dangerous because the failure is not indicated explicitly to the crew, but also because the failure may cause nearly imperceptible changes to the aircraft dynamics at first. Such situations may go unnoticed by the crew until the aircraft is near the limits of safe operation [33].

Relevant information is also difficult to discern for complex systems. Modern aviation automation has become complicated with the increased number of modes and highly coupled nature of those modes. Many modes allow for great system flexibility, such as the fact that one device can perform multiple tasks or the same task can be accomplished in multiple ways [20], [5]. But additional flexibility of a mode-rich system comes with great complexity.

Jamieson and Vicente note that a proliferation of modes creates greater opportunity for system failure and complicates diagnosis of a problem [20]. To properly identify a problem and mitigate its effects on the system, analytical redundancy must

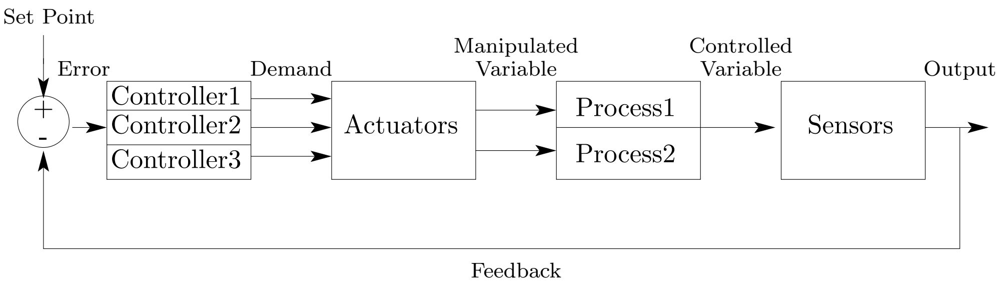

Figure 1.3: Simple negative feedback control system model showing different controller and process modes.

be maintained [20]. Analytical redundancy is achieved when a model of each system component is compared with a measured output signal from each component. A system is deemed to be functioning properly if the predicted output of each system component, according to the component model, closely resembles the measured output. But then the operator must have access to a model of each element in the system and each signal between elements of the system, which greatly increases the amount of information the operator must process [20]. A simple negative feedback control loop diagram is given in Figure 1.2 to illustrate the system components (boxes) and signals (arrows) for which the user must have information in order to identify problems effectively. To illustrate multiple controller and process modes, the controller and process boxes have been segmented into different parts, as shown in Figure 1.3. These segments demonstrate the ever increasing amount of information the pilot needs to maintain analytical redundancy with the increasing number of modes.

Ultimately, the user must have access to the information that indicates the current mode of operation of the aircraft. The complexity of systems adds to the difficulty of understanding system behavior and knowing the modes that correspond to particular behaviors. Tracking automation behavior and associating such behavior with the appropriate mode is known as “mode awareness” [30], [20]. Disintegration of mode awareness is known as “mode confusion” and can result in “mode error,” in which the supervisor takes action appropriate for a one mode of the system but the action is inappropriate for the current mode [30], [20].

A lack of mode awareness can lead to serious accidents, especially in aircraft. This mode confusion often stems from inconsistencies between the pilot's interpretation of how the system works and the actual system functionality. The user's conceptual model of the system is known as his "mental model," which may not match the true system structure [30]. An ideal user-interface would display the relevant portions of the true system structure so that the system operation could not be misinterpreted from the information given.

However, currently existing interfaces for automated aviation systems can still be misleading. For this reason, Degani and Heymann used formal verification techniques to find discrepancies between a user's mental model and the system interface [13]. We also use mathematical techniques to evaluate a user-interface for relevant information, namely, information that will allow the user to reconstruct the current state of the system.

Mathematical techniques offer a huge advantage in the HAI research arena because they offer the ability to quantify vague problems and system requirements that human factors researchers have referenced only through ambiguous narrative. One mathematical approach taken in the investigation of HAI is user modeling [16], [23], [19]. These researchers chose to model the response of a human in the control loop. They found that the human exhibited an affect on the system like that of an integrator. These researchers, then, modeled the automated system as a series of concentric control loops where the human acted as an integrator and analyzed the system for instability.

Another mathematical approach which has become popular in recent years consists of formulating the problem of mode awareness as an observability problem. In particular, some researchers have quantified the human factors indication of humans' limited memory in order to mathematically account for the human as a state observer [26], [15]. But other techniques for automated observers of hybrid systems also inform the problem of mode awareness in relation to HAI [9], [12], [10], [6], [32]. We extend mathematical techniques from the research concerned with both the human as an observer and hybrid system observability to evaluate user-interfaces for relevant information.

1.3.2 Hybrid System Observability

Some researchers have discovered that mathematical techniques can be used to characterize the HAI problem of mode awareness as an observability problem. The formal definition of state observability is the ability to determine the state of the system from knowledge of the input and the corresponding output over some finite time interval [2]. If the user of an automated system cannot determine the system state or mode, problems arise. Observability analysis techniques can be used as a tool to reveal such problems.

Furthermore, hybrid systems are studied in relation to HAI because of the continuous and discrete behavior of cyberphysical systems. But while standard observability for linear time-invariant (LTI) autonomous systems is well-known, hybrid system observability is not well-defined. Different observability conditions for hybrid systems can be developed depending on the type of problem being solved and the assumptions being made. We now give a brief description of the relevant hybrid system observability problems and resulting observability conditions that exist in the literature.

Collins and van Schuppen developed necessary and sufficient observability conditions for piecewise-affine hybrid systems (PAHS) [12]. They introduced the concept of *detectability*, which asserts that a discrete event is detectable if it produces a measurable change in the output. A system is *event detectable* if all events are detectable. They also state that while linear systems are observable in infinitesimal time, meaning the system is observable in some small time increment $\epsilon > 0$, PAHS can be observable in either infinitesimal, finite, or infinite time [12].

Investigation of observability for switched linear systems (SLS) was done by Babaali and Pappas [10]. SLS are a class of hybrid systems in which the continuous state is governed by linear equations that switch according to the discrete state of the system. These researchers provide initial state an mode observability characterizations for systems with both autonomous and non-autonomous continuous dynamics and unobserved, arbitrary switching [10]. They assert that a mode is discernible from another mode if the continuous output of the first mode is different from that of the second. They also found that discernibility is independent of the time, or length, of observation. These researchers further claim that any discrete mode of an SLS is observable if and only if every pair of different modes is mutually discernible from one another [10]. They also develop conditions for observability of the initial continuous state from the continuous output.

Balluchi et al. propose a synthesis method for a hybrid system observer composed of two parts: a *location observer* and a *continuous observer* [6]. These researchers propose such an observer for hybrid systems with linear continuous dynamics and a discrete mode transition function that is not necessarily deterministic. They found that for a certain set of conditions on the hybrid system, the location observer can determine the discrete mode of the system after a finite number of time steps. Under other conditions, the continuous observer generates an estimation of the continuous state, where the estimation error is shown to converge exponentially to some maximum acceptable error value [6]. First, they consider the case in which the location, or discrete mode, of the system may be determined after a finite number of steps from only discrete information. Then, they consider the case in which the discrete inputs and outputs do not provide enough information to determine the discrete mode. For this case, the continuous inputs and outputs must be used to determine the discrete mode.

Vidal et al. extend the well-known Popov-Belevic-Hautus rank condition for standard autonomous LTI system observability to SLS [32]. These researchers show that the discrete mode can be distinguished from the continuous output observations alone if those observations lie in the range space of the observability matrix for that particular mode [32]. Once the current discrete mode is determined, familiar observability analysis methods can be employed to determine the current continuous state. This group of researchers also indicates that the switching times are observable if and only if the difference between observability matrices for any pair of modes is nonsingular [32].

Of these discussed hybrid system observability methods, none provides a solution to the specific problem we are interested in. We seek to characterize linear hybrid system observability where the continuous dynamics are non-autonomous, the input is partially unknown, and the human is the observer. Collins and van Schuppen characterized the observability of discrete modes as event detectability, using information about the discrete events and the continuous output. But this method fails to take into account the fact that humans cannot easily perceive changes in the continuous output, especially if these changes are quite small. Also, a measurable change for an automated observer will be quite different from that for a human observer. Because of humans' difficulty in perceiving changes in the continuous output, we choose to use only discrete information to reconstruct the discrete mode, when available. We then use information from the continuous output if the discrete information is inadequate for the user to reconstruct the system state. Our observability conditions further differ from event detectability in that a given output can only result from a single mode. This is more restrictive than simply requiring that the output change in some way in order detect an event.

Babaali and Pappas develop conditions for discrete mode and continuous state observability for SLS with both autonomous and non-autonomous dynamics, but they do not use discrete output information in the reconstruction of the system state. This assumes that either a discrete output map does not exist or that the observer has no access to the discrete outputs. Since aircraft automation and controls have been designed to provide feedback to the pilot in the form of light indicators, push buttons, and aural chimes, which can be modeled as discrete outputs, we choose to include the discrete output in the repository of information available to the user. Therefore, the observability conditions we develop include information from the discrete output.

Balluchi et al. describe conditions required to design an observer, which is a slightly different problem than the problem of system observability that we are interested in. Furthermore, these researchers do not require that the current state be observable immediately. We require that the system state be observable from current information because humans cannot remember long sequences of modes or states. Humans need to be able to reconstruct the current state immediately in order to properly control and/or monitor an automated system.

Vidal et al. provide useful conditions for observability of SLS with autonomous dynamics. However, our problem includes an input, so we extend these conditions to linear hybrid systems with non-autonomous dynamics.

1.3.3 Human As Observer

Other observability research has been done by assuming the human is a special type of observer, who cannot reliably remember past observations, is limited in the amount of information that he can process at once, and only has access to the current human input but not higher derivatives of his input [14], [15]. These assumptions are formulated into mathematical restrictions on the information available to the observer.

Such restrictions are motivated by Human Factors research into common HAI problems, where humans have been known to struggle to recall past observations, sometimes referred to as “mental bookkkeeping” [30]. Humans have also been known to struggle with an overwhelming amount of information or burdensome information processing while performing basic control tasks, as sometimes occurs during aircraft operation [33], [5]. Other difficulties human aircraft operators often face is spatial disorientation, sometimes associated with the umbrella term known as “situation awareness” [14]. Spatial disorientation can occur, for instance, when an aircraft pilot perceives a pitch-up attitude when the aircraft is, in fact, only undergoing linear acceleration. If the pilot increases the thrust to cause a linear acceleration, but then believes that the aircraft is ascending, his understanding of how his inputs affect the system is inaccurate. This phenomenon suggests that human operators cannot make use of the higher derivatives of their inputs to reconstruct the system state.

Oishi et al. used discrete event systems (DES) to analyze observability problems with human-machine interfaces [26]. They formulated the concept of *immediate observability* for deterministic DES in order to capture the importance of easily understandable interfaces in time and safety critical systems. Immediate observability requires that the human user be able to uniquely determine the current state of the DES from the current discrete output and either the next or last event [26]. The conditions these researchers developed for immediate observability were then used to analyze DES for likely observability issues. This group also extended the methods to include design of DES which were immediately observable minimal representations of the underlying system [26].

Eskandari and Oishi model human interaction with an automated system, like an aircraft, as a continuous LTI system under shared control [15]. A system under shared control is one for which the automation may control certain inputs, the human may control other inputs separately, and some inputs may be controlled by both the human and the automation [15]. As often seems a reasonable assumption from the information presented in aircraft manuals and accident reports, it is assumed that the user does not have knowledge of the automation input. Furthermore, the human is treated as a special type of state observer that cannot recall past outputs and cannot utilize higher derivatives of his input to reconstruct the system state [14], [15]. This assumption used to solve the state observability problem is manifest mathematically as a partially unknown input observability problem.

Similar to [9], projection matrices were used to eliminate the unknown input. But the successive multiplication of projection matrices was applied in order to eliminate each term of the unknown input individually, including the automation input, the combined control input, and the unknown derivatives of the human input. The resulting terms were then used to construct the *user-observable* subspace, which results from the assumption about a human observer and is smaller than the standard observable subspace. These researchers also defined a construct called the *user-predictable* subspace in order to accommodate the human factors belief that automated system must behave in a predictable manner to ensure effective HAI. The user-predictable subspace is a subset of the user-observable subspace.

The previously mentioned research into formal methods of representing the human as an observer relate to our problem because we also restrict our observability conditions to account for a human as the state observer. However, the research presented does not apply to hybrid systems. We extend the immediate observability conditions of [26] and the concepts of user-observable and user-predictable subspaces of [15] to formulate *user-observability* and *user-predictability* of linear hybrid systems.

1.4 Theoretical Contributions

The novel contribution of this work includes sufficient conditions for user-observability and user-predictability of discrete event systems, extended from the concept of immediate observability developed in [26]. The conditions developed for user-observability and user-predictability of discrete event systems is combined with methods for determination of the continuous user-observable and user-predictable subspaces for LTI continuous systems in [15] to generate sufficient conditions for user-observability and user-predictability of linear hybrid systems with partially unknown input. I have developed two algorithms utilizing these conditions to determine if a linear hybrid system is both user-observable and user-predictable or neither. The first algorithm indicates whether or not the user can uniquely reconstruct the initial hybrid state of the system. The second algorithm indicates whether or not the user can uniquely reconstruct both the current hybrid state and next hybrid state of the system. Furthermore, I have developed an algorithm describing construction of the user-observable hybrid subspace as well as an algorithm denoting construction of the user-predictable hybrid subspace. I then apply Algorithms 1 and 2 to the 2009 Air France Flight 447 scenario to demonstrate the appearance of problems with user-interface observability in real systems. I plan to submit this work to the *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans* with co-author Dr. Meeko Oishi during the spring of 2014.

1.5 Outline

Chapter 1 of this work introduces the concept of HAI and describes research done to investigate HAI problems from the Human Factors perspective. An overview of research on hybrid system observability is given. Research on mathematical techniques which characterize a human as an observer is also presented. The novel theoretical contributions of this work are also stated in Chapter 1. Chapter 2 specifies user-observability and user-predictability of discrete event systems, extended from the concept of immediate observability developed in [26]. Chapter 3 extends the familiar Popov-Belevic-Hautus rank condition to mode distinguishability for linear hybrid systems with partially unknown input. Also in Chapter 3 are algorithms detailing a procedure to determine if a linear hybrid system is both user-observable and user-predictable or neither. Chapter 3 concludes with an aircraft example abstracted from the Air France Flight 447 incident. Finally, Chapter 4 is devoted to concluding remarks and directions for future work.

Chapter 2

User-Observability and User-Predictability of Discrete Event Systems

Described briefly in Subsection 1.3.3, immediate observability of DES indicates that the current discrete state can be uniquely reconstructed from current information alone [26]. States that can be reconstructed from current information alone correspond to states that can be determined immediately. These immediately observable states are highly important in dynamic systems such as aircraft, where even short periods of operator confusion can lead to deadly accidents. In this chapter, the conditions for immediate observability are decomposed into to user-observability and user-predictability of DES. This decomposition will allow for further extension of user-observability and user-predictability to linear hybrid systems in Chapter 3.

2.1 Problem Formulation

Consider a deterministic discrete event system $G = (Q, \Sigma, \varphi)$, in which Q is a finite set of states, $\Sigma = \Sigma_o \cup \Sigma_{uo}$ is a finite set of events, composed of the set of observable events Σ_o and the set of unobservable events Σ_{uo}, and the state transition function is represented by $\varphi : Q \times \Sigma \to Q$. We presume the set of initial states is $Q_0 \subset Q$. We also define an output map $h : Q \to Y$, such that $h(q) = y$ for $y \in Y, q \in Q$ [26].

Define the following sets:

$$\begin{aligned}
Q_y &:= \{q \in Q \ \mid \exists y \in Y, y = h(q)\} \\
Q_{y'} &:= \{q' \in Q \ \mid \exists y' \in Y, y' = h(q'), q' = \varphi(q, \sigma)\} \\
I_\sigma^f &:= \{q' \in Q \mid \forall q \in Q, q' = \varphi(q, \sigma)\} \\
I_\sigma^b &:= \{q \in Q \ \mid \forall q' \in Q, q' = \varphi(q, \sigma)\}
\end{aligned}$$

where Q_y is the set of all states whose output is $y \in Y$, I_σ^f is the set of all states forward reachable through an event $\sigma \in \Sigma$ from any $q \in Q$, and I_σ^b is the set of all states backwards reachable through an event $\sigma \in \Sigma$.

The set of events Σ can also be partitioned into those events that are controlled by the automation (Σ_a) and those events that are under human authority (Σ_h). The automation-controlled events may represent automatic transitions (e.g., when the aircraft touches down, it automatically transitions into rollout mode to steer itself along the runway). While it is most straightforward to presume that the human-controlled events are observable (e.g., $\Sigma_h = \Sigma_o$) and automatic transitions are unobservable (e.g., $\Sigma_a = \Sigma_{uo}$) (as we do here in this paper), this assignment may be problem dependent. For example, if a display annunciates an automatic transition (for example, an audio indicator that indicates rollout mode is now in operation), it may be more appropriate to assign that transition to the set of observable events Σ_o. Or alternatively, consider a human-controlled event that is done so frequently that it the user is unaware of doing it, or a distracted operator who pushes a button while thinking about something else, and is unaware of the event they just triggered. In these cases, the human-controlled event may actually be best represented as an unobservable event in Σ_{uo}.

2.2 Methodology: User-Interface Analysis

For a user-interface to have correct and complete content, the interface must allow the user to uniquely reconstruct the current state of the system and predict the next state. We formulate sufficient conditions for these two concepts: *user-observability* and *user-predictability* for discrete event systems by extending the methods in [26] for *immediate observability.* As opposed to standard definitions of observability and predictability, only *current* information about the input and output may be taken into consideration when the observer is the user (as opposed to an automated observer).

2.2.1 User-Observability

Definition 1 *The deterministic DES $G = (Q, \Sigma, \varphi)$ with initial set of states Q_0 and an output map $y = h(q)$ is* user-observable *if the current state can be determined uniquely from the current output and the last event.*

Proposition 1 *The DES $G = (Q, \Sigma, \varphi)$ with set of initial states Q_0 and output map $y = h(q)$ for all $q \in Q$ is* user-observable *if and only if for any state $q \in Q$, $|Q_y \cap I_\sigma^f| \leq 1$, and for any initial state $q_0 \in Q_0$, $y_0 = h(q_0)$, $h^{-1}(y_0)$ exists.*

Proof 1 **(If)** *Assume $|Q_y \cap I_\sigma^f| \leq 1$. Then, at most, a single state is associated with every combination of the current output y and last event σ. Hence Definition 1 is satisfied.* **(Only if)** *Assume G is user-observable. The user can determine the* *current state from the current output and last event because both the output map h and the transition function φ are deterministic.*

2.2.2 User-Predictability

Definition 2 *The deterministic DES $G = (Q, \Sigma, \varphi)$ with initial set of states Q_0 and an output map $y = h(q)$ is* user-predictable *if the next state can be determined uniquely from 1) knowledge of the next output and the next event (if the next output is available), or 2) knowledge of the current output and next event (if the next output is not available and the current state is known), or 3) knowledge of the set of states with the current output and the set of backward reachable states through the next event (if the next output is not available and the current mode is not known exactly).*

Proposition 2 *The DES $G = (Q, \Sigma, \varphi)$ with initial set of states Q_0 and output map $y' = h(q')$ for all $q' \in Q$ is* user-predictable *if and only if for any state $q' \in Q$, $|Q_{y'} \cap I^b_{\sigma'}| \leq 1$, and for any initial state $q_0 \in Q_0$, $y_0 = h(q_0)$, $h^{-1}(y_0)$ exists.*

Proof 2 **(If)** *Assume $|Q_{y'} \cap I^b_{\sigma'}| \leq 1$. Then, at most, a single state is associated with every combination of next output y' and next event σ'. Hence Definition 1 is satisfied.* **(Only if)** *Assume G is user-predictable. The user can determine the next state from the next output and next event because both the output map h and the transition function φ are deterministic.*

Remark: In the specific case where the next output is known, user-predictability is independent of user-observability.

Proposition 3 *The DES $G = (Q, \Sigma, \varphi)$ with initial set of states Q_0 and output map $y = h(q)$ for all $q \in Q$ is* user-predictable *if and only if G is also user-observable, and for any initial state $q_0 \in Q_0$, $y_0 = h(q_0)$, $h^{-1}(y_0)$ exists.*

Proof 3 **(If)** *Assume G is user-observable. Then, the current state is known, so the next state can be determined exactly. Hence, Definition 2 is satisfied.* **(Only if)** *Assume G is user-predictable. The user can determine the next state from the next output and next event because both the transition function φ and the output map h are deterministic.*

If G is not user-observable, the set of states in $Q_y \cap I_\sigma^f$ can be used to determine q'.

Proposition 4 *The DES $G = (Q, \Sigma, \varphi)$ with initial set of states Q_0 and output map $y = h(q)$ for all $q \in Q$ is* user-predictable *if and only if for any state $q' \in Q_y \cap I_\sigma^f$, $|Q_{y'} \cap I_{\sigma'}^b| \leq 1$, and for any initial state $q_0 \in Q_0$, $y_0 = h(q_0)$, $h^{-1}(y_0)$ exists.*

Proof 4 **(If)** *Assume $|Q_{y'} \cap I_{\sigma'}^b| \leq 1$. Then, at most, a single state is associated with every combination of the next output and next event σ'. Hence Definition 2 is satisfied.* **(Only if)** *Assume G is user-predictable. The user can determine the next state from the next output and next event because both the output map h and the transition function φ are deterministic.*

2.2.3 Immediate Observability

The following proposition from [26] is shown to illuminate the relationship among user-observability, user-predictability, and immediate observability.

Proposition 5 *The system $G = (Q, \Sigma, \varphi)$ with initial set of states Q_0 is immediately observable if and only if the following conditions hold ($\forall \sigma, \sigma' \in \Sigma, \forall y \in Y$):*

1. *(For initial state: $y_0 = h(q_0)$) for all $q_0 \in Q_0$*
   - *(a) $h^{-1}(y_0)$ exists (Q_{y_0} available)*
   - *(b) $|Q_{y_0} \cap I^b_{\sigma'}| \leq 1$ (Q_{y_0} and the next event available)*
2. *(For any state: $y = h(q)$) for all $q \in Q$ and for $k \in N^+$*
   - *(a) $h(q(k-1)) \neq h(q(k))$ if $q(k) \in \varphi(q(k-1), \sigma \in \Sigma_{uo})$, and*
   - *(b) $|Q_y \cap I^f_\sigma| \leq 1$ (Q_y and the last event available), or*
   - *(c) $|Q_y \cap I^b_{\sigma'}| \leq 1$ (Q_y and the next event available), or*
   - *(d) $|I^f_\sigma \cap Q_y \cap I^b_{\sigma'}| \leq 1$ (Q_y, the last, and next events available).*

Note that conditions 2(b) and 2(c) are not synonymous with 2(d) unless $I^f_\sigma = I^b_\sigma$.

Proposition 6 *A DES $G = (Q, \Sigma, \varphi)$ is immediately observable if it is either user-observable or user-predictable.*

Proof 5 *For a DES G that is user-observable, conditions 2(b) and 2(d) of Proposition 5 hold, indicating that the system is immediately observable. For a DES G that is user-predictable, conditions 2(c) and 2(d) of Proposition 5 hold; hence G is immediately observable.*

Example: Consider a deterministic DES $G = (Q, \Sigma, \varphi)$ with $Q = \{1, 2, 3, 4\}, \Sigma = \{a, b\}, Q_0 = \{1, 4\}$, and the state transition function φ defined as in Figure 2.1. We define the output $Y = \{A, B\}$, and the output map h is defined by $h(1) = h(4) = B$ and $h(2) = h(3) = A$. The forward reachable states through event a

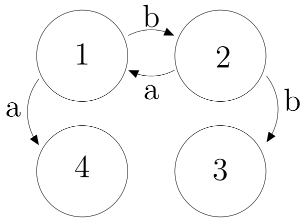

Figure 2.1: Illustrative example of a system which is immediately observable but not user-observable.

are $I_a^f = \{1, 4\}$. The forward reachable states through event b are $I_b^f = \{2, 3\}$. If any of these forward reachable states cannot be determined uniquely, the system is not user-observable because the condition described in Proposition 1 does not hold for any output $y \in Y$ and last event σ. For instance, applying the condition in Proposition 1 for output $y = B$ and last event a yields $|Q_B \cap I_a^f| = |\{1, 4\}| = 2$, so G is not user-observable. However, every combination of output and next event in this system is associated with, at most, one state. The backward reachable states through event a are $I_a^b = \{1, 2\}$, and the backward reachable states through event b are also $\{1, 2\}$. Applying condition 2(b) of Proposition 5 for both output $y = A$ and $y = B$ yields $|Q_A \cap I_a^b| = |\{2\}| = 1$, $|Q_A \cap I_b^b| = |\{2\}| = 1$, $|Q_B \cap I_a^b| = |\{1\}| = 1$, $|Q_B \cap I_b^b| = |\{1\}| = 1$. Thus, the DES is user-predictable, and hence immediately observable, but not user-observable.

Remark: For a continuous system, user-predictability implies user-observability [15]. However, for a discrete event system, user-predictability does not imply user-observability.

2.3 Summary

The definitions of user-observability and user-predictability for DES presented in this chapter can be combined with the conditions for user-observability and user-predictability of continuous LTI systems from [15] to generate conditions for linear hybrid systems. These conditions are developed in the next chapter.

Chapter 3

User-Observability and User-Predictability of Hybrid Systems

Sufficient conditions for user-observability and user-predictability of linear hybrid systems with partially unknown input are generated in this chapter. First, we present the formal framework used to solve the problem of state observability for linear hybrid systems. Then, mode distinguishability for linear hybrid systems with partially unknown input is developed. The algorithms incorporating conditions for user-observability and user-predictability are presented, and these algorithms are applied to real aircraft examples.

3.1 Problem Formulation

Consider an abstraction of the longitudinal dynamics of an aircraft flight management system as a hybrid system model represented by the tuple $H = (Q, X, \Sigma, R, \varphi, f_q)$, with discrete modes $q \in Q$, continuous state $x \in X$, discrete events $\sigma \in \Sigma$, continuous reference inputs $r \in R$, discrete transition function $\varphi : Q \times X \times \Sigma \times R \to Q$, for which we assume an identity reset map, and the continuous dynamics $f_q : X \times R \to X$ indexed by mode $q \in Q$, where $f_q = A_q x + B_q u + B_{q,\lambda}\lambda$, u is the primary human user's input, λ is the combined unknown automation input and input from other human users, and $u = -Kx + Nr$. We further define the hybrid output map $h : Q \times X \to Y_q \times Y_x = \Psi$, where the output Ψ is composed of both discrete and continuous elements, such as is shown using the notation $\Psi = (h_q(q), h_x(q, x)) = (y_q, y_x)$ where $y_x = C_q x + D_q u$.

Multiple representations of discrete events Σ for a hybrid system can be adopted. One description fitting of a problem concerning aircraft controls could consist of a set of discrete events that are initiated by the pilot Σ_{h1}, a set of events initiated by the copilot Σ_{h2}, and a set of events initiated by the automation Σ_{auto}. The mathematical description of this representation is given by $\Sigma = \Sigma_{h1} \cup \Sigma_{h2} \cup \Sigma_{auto}$.

However, we use an equivalent, but alternative, representation which highlights the known and unknown information from the perspective of a single user, such as the pilot. We choose to group the discrete events into those annunciated to the the pilot, and those not annunciated to the pilot. We write this as $\Sigma = \Sigma_1^{annun} \cup \Sigma_1^{non}$. Oftentimes, the events initiated by the automation and those initiated by human users other than the pilot will map to the set of events not annunciated to the pilot Σ_1^{non}. But this mapping will depend upon a given system structure.

The usefulness of this discrete event representation can be illustrated with a brief example. Consider the event σ_{AP}, which is used to represent the push-button input to engage the autopilot. First, assume that this event σ_{AP} is only annunciated to the user that initiates it. Also assume that the pilot initiates σ_{AP}. As such, σ_{AP} resides in Σ_1^{annun}, utilizing the representation which corresponds to the pilot's point of view. Similarly, if we use the representation which corresponds to the copilot's perspective, σ_{AP} lies in Σ_2^{non}.

3.2 Methodology: User-Interface Analysis

The observability techniques for DES described in Section 2.2 and the observability techniques for continuous LTI systems under collaborative control described in [15] can be combined to generate observability techniques for linear hybrid systems under collaborative control. An algorithm is developed which indicates whether or not the current hybrid state and the instantaneous next hybrid state can be determined uniquely given the system model H, limited information about the output Ψ, the discrete inputs $\sigma \in \Sigma$, and the continuous inputs $r \in R$. The ability of human operators of automated systems to reconstruct the current hybrid state and next hybrid state from currently available information constitutes effective HAI.

The algorithm takes advantage of the simplest information available to the user first, which is comprised of the discrete information. This approach is similar to that of Balluchi et al., where a location observer is first employed to determine the mode of a hybrid system and then a continuous observer is used to determine the continuous state [6]. However, the class of systems considered here—linear hybrid systems under collaborative control—have non-autonomous continuous dynamics, so standard continuous state observability cannot be utilized even once the discrete mode is known.

Furthermore, the class of systems under consideration in this work not only have the input present, but some inputs are controlled by the human user while other inputs are controlled by the automation, which we refer to as collaborative control. Also, observability restrictions arise from the fact that the human is considered to be the observer, rather than the automation. For instance, the human has knowledge of his continuous input but not higher derivatives of his continuous input, which we formalize as partially unknown input. The human operator also cannot be expected to remember past events. Finally, automation inputs as well as inputs generated by other human users may be unannunciated to the primary human user.

The Human Factors community claims that "good" human-centered design of automated systems allows the human user to determine the current state of the automation and be able to predict the behavior of the automation [33], [5]. We formulate these concepts mathematically as user-observability and user-predictability, respectively.

Even with these restrictions placed on the information available for state reconstruction, the current and next mode may be determined from the discrete information alone. If this case manifests for a given system, then the continuous state can be determined using partially unknown input observability methods for LTI continuous systems described in [15]. If, however, the current and next modes cannot be determined uniquely from the discrete information, the continuous output must be used to distinguish the modes.

The following text enumerates a novel method of mode distinguishability via continuous output information for linear hybrid systems with partially unknown input. For convenience of notation, the input vector and the output vector are defined as $\mathcal{U} = \begin{bmatrix} u & \dot{u} & \ddot{u} & \dots & u^{(n-1)} \end{bmatrix}^T$ and $\mathcal{Y} = \begin{bmatrix} y_x & \dot{y}_x & \ddot{y}_x & \dots & y_x^{(n-1)} \end{bmatrix}^T$, respectively.

It is useful to note that the output vector of an autonomous LTI system can be expressed as the following.

$$\mathcal{Y} = \mathcal{O}x \tag{3.1}$$

As discussed in [2], a unique solution, or trajectory, x exists for (3.1) if $\mathcal{Y}$ lies in the range space of $\mathcal{O}$. In other words, trajectory x can be distinguished uniquely from the information contained in the output if the following condition holds.

$$rank(\mathcal{O}) = rank([\mathcal{O}\ \mathcal{Y}]) \tag{3.2}$$

Vidal et al. extend the well-known condition (3.2) to autonomous switched linear systems, where mode $q \in Q$ can be determined from observation of the continuous output if the following condition holds [32].

$$rank(\mathcal{O}_q) = rank([\mathcal{O}_q\ \mathcal{Y}]) \wedge rank(\mathcal{O}_{q'}) \neq rank([\mathcal{O}_{q'}\ \mathcal{Y}]) \tag{3.3}$$

Note that (3.3) must be evaluated for each possible mode pair in order to ensure that each discrete mode in the hybrid system can be determined uniquely via observation of the continuous output.

We extend this condition to non-autonomous linear hybrid systems. But first, we define the Hankel matrix in (3.4) to make further statements about the information contained in the continuous output.

The Hankel matrix is a square, lower triangular matrix.

$$\Gamma_q = \begin{bmatrix} D_q & \dots & 0 & 0 \\ C_q B_q & \dots & 0 & 0 \\ C_q A_q B_q & \dots & \vdots & 0 \\ \vdots & \dots & D_q & \vdots \\ C_q A_q^{n-2} B_q & \dots & C_q B_q & D_q \end{bmatrix} \tag{3.4}$$

A_q, B_q, C_q, D_q define the LTI continuous dynamics of a particular mode within the hybrid system [10]. The Hankel matrix also makes for a convenient way to express the output vector for non-autonomous continuous dynamics of a linear hybrid system where the input and higher derivatives of the input are known, as shown in (3.5).

$$\mathcal{Y} = \mathcal{O}_q x + \Gamma_q \, \mathcal{U} \tag{3.5}$$

Then, (3.3) becomes the following for non-autonomous linear hybrid systems.

$$rank(\mathcal{O}_q) = rank([\mathcal{O}_q \;\; (\mathcal{Y} - \Gamma_q \, \mathcal{U})]) \wedge rank(\mathcal{O}_{q'}) \neq rank([\mathcal{O}_{q'} \;\; (\mathcal{Y} - \Gamma_{q'} \, \mathcal{U})]) \tag{3.6}$$

We further extend (3.6) to linear non-autonomous hybrid systems with partially unknown input to reflect the special requirements of the human as an observer. However, we first rewrite (3.5) to include the partially unknown human input as well as terms to represent the automation input and the input of other human users. This formulation reflects the structure of an automated system under collaborative control.

$$\mathcal{Y} = \mathcal{O}x + \Gamma_{q,1}u + \Gamma_{q,2}\hat{u} + \Gamma_{q,3}\lambda \tag{3.7}$$

Equation (3.7) shows a decomposition of the input into known and unknown components, where $\hat{u}$ is a vector consisting of the time derivatives of the primary human user's input $\begin{bmatrix} \dot{u} & \ddot{u} & \dots & u^{(n-1)} \end{bmatrix}^T$, and λ constitutes the combined effect of inputs contributed by the automation and other users. Note that u is known, while $\hat{u}$ and λ are unknown.

Similar to the methods used in [9] and [15], consider the projection matrix $P_{q,1}$ such that $P_{q,1}\Gamma_{q,3} = 0$. This equation can also be expressed as $\Gamma_{q,3}^T P_{q,1}^T = 0$, so that

$P_{q,1}$ is in the left null space of $\Gamma_{q,3}$. In other words, $P_{q,1} = (\mathcal{N}(\Gamma_{q,3}^T))^T$, where $\mathcal{N}(\cdot)$ denotes the null space. Multiplying (3.7) by the projection matrix $P_{q,1}$ yields the following.

$$P_{q,1}\mathcal{Y} = P_{q,1}\mathcal{O}_q x + P_{q,1}\Gamma_{q,1} u + P_{q,1}\Gamma_{q,2}\hat{u} \tag{3.8}$$

A projection matrix $P_{q,2}$ can be used in the same way to set the unknown higher derivatives of the primary user's inputs to zero: $P_{q,2}\Gamma_{q,2} = 0$. We now multiply (3.8) by the projection matrix $P_{q,2}$.

$$P_{q,2}P_{q,1}\mathcal{Y} = P_{q,2}P_{q,1}\mathcal{O}_q x + P_{q,2}P_{q,1}\Gamma_{q,1} u \tag{3.9}$$

Now that the unknowns have been eliminated, the rank condition (3.6) can be further extended according to the following proposition.

Proposition 7 *Any mode $q \in Q$ is distinguishable from any mode $q' \in Q$ via information from the continuous output y_x if $rank(P_{q,2}P_{q,1}\mathcal{O}_q) = rank([P_{q,2}P_{q,1}\mathcal{O}_q \;\; P_{q,2}P_{q,1}(\mathcal{Y} - \Gamma_{q,1}u)]) \wedge rank(P_{q',2}P_{q',1}\mathcal{O}_{q'}) \neq rank([P_{q',2}P_{q',1}\mathcal{O}_{q'} \;\; P_{q',2}P_{q',1}(\mathcal{Y} - \Gamma_{q',1}u)])$.*

Proposition 7 is used to construct Algorithms 1 and 2, which indicate whether or not a hybrid system H is both user-observable and user-predictable.

Algorithm 1 User-Observability and User-Predictability of Initial Hybrid State (q_0, x_0)

Require: Observations $\psi \in \Psi$ are available, and sets $Q_{y_{q,0}}$ and $I^b_{\sigma'}$ are known.

Ensure: The initial hybrid state (q_0, x_0) is distinguishable.

```
for $q_0 \in Q$, $y_{q,0} = h_q(q_0)$ do
    if $h_q^{-1}(y_{q,0})$ exists $\wedge$ $|Q_{y_{q,0}} \cap I^b_{\sigma'}| \leq 1$ then
        if $rank(\mathcal{O}(q_0)) = n$ then
            return "Yes"
        else
            return "No"
        end if
    else if $rank(\mathcal{O}(q_0)) = rank([\mathcal{O}(q_0)\ \mathcal{Y}]) \wedge rank(\mathcal{O}(q')) \neq rank([\mathcal{O}(q')\ \mathcal{Y}])$ then
        if $rank(\mathcal{O}(q_0)) = n$ then
            return "Yes"
        else
            return "No"
        end if
    else
        return "No"
    end if
end for
```

Algorithm 2 User-Observability of Current Hybrid State (q, x) and User-Predictability of Next Hybrid State (q', x^+)

Require: Observations $\psi \in \Psi$ are available, and sets $Q_y, Q_{y'}, I^f_\sigma$, and $I^b_{\sigma'}$ are known.

Ensure: The current hybrid state (q, x) is distinguishable.

```
for $q, q' \in Q$ do
    for $\sigma \in \Sigma^{non}$, $q' = \varphi(q, \sigma)$ do
        if $(h_q(q), h_x(q, x)) \neq (h_q(q'), h_x(q', x^+)) \wedge |Q_y \cap I^f_\sigma| \leq 1 \wedge |Q_{y'} \cap I^b_{\sigma'}| \leq 1$ then
            if $\mathcal{O}_{H,x} = \mathbb{R}^n \wedge \mathcal{P}_{H,x} = \mathbb{R}^n$ then
                return “Yes”
            else
                return “No”
            end if
        else if $rank(P_{q,2}P_{q,1}\mathcal{O}_q) = rank([P_{q,2}P_{q,1}\mathcal{O}_q \quad P_{q,2}P_{q,1}(\mathcal{Y} - \Gamma_{q,1}u)]) \wedge rank(P_{q',2}P_{q',1}\mathcal{O}_{q'}) \neq rank([P_{q',2}P_{q',1}\mathcal{O}_{q'} \quad P_{q',2}P_{q',1}(\mathcal{Y} - \Gamma_{q',1}u)])$ then
            if $\mathcal{O}_{H,x} = \mathbb{R}^n \wedge \mathcal{P}_{H,x} = \mathbb{R}^n$ then
                return “Yes”
            else
                return “No”
            end if
        else
            return “No”
        end if
    end for
end for
```

The terms $\mathcal{O}_{H,x}$ and $\mathcal{P}_{H,x}$ in Algorithm 2 represent the continuous user-observable and user-predictable subspaces, respectively. The continuous dynamics contribute enough information for the hybrid system to be user-observable and user-predictable only if each subspace covers the entire state space [15].

Also, note that the hybrid system H is only user-observable and user-predictable if the conditions specified in both Algorithm 1 and Algorithm 2 are satisfied.

Proposition 7 is also used to generate Algorithms 3 and 4. Algorithm 3 proposes a method of construction of the hybrid user-observable subspace, while Algorithm 4 proposes a method of construction of the hybrid user-predictable subspace of the system H. If the system H is either not user-observable, or not user-predictable, or neither, Algorithms 3 and 4 can be used to determine where the system falls short of user-observability and/or user-predictability. The terms $\mathcal{N}(\cdot)$ and $\mathcal{R}(\cdot)$ denote the null space and range space, respectively. The symbol $\odot$ represents the hybrid sum of subspaces.

Algorithm 3 User-Observable Subspace of Hybrid System H

```
Require: Observations $\psi \in \Psi$ are available, and sets $Q_{y_q}$ and $I_\sigma^f$ are known.
Ensure: User-observable hybrid subspace $\mathcal{O}_H$

$\mathcal{O}_{H,q} \leftarrow \{q \in Q \mid Q_{y_q} \cap I_\sigma^f \leq 1\}$
if $\mathcal{O}_{H,q} = \emptyset$ then
    $\mathcal{O}_{H,q} \leftarrow \{q \in Q \mid rank(P_{q,2}P_{q,1}\mathcal{O}_q) = rank([P_{q,2}P_{q,1}\mathcal{O}_q \;\; P_{q,2}P_{q,1}(\mathcal{Y}-\Gamma_{q,1}u)]) \wedge rank(P_{q',2}P_{q',1}\mathcal{O}_{q'}) \neq rank([P_{q',2}P_{q',1}\mathcal{O}_{q'} \;\; P_{q',2}P_{q',1}(\mathcal{Y}-\Gamma_{q',1}u)])\}$
    if $\mathcal{O}_{H,q} = \emptyset$ then
        return $\mathcal{O}_{H,q} = \emptyset$
    else
        $\mathcal{O}_{H,x} \leftarrow \mathcal{R}(C_q^T) \oplus A_q^T(\mathcal{R}(C_q^T) \cap \mathcal{N}(B_{q,\lambda}^T) \oplus \sum_{i=1}^{n-2} \mathcal{R}((A_q^i)^T C_q^T) \cap \mathcal{N}(B_{q,\lambda}^T) \cap A_q^T \mathcal{N}(B_q^T))$
        $\mathcal{O}_H \leftarrow \mathcal{O}_{H,q} \odot \mathcal{O}_{H,x}$        ▷ $\odot$: hybrid sum of subspaces
    end if
else
    $\mathcal{O}_{H,x} \leftarrow \mathcal{R}(C_q^T) \oplus A_q^T(\mathcal{R}(C_q^T) \cap \mathcal{N}(B_{q,\lambda}^T) \oplus \sum_{i=1}^{n-2} \mathcal{R}((A_q^i)^T C_q^T) \cap \mathcal{N}(B_{q,\lambda}^T) \cap A_q^T \mathcal{N}(B_q^T))$
    $\mathcal{O}_H \leftarrow \mathcal{O}_{H,q} \odot \mathcal{O}_{H,x}$        ▷ $\odot$: hybrid sum of subspaces
end if
```

Algorithm 4 User-Predictable Subspace of Hybrid System H

```
Require: Observations $\psi \in \Psi$ are available, and sets $Q_{y'_q}$ and $I^b_{\sigma'}$ are known.
Ensure: User-predictable hybrid subspace $\mathcal{P}_H$
$\mathcal{P}_{H,q} \leftarrow \{q \in Q \mid Q_{y'_q} \cap I^b_{\sigma'} \leq 1\}$
if $\mathcal{P}_{H,q} = \emptyset$ then
    $\mathcal{P}_{H,q} \leftarrow \{q \in Q \mid rank(P_{q,2}P_{q,1}\mathcal{O}_q) = rank([P_{q,2}P_{q,1}\mathcal{O}_q \;\; P_{q,2}P_{q,1}(\mathcal{Y}-\Gamma_{q,1}u)]) \wedge rank(P_{q',2}P_{q',1}\mathcal{O}_{q'}) \neq rank([P_{q',2}P_{q',1}\mathcal{O}_{q'} \;\; P_{q',2}P_{q',1}(\mathcal{Y}-\Gamma_{q',1}u)])\}$
    if $\mathcal{P}_{H,q} = \emptyset$ then
        return $\mathcal{P}_{H,q} = \emptyset$
    else
        $\mathcal{P}_{H,x} \leftarrow E_{\mathcal{O}_H}(\mathcal{N}(\bar{A}^T_{q,12}) \cap \mathcal{N}(\bar{B}^T_{q,\lambda\mathcal{O}_H}))$
        $\mathcal{P}_H \leftarrow \mathcal{P}_{H,q} \odot \mathcal{P}_{H,x}$        ▷ $\odot$: hybrid sum of subspaces
    end if
else
    $\mathcal{P}_{H,x} \leftarrow E_{\mathcal{O}_H}(\mathcal{N}(\bar{A}^T_{q,12}) \cap \mathcal{N}(\bar{B}^T_{q,\lambda\mathcal{O}_H}))$
    $\mathcal{P}_H \leftarrow \mathcal{P}_{H,q} \odot \mathcal{P}_{H,x}$        ▷ $\odot$: hybrid sum of subspaces
end if
```

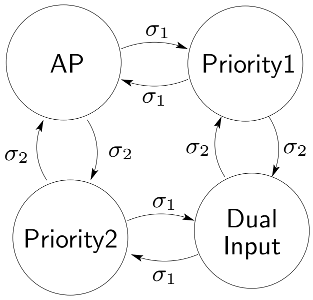

Figure 3.1: Pilot and copilot "fight" for control authority.

Now we present an aircraft example for which Algorithms 1 and 2 can be used to analyze the system for user-observability and user-predictability.

Example: Consider an abstraction of the longitudinal dynamics of an aircraft FMS as a linear hybrid system $H = (Q, X, \Sigma, R, \varphi, f_q)$, where $Q = \{Priority\ 1, Priority\ 2, Dual\ Input, AP\}$, the continuous state $x \in X$, the set of discrete events Σ composed of the human-initiated events σ_1 and σ_2, continuous reference inputs $R = \{r_1, r_2, r^-\}$, the state transition function φ is deterministic and defined as in Figure 3.1, and the continuous dynamics are defined as in Section 3.1.

We define the hybrid output map as in Section 3.1 as well. The set of discrete outputs is given by $Y_q = \{AP, MAN\}$, where $h_q(AP) = AP$ and $h_q(Priority\ 1) = h_q(Priority\ 2) = h_q(Dual\ Input) = MAN$.

The aircraft in *AP* mode signifies that autopilot is sending command signals to the flight controls. The aircraft in *Priority* 1 mode, shortened to $P1$ for convenience, is representative of the first user, or pilot, having authority over the flight controls. Similarly, *Priority* 2 mode, shortened to $P2$, is representative of the second user, or copilot, having authority over the flight controls. Finally, the mode *Dual Input*, shortened to *Dual*, represents the special case in which both the pilot and copilot take authority over the flight controls. Further explanation of this special case is given with respect to the users' continuous reference inputs.

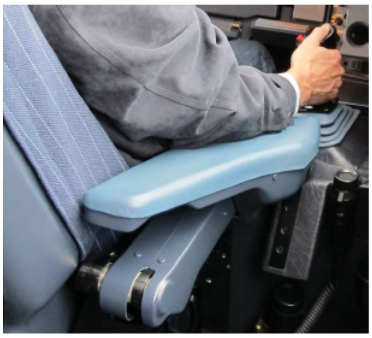

Figure 3.2: Right seat armrest showing the sidestick on an A-330 aircraft [1].

For further clarification, the event σ_1 represents the pilot commanding authority of the manual flight controls; this event occurs when the pilot presses the Priority button on his respective sidestick controller (Fig. 3.2). The event σ_2 represents the copilot commanding authority of the manual flight controls; this event occurs when the copilot, in turn, presses the Priority button on his respective sidestick controller. Normal aircraft operation procedure requires that the pilot not flying (PNF) call out, or notify, the pilot flying (PF) of his intent to take over the controls by pushing his respective Priority button. This procedure would indicate that both σ_1 and σ_2 are annunciated events. However, we consider the case, as occurs in the Air France Flight 447 [1], in which the flight crew do not notify one another of the intent to take the controls because they have become preoccupied with the event of an impending stall. By assuming the perspective of the first human user, the pilot, σ_1 is annunciated but σ_2 is not annunciated. Figure 3.2 supports the choice to allow event σ_2 to be unannunciated to the pilot because the copilot's sidestick controller movements and button presses are obscured from the pilot.

The open-loop dynamics of H are based on a linearized model of the longitudinal aircraft dynamics for a B-747 in level flight at 40,000 feet traveling with a horizontal speed of 774 feet per second (fps) [21], where the state vector is $x = [V, \alpha, \dot{\theta}, \theta]^T$. V represents deviations from the trim horizontal speed in fps, α is the angle of attack

(AOA) in radians, $\dot{theta}$ is the pitch rate in radians per second, and θ is the pitch angle in radians. The input is the elevator deflection δ_e in radians. The open-loop matrices are the same for each mode and are defined as follows.

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -0.003 & 0.039 & 0 & -0.322 \\ -0.065 & -0.319 & 7.74 & 0 \\ 0.020 & -0.101 & -0.429 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0.010 \\ -0.180 \\ -1.160 \\ 0 \end{bmatrix} \delta_e \\ y_x &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x \end{aligned} \tag{3.10}$$

The closed-loop dynamics are determined by a reference tracking feedback control law, where the reference inputs $r \in R$ vary for each mode.

Priority 1 Mode

$$\delta_e = -Kx + Nr_1$$

Priority 2 Mode

$$\delta_e = -Kx + Nr_2$$

Dual Input Mode

$$\delta_e = -Kx + N\Big(\frac{r_1 + r_2}{2}\Big)$$

AP Mode

$$\delta_e = -Kx + Nr^-$$

In $P1$ mode, it is assumed that the first user applies a reference input r_1, and in $P2$ mode the second user applies a reference input r_2. In *Dual* mode, both users have commanded control authority, so their reference inputs are averaged [1]. In AP mode, the reference input is the same as the previous reference input, meaning that the reference input remains the same as it was before the mode change into AP mode. This behavior reflects the attempt of many automation designers to create an autopilot that helps the human user by inferring the human's intent.

Before we discuss the role of the reference inputs further, we analyze the discrete modes and transitions. Let the initial system mode be $Q_0 = AP$. By first applying Algorithm 1 to H, it is clear that the system meets the conditions stated in lines 2 and 3 of Algorithm 1. Therefore, the initial hybrid state is both user-observable and user-predictable.

Applying Algorithm 2 to H indicates that the system fails to meet the conditions in line 3. For instance, the first term in line 3 states that the discrete output of two modes, between which an unannunciated event occurs, must be different. The system fails to meet this condition for the transitions between modes *Priority* 1 and *Dual Input* since the event σ_2, representing the copilot's attempt to take control authority, is unannunciated to the pilot.

Furthermore, the system fails the second and third conditions in line 3 of Algorithm 2. Note that $Q_{MAN} = \{P1,\ P2,\ Dual\}$, $Q_{AP} = \{AP\}$, and $I^f_{\sigma_1} = \{P1,\ P2,\ Dual,\ AP\} = I^b_{\sigma_1} = I^f_{\sigma_2} = I^b_{\sigma_2}$. Therefore, only the interection of set Q_{AP} and each of the forward and backward reachable sets for each event will meet the conditions. For instance, $|Q_{AP} \cap I^f_{\sigma_1}| = |\{AP\}| = 1$ and $|Q_{AP} \cap I^b_{\sigma_2}| = |\{AP\}| = 1$. But the system fails the conditions for the intersection of set Q_{MAN} and the forward and backward reachable sets, like $|Q_{MAN} \cap I^f_{\sigma_1}| = |\{P1,\ P2,\ Dual,\ AP\}| = 4$.

Since the discrete information does not allow the pilot to uniquely reconstruct the current mode of the system, we must continue on to line 9 of Algorithm 2 in order to determine if the discrete mode can be distinguished via the continuous output.

In particular, we focus on distinguishing $P1$ mode from $Dual$ mode and vice versa because these modes share the same discrete output and are separated only by an unannunciated event.

The continuous output and its derivatives for $P1$ mode can be expressed as in (3.11), where only the pilot's input is present. Neither the automation nor the copilot have an effect on the control input in $P1$ mode. The continuous output and its derivatives for $Dual$ mode can be expressed as in (3.12), where the automation input does not appear but the copilot can input to the system.

$$\mathcal{Y} = \mathcal{O}_{P1}x + \Gamma_{P1}\ N\,\mathcal{U} \tag{3.11}$$

$$\mathcal{Y} = \mathcal{O}_{Dual}x + \Gamma_{Dual}\ \frac{N}{2}\,\mathcal{U} + \Gamma_{Dual}\ \frac{N}{2}\ \lambda \tag{3.12}$$

In this case, the term $\sqcap$ represents the reference input and its higher derivatives. However, the reference inputs are constant for this example, so the higher derivatives of the reference inputs are zero by construction. The matrices Γ_{P1} and Γ_{Dual} happen to be equal in this case. The value of Γ_{P1} is given in (3.13).

$$\Gamma_{P1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ CB & 0 & 0 & 0 \\ C(A-BK)B & CB & 0 & 0 \\ C(A-BK)^2B & C(A-BK)B & CB & 0 \end{bmatrix} \tag{3.13}$$

Note that projection matrices are not required to eliminate the higher derivatives of the pilot's reference input because they are already equal to zero. No automation inputs are present, so projection matrices are not required to eliminate those as unknown inputs either. Only in *Dual* mode does the copilot's input appear. A projection matrix is required to eliminate the unknown term associated with the copilot's input. However, eliminating the effect of λ using a projection matrix such that $P\Gamma_{Dual} = 0$ will also eliminate the effect of $\mathcal{U}$.

Eliminating the appropriate terms yields the following rank conditions to distinguish $P1$ mode from *Dual* mode according to line 9 of Algorithm 2.

$$\begin{aligned} rank(\mathcal{O}_{P1}) &= rank([\mathcal{O}_{P1} \ \ (\mathcal{Y} - \Gamma_{P1,1} N \ u)]) \wedge \\ rank(P\mathcal{O}_{Dual}) &\neq rank(P[\mathcal{O}_{Dual} \ \ (\mathcal{Y} - \Gamma_{Dual,1} \tfrac{N}{2} u)]) \end{aligned} \tag{3.14}$$

To distinguish *Dual* mode from $P1$ mode, the rank conditions become

$$\begin{aligned} rank(P\mathcal{O}_{Dual}) &= rank(P[\mathcal{O}_{Dual} \ \ (\mathcal{Y} - \Gamma_{Dual,1} \tfrac{N}{2} u)]) \wedge \\ rank(\mathcal{O}_{P1}) &\neq rank([\mathcal{O}_{P1} \ \ (\mathcal{Y} - \Gamma_{P1,1} N \ u)]) \end{aligned} \tag{3.15}$$

In this case, P is given by the following.

$$P = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.16}$$

However, checking the conditions reveals that these two modes cannot be distinguished from one another using the continuous output either. The second part of each rank condition requires non-equivalence of the left-hand-side and right-hand-side of the equation, but the system fails to satisfy this requirement.

While Algorithm 2 indicates that the modes of H cannot be distinguished via either the discrete or continuous information, there may be aspects of this problem

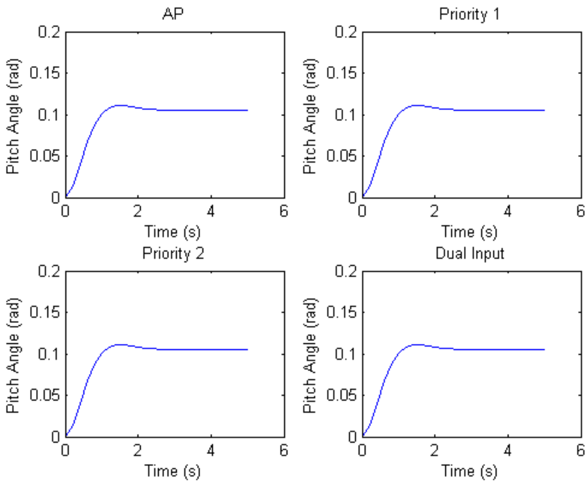

Figure 3.3: Closed-loop step responses for modes *AP*, *Priority* 1, *Priority* 2, and *Dual Input* when the reference inputs r_1, r_2, and r^- are equal.

that the algorithm does not yet capture. For instance, consider the scenario in which the reference inputs r_1 and r_2 are equal. Let $r_1 = r_2 = +6°$ (Fig. 3.3). In this situation, the discrete mode of aircraft operation does not matter because the reference input is the same for all of the modes, including *Dual* mode because the average of the reference inputs happens to be the same as the pilot's reference input alone. This configuration means that the continuous output is consistent with the pilot's expectations regardless of the mode. Thus, the pilot can accurately predict the continuous state of the system, a nuance that Algorithm 2 does not capture. This result illustrates the fact that the conditions for user-observability and user-predictability presented here are sufficient, but not necessary. In other words, there are very special cases in which a system may not need to satisfy the user-observability and user-predictability conditions to ensure effective human-automation interaction.

However, if even a slight change in the reference inputs occurs, this effect will be lost, and the system will once again fail to be user-predictable. When reference inputs r_1 and r_2 are significantly different, the negative effect on user-predictability is

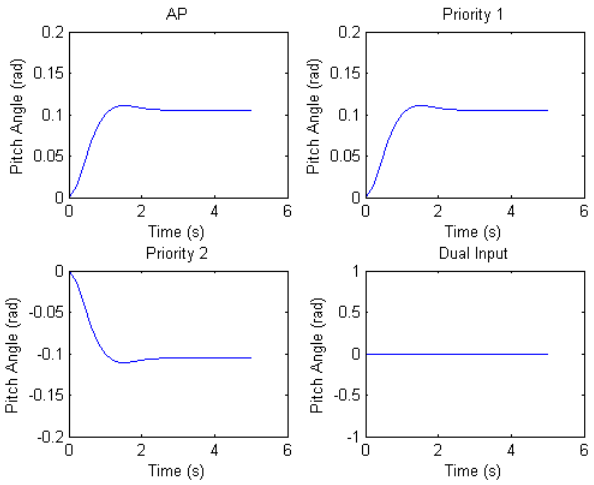

Figure 3.4: Closed-loop step responses for modes AP, $Priority$ 1, $Priority$ 2, and $Dual\ Input$ when the reference inputs r_1 and r_2 are not equal.

even more prevalent. Take the example in which $r_1 = +6°$ and $r_2 = -6°$ so that the pilot is commanding the aircraft to ascend and the copilot is commanding the aircraft to descend (Fig. 3.4). Also let the reference input in AP mode be equal to the pilot's last input so that $r^- = r_1$. Despite the fact that the reference inputs for AP mode and $P1$ mode are identical, the event σ_1 is annunciated to the pilot, so these modes are distinguishable. Furthermore, the pilot is be able to distinguish between modes AP and $P2$ due to the discrete output, but modes $P1$ and $Dual$ are indistinguishable via the discrete information. It also follows that the continuous output in $Dual$ mode is not only inconsistent with the pilot's input, but also inconsistent with the copilot's input, as neither the ascent command nor the descent command is executed. This system behavior leads to a kind of mode mismatch in which it is not only possible for the user to confuse two modes (mode confusion) but it also possible for the user to confuse any and all modes of the system at once.

A proposed solution to this problem of mode mismatch is to alter the discrete outputs so that $Dual$ mode is explicitly indicated in the flight mode annunciator,

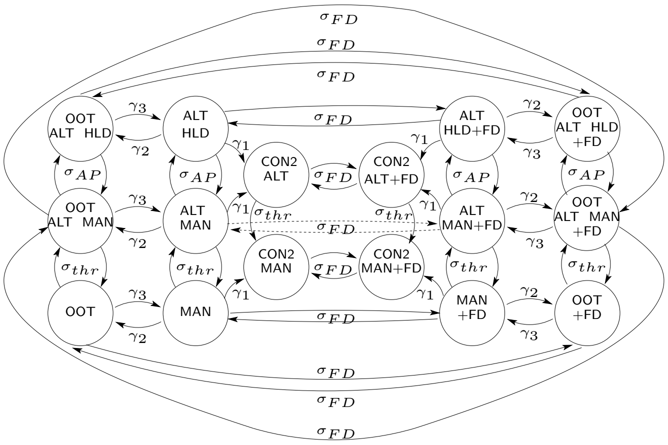

Figure 3.5: Hybrid system model of the aircraft FMS in the events leading up to the crash of Air France Flight 447.

or elsewhere in the flight displays. Then the discrete output for *Dual* mode could resemble the following: $h_q(Dual) = Double\ Input$.

The simple system shown in Figure 3.1, which characterizes a common scenario in flight, has been shown to be neither user-observable nor user-predictable in certain off-nominal flight conditions. The user of such a system would be unable to uniquely reconstruct the current state of the system or predict the next state of the system. This result suggests that the more complex aircraft systems also exhibit poor observability.

3.3 Example: Pilot Display

We consider the 2009 Air France Flight 447 from Rio de Janeiro to Paris, in which a number of HAI problems took place and eventually led to an unrecoverable stall situation.

Table 3.1: Summary of Discrete Outputs

Output (y_q)	Modes with Output y_q
AP	$ALT\ HLD$, $OOT\ ALT\ HLD$
$AP+FD$	$ALT\ HLD+FD$, $OOT\ ALT\ HLD+FD$
A/THR	$ALT\ MAN$, $CON2\ ALT$, $OOT\ ALT\ MAN$
$A/THR+FD$	$ALT\ MAN+FD$, $CON2\ ALT+FD$, $OOT\ ALT\ MAN+FD$
FD	$MAN+FD$, $CON2\ MAN+FD$, $OOT+FD$
MAN	MAN, $CON2\ MAN$, OOT

3.3.1 Model Description

We abstract the longitudinal dynamics of the FMS to construct a linear hybrid system $H=(Q,X,\Sigma,R,\varphi,f_q)$, with discrete modes $Q=\{ALT\ HLD$, $ALT\ HLD+FD$, $OOT\ ALT\ HLD$, $OOT\ ALT\ HLD+FD$, $ALT\ MAN$, $ALT\ MAN+FD$, $OOT\ ALT\ MAN$, $OOT\ ALT\ MAN+FD$, MAN, $MAN+FD$, OOT, $OOT+FD$, $CON2\ ALT$, $CON2\ ALT+FD$, $CON2\ MAN$, $CON2\ MAN+FD\}$, initial state $Q_0=\{ALT\ HLD+FD\}$, continuous state $x\in X$, the set of events Σ composed of the human initiated events σ_{AP}, σ_{thr}, and σ_{FD} as well as the automatic transitions γ_1, γ_2, and γ_3, continuous reference inputs $r\in R$, discrete transition function φ is defined in Figure 3.5, and the continuous dynamics are defined as in Section 3.1.

We define the hybrid output map $h: Q\times X\to Y_q\times Y_x=\Psi$, where the output Ψ is composed of both discrete and continuous elements as is shown using the notation $\Psi=(h_q(q),h_x(q,x))=(y_q,y_x)$. The set of discrete outputs is given by $Y_q=\{AP$, $AP+FD$, A/THR, $A/THR+FD$, FD, $MAN\}$, and h_q is summarized in Table 3.1. These discrete outputs are modeled from the indications visible in the pilot's flight mode annunciator (FMA), information for which was taken from [17] and [1]. The indicators appearing in the FMA for each mode are shown in Figure 3.6.

The discrete event σ_{FD} represents the pilot pushing the flight director (FD) but-

Table 3.2: Summary of Automatic Transitions

Event	Variables Affected	Condition	Annunciated Event?
γ_1	Speed (V)	$V_{avg}(t) - V_{avg}(t-1) \leq -30$ kts	No
γ_2	Angle of Attack (α)	$\alpha \geq \alpha_{prot}$	Yes
γ_3	Angle of Attack (α)	$\alpha < \alpha_{prot}$	Yes

ton on the flight control unit (FCU) panel (Fig. 3.7). The discrete event σ_{thr} represents the pilot pushing the autothrust button on the FCU panel. Also, a summary of the automatic transitions is given in Table 3.2.

The open-loop dynamics of H are based on a linearized model of the longitudinal aircraft dynamics for a B-747 in level flight at 40,000 ft and 774 fps [4]. The state vector is $x = [V, \alpha, \dot{\theta}, \theta]^T$ such that V is the deviation from the trim horizontal aircraft speed in fps, α is the AOA in radians, $\dot{\theta}$ is the pitch rate in rad/s, and θ is the pitch angle in radians. The actuator inputs δ_e and δ_t correspond to the elevator deflection and thrust, respectively.

$$\dot{x} = \begin{bmatrix} -0.003 & 0.039 & 0 & -0.322 \\ -0.065 & -0.319 & 7.74 & 0 \\ 0.020 & -0.101 & -0.429 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0.010 & 1 \\ -0.180 & -0.040 \\ -1.160 & 0.598 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix} \quad (3.17)$$

Since the authority over control surfaces varies among modes, Table 3.3 summarizes the modes for which the human or automation or both has control authority.

The continuous output y_x consists of the horizontal speed, pitch angle, and flight path angle, which are readily available in the displays for an A-330 aircraft [17].

Table 3.3: Control Authority in Various Modes of Flight 447

Modes	Goal	Automation Control	Human Control
$ALT\ HLD$	$\gamma \to 0$	δ_e, δ_t	—
$ALT\ HLD + FD$			
$OOT\ ALT\ HLD$	$\theta \to -3$	δ_e, δ_t	—
$OOT\ ALT\ HLD + FD$			
$OOT\ ALT\ MAN$	$\theta \to -3$	δ_e, δ_t	—
$OOT\ ALT\ MAN + FD$			
$ALT\ MAN$	$\gamma \to 0$	δ_t	δ_e
$ALT\ MAN + FD$			
$CON2\ ALT$	$V \to 10$	δ_t	δ_e
$CON2\ ALT + FD$			
OOT	$\theta \to -3$	δ_e	δ_t
$OOT + FD$			
MAN	$\gamma \to 0$	—	$\delta_e,\ \delta_t$
$MAN + FD$			
$CON2\ MAN$	$V \to 10$	—	δ_e, δ_t
$CON2\ MAN + FD$			

$$y_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \end{bmatrix} x \tag{3.18}$$

The closed-loop dynamics result from a reference tracking feedback control law (3.19), where only two outputs are available for tracking at a time. The two outputs available for tracking in each mode are assumed to be the horizontal speed V and the output associated with the goal for each mode. The goal for each mode, indicated by the reference input $r \in R$, is defined in Table 3.3.

$$\begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix} = -K_q x + N_q r \tag{3.19}$$

AP FD A/THR	AP A/THR	FD A/THR	A/THR	FD	

Figure 3.6: Indicators on the the automated systems status section of the flight mode annunciator (FMA), part of the primary flight display (PFD) for flight crew [1]. The six possible indicator combinations shown correspond to the discrete outputs for modes of hybrid system H.

3.3.2 Accident Description

This accident scenario begins with the aircraft in level flight, which can be achieved in a number of different ways according to the FMS mode. However, the accident investigation report indicates that the aircraft was in $ALT\ HLD + FD$ mode at the beginning of the accident scenario [1]. $ALT\ HLD$ and $ALT\ HLD + FD$ modes maintain or hold an altitude set by the user. In this case, the aircraft was maintaining an altitude of 35,000 ft. About two hours into the flight, at least two of the the three Pitot probes, located on the front nose of the aircraft, became obstructed by ice, generating erroneous airspeed measurements. Because these airspeed measurements deviated significantly from one another and the third airspeed measurement, the pilot's flight management guidance and envelope computer (FMGEC) could not function [1]. This caused the autopilot and autothrust to disconnect automatically, sending the aircraft into the mode $CON2\ ALT + FD$ via the automatic transition γ_1. Since γ_1 is not annunicated, the flight crew were unaware that they had entered mode $CON2\ ALT + FD$.

Furthermore, the A-330 has fly-by-wire flight controls, meaning that the pilot's movement of the sidestick controller is converted into an electrical signal that is sent to the flight control primary computer (FCPC) [1]. This computer calculates the appropriate command to send to the actuators, such as the elevators, based on the pilot's orders. The mathematical relationship used to convert the pilot's orders into actuator commands is called a control law [1]. Under normal aircraft operation,

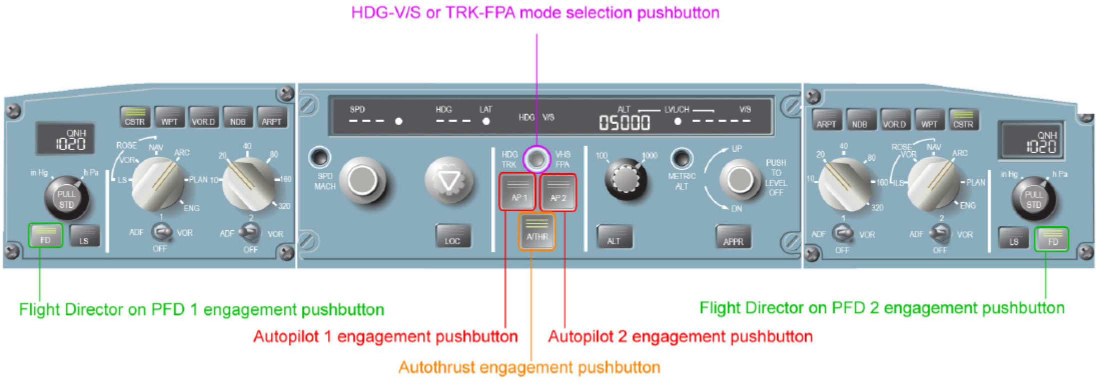
HDG-V/S or TRK-FPA mode selection pushbutton
QNH
PULL STD
CSTR
WPT
VOR.D
NDB
ARPT
ROSE
NAV
ARC
PLAN
ENG
ADF
VOR
OFF
FD
LS
SPD
HDG
LAT
V/S
ALT
LVL/CH
SPD MACH
TRK
FPA
AP 1
AP 2
A/THR
LOC
METRIC ALT
UP
DN
PUSH TO LEVEL OFF
APPR
Flight Director on PFD 1 engagement pushbutton
Flight Director on PFD 2 engagement pushbutton
Autopilot 1 engagement pushbutton
Autopilot 2 engagement pushbutton
Autothrust engagement pushbutton

Figure 3.7: Flight control unit (FCU) panel in the aircraft cockpit [1].

meaning that all system components are working properly and sensor readings are consistent, the “normal” control law is employed in the A-330 [1]. But when certain system components are malfunctioning or key sensor readings are inconsistent, the aircraft may adopt an “alternate” control law, in which some automated systems are disabled and control of some actuators is done directly, meaning that the actuators will move in direct proportion to the pilot’s orders. This direct control contrasts the normal operation of the aircraft in which the control surfaces move according to some transformation of the pilot’s orders.

During Flight 447, the high variation in the speed readings affected the flight control primary computers (FCPC), which caused a control law reconfiguration to “alternate 2” law. This control law reconfiguration corresponds to mode $CON2\ ALT{+}FD$ and caused the autothrust to become locked in the previous position until disabled manually by the crew and triggered a loss of high pitch and high angle of attack (AOA) protection. Loss of such safe flight envelope protection means that the aircraft will respond to pilot orders even if those orders cause the aircraft to assume an unsafe attitude, whereas “normal” law prevents actuators from responding to pilot orders that could cause the aircraft to exceed safe limits of operation. The modes for which the automation takes control of the elevator due to an unsafe aircraft attitude

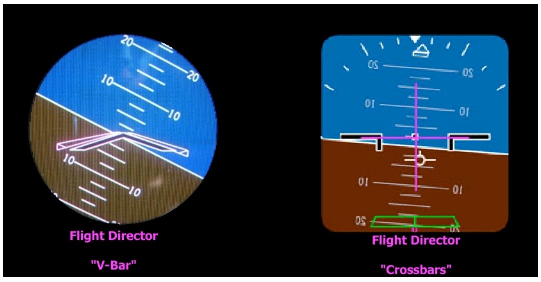

Figure 3.8: *Left:* Flight director "V" indicator for flight path angle. *Right:* Flight director crossbars on the primary flight display [28].

corresponds to modes $OOT\ ALT\ HLD$, $OOT\ ALT\ HLD + FD$, $OOT\ ALT\ MAN$, $OOT\ ALT\ MAN + FD$, OOT, and $OOT + FD$, where "OOT" stands for out-of-trim. and subsequent loss of flight envelope protections also correspond to the $ALT2\ THR + FD$ mode.

Despite the erroneous speed readings and malfunctioning flight computers, the pilot's FD was still engaged in the mode $CON2\ ALT + FD$. The FD is a computerized function in which the appropriate pitch and roll guides for a given maneuver are integrated into crossbar indicators, or a flight path angle guide is integrated into a "V" indicator, on the primary flight display (PFD) (see Figure 3.8). The FD is intended to help the pilot complete maneuvers smoothly and efficiently [1], [17]. However, the FCPC computes the correct position of the FD, so the FD was calculating orders based on erroneous data. The FD also became unavailable for short periods of time during the incident when all three speed measurements were invalid, but the flight crew still never turned it off [1].

The pilot did, however, disengage the autothrust, thereby discontinuing the thrust lock function, which corresponds to the mode $CON2\ MAN + FD$. The aircraft remained in this mode until the end of the incident without the flight crew's acknowledgment of an unsafe attitude. Due to the control law reconfiguration, transition into an out-of-trim mode, in which the automation would take over control of the elevator if the crew drove the aircraft too near the boundaries of the flight envelope, was impossible. Therefore, the crew caused the aircraft to stall while still under the impression that they were operating the aircraft safely.

3.3.3 Interface Analysis

These issues with mode confusion suggest that the FMS is not user-observable and user-predictable. We apply Algorithm 1 and Algorithm 2 to determine if, in fact, the system failed the user-observability and user-predictability requirements. First, we examine the initial mode $ALT\ HLD + FD$. $ALT\ HLD + FD$ fails to meet the second condition in line 2 of Algorithm 1 for $|Q_{AP+FD} \cap I^b_{\sigma_{AP}}| = |Q_{AP+FD} \cap I^b_{\sigma_{FD}}| = 2$. Progressing through Algorithm 1 to line 8 also reveals that the condition stated is not met for initial state $ALT\ HLD + FD$. However, the condition in line 9 is satisfied because the observability matrix associated with mode $ALT\ HLT + FD$ is full rank, which indicates that the continuous initial state can be determined if the initial mode is known.

Further analyzing the system using Algorithm 2 yields the following: the system fails the first condition in line 3 because the discrete output is the same for some modes separated by an unannunciated event, such as $h(ALT\ MAN + FD) = h(CON2\ ALT + FD)$ for unannunciated event γ_1. The system also fails the other conditions stated in line 3 of Algorithm 2. For example, $|Q_{A/THR} \cap I^f_{\sigma_{FD}}| = |Q_{A/THR} \cap I^b_{\sigma_{FD}}| = |Q_{A/THR}| = 3$ since $I^f_{\sigma_{FD}} = I^b_{\sigma_{FD}} = Q$. Thus, neither the current nor the next discrete mode can be determined from the discrete information alone. Applying line 9 of the algorithm to the system reveals that the discrete mode cannot be distinguished from the continuous output either.

For instance, the modes $ALT\ MAN + FD$ and $CON2\ ALT + FD$ cannot be distinguished from one another via discrete information, but the continuous output does not provide enough information to distinguish between them either. The continuous output and higher derivatives for mode $ALT\ MAN + FD$, where $q = ALT\ MAN + FD$ is given by

$$\mathcal{Y} = \mathcal{O}_q\ x + \Gamma_e N_{q,e}\ \mathcal{U} + \Gamma_t N_{q,t} \lambda \tag{3.20}$$

where λ represents the thrust reference input, which is controlled by the automation in this mode. Thus, λ is considered to be the unknown automation input. To eliminate this term, only one projection matrix is required. The continuous output and higher derivatives for mode $CON2\ ALT + FD$ has the same structure as that shown in (3.20), where $q = CON2\ ALT + FD$, since the automation also controls the thrust in $CON2\ ALT + FD$.

To distinguish mode $ALT\ MAN + FD$ from mode $CON2\ ALT + FD$, let $q = ALT\ MAN + FD$ and $q' = CON2\ ALT + FD$. The rank condition of Proposition 7 becomes to the following

$$\begin{array}{rcl} rank(P\mathcal{O}_q) & = & rank([P\mathcal{O}_q\ P(\mathcal{Y} - \Gamma_e N_{q,e}\ \mathcal{U})]) \wedge \\ rank(P\mathcal{O}_{q'}) & \neq & rank([P\mathcal{O}_{q'}\ P(\mathcal{Y} - \Gamma_e N_{q',e}\ \mathcal{U})]) \end{array} \tag{3.21}$$

for the projection matrix

$$P = \begin{bmatrix} -0.8674 & 0 & -0.0011 & 0.1173 & -0.4836 \\ 0.3896 & 0 & -0.0072 & 0.7170 & -0.5249 \\ -0.2972 & 0 & -0.0062 & 0.6023 & 0.6792 \\ 0.0030 & 0 & 5.9180e-05 & -0.0062 & -0.0068 \\ 0 & 1 & 0 & 0 & 0 \\ -4.4238e-04 & 0 & 0.9999 & 0.0072 & 1.6592e-04 \\ 0.0440 & 0 & 0.0072 & 0.2924 & -0.0080 \\ -0.0743 & 0 & -0.0015 & 0.1544 & 0.1707 \end{bmatrix}^T \tag{3.22}$$

However, the second part of condition (3.21) is not satisfied for this case, or when attempting to distinguish mode $ALT\ MAN + FD$ from $OOT\ ALT\ MAN + FD$ using the continuous output.

This result occurs for many combinations of current mode q and next possible mode q' for the system H. The results suggests that the discrete mode is nearly impossible to decipher via the continuous output for systems with many similar parameters and constant reference inputs.

Furthermore, Algorithms 1 and 2 may not capture the most interesting system phenomena. For instance, the most distinguishing characteristic of a given mode may be the higher derivatives of the input. Since the algorithms restrict the available information to that of the zeroth derivative of the input, they do not capture the most dynamic system behavior, which may aid in mode distinguishability.

Finally, a possible solution to the lack of information available to the human user for state reconstruction would be to include the automation input in the pilot display.

Chapter 4

Concluding Remarks

Methods for mode distinguishability via the continuous output were presented for linear hybrid systems with partially unknown input. Sufficient conditions for user-observability and user-predictability of linear hybrid systems with partially unknown input were also given, as well as algorithms detailing the procedure to check for user-observability and user-predictability of linear hybrid systems. These algorithms were then applied to two aircraft examples: one in which the interaction between the pilot, copilot, and the automation affected observability from the pilot's perspective, and another abstracted from Air France Flight 447.

The first of these examples demonstrated that the user-observability and user-predictability conditions are sufficient, but not necessary. In other words, very special cases may exist in which a system does not need to satisfy the user-observability and user-predictability conditions to ensure proper human-automation interaction, but such cases may never occur in reality due to the presence of disturbances, which is not considered here since only deterministic systems are studied. The final example illustrated that the system's continuous input may provide the information required to distinguish modes via the continuous output, but by eliminating unknown input terms, that information becomes unavailable for mode distinguishability. A solution to this lack of information contained in the output would be to include the automation input in the pilot display so that such information could be used for state reconstruction. Future work includes investigation of which system subsets are user-observable and user-predictable even though the entire system may not be.

References

[1] "Interim Report On the accident on 1st June 2009 to the Airbus A330-203 registered F-GZCP operated by Air France flight AF 447 Rio de Janeiro - Paris," *France, Tech. Rep.*, July 29 2011.

[2] P. Antasklis and A. Michel, *A Linear Systems Primer*,1st ed, Massachusetts: Birkhauser, 2007.

[3] K. Abbott, S. Slotte, and D. Stimson, "The interfaces between flightcrews and modern flight deck systems, *Federal Aviation Administration*, Human Factors Team Report, June 1996.

[4] A. Bryson, *Control of Spacecraft and Aircraft*, Princeton Univ. Press, 1994.

[5] C. Billings, "Human-Centered Aircraft Automation: Principles and Guidelines," *NASA Technical Memorandum 110381*, NASA-Ames Research Center, 1996.

[6] A. Balluchi, L. Benvenuti, M. Di Benedetto, and A. Sangiovanni-Vincentelli, "Design of observers for hybrid systems," *Hybrid systems: Computation and control*, Springer Berlin Heidelberg, pp. 76-89, 2002.

[7] M. Bolton, E. J. Bass, and R. Siminiceaunu, "Using Formal Verification to Evaluate Human-Automation Interaction: A Review," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 43, no. 3, pp. 488-503, 2013.

[8] W. Brinkman, D. Haggan, and W. Troutman, "A History of the Invention of the Transistor and Where It Will Lead Us," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 12, pp. 1858-1865, 1997.

[9] G. Basile and G. Marro, "Observability of linear, time-invariant systems with unknown inputs," *Journal of Optimization Theory and Applications*, vol. 3, no. 6, pp. 410-415, 1969.

[10] M. Babaali and G. Pappas, “Observability of switched linear systems in continuous time,” *International Workshop on Hybrid Systems: Computation and Control*, vol. 8, pp. 103-117, 2005.

[11] M. Campbell-Kelly and W. Aspray, *Computer: A History of the Information Machine*, New York: Basic Books, 1996.

[12] P. Collins and J. van Schuppen, “Observability of piecewise-affine hybrid systems,” *International Conference on Hybrid Systems: Computation and Control*, pp. 265-279, 2004.

[13] A. Degani and M. Heymann, “Formal verification of human-automation interaction,” *Human Factors*, vol. 44, no. 1, pp. 28-43, 2002.

[14] M. Endsley, “Toward a theory of situation awareness in dynamic systems,” *The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 32-64, 1995.

[15] N. Eskandari and M. Oishi, “Computing observable and predictable subspaces to evaluate user-interfaces of LTI systems under shared control,” *IEEE International Conference on Systems, Man, and Cybernetics*, 2011.

[16] P. Fitts, *Human engineering for an effective air navigation and traffic control system*, Columbus, OH: Ohio State University Research Foundation, 1951.

[17] *A330 & A340 Flight Crew Training Manual*, rev. 1, ed. 10, Cathay Pacific, June 2005.

[18] “About HFES,” *Human Factors and Ergonomics Society*, Tecra Systems, Inc., 2013, Website [Online], Available: http://www.hfes.org/web/abouthfes/about.html

[19] R. Hess, “Human-in-the-loop control,” *The Control Handbook*, W. Levine, Ed. CRC Press, Inc., pp. 1497-1505, 1996.

[20] G. Jamieson and K. Vicente, “Implications of a control-theoretic approach to human-automation-plant interface design,” *Symposium on Human Interaction with Complex Systems*, pp. 90-98, March 1998.

[21] S. Kaynama and M. Oishi, “Complexity reduction through a Schur-based decomposition for reachability analysis of linear time-invariant systems, *International Journal of Control*, vol. 84, no. 1, pp. 165-179, 2011.

[22] N. Leveson and E. Palmer, “Designing automation to reduce operator errors,” *Systems, Man, and Cybernetics. IEEE Conference on Computational Cybernetics and Simulation*, vol. 2, no. ,pp. 1144-1150, 1997.

[23] D. McRuer, “Human dynamics in man-machine systems,” *Automatica*, vol. 16, no. 3, pp. 237-253, 1980.

[24] N. Matni and M. Oishi, “Reachability-based abstraction for an aircraft landing under shared control,” *American Control Conference*, vol., no., pp. 2278-2284, June 2008.

[25] “Aircraft Accident Investigation Report 96-5 on 26th April 1994 to the Airbus A300-B4 registered 622R operated by China Airlines flight B1816 Taipei - Nagoya,” *Japan, Tech. Rep.*, July 19 1996.

[26] M. Oishi, I. Hwang, and C. Tomlin, “Immediate observability of discrete event systems with application to user-interface design,” *42nd IEEE Conference on Decision and Control*, vol. 3, 2003.

[27] R. Parasuraman, T. Sheridan, and C. Wickens, “A model for types and levels of human interaction with automation,” *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 30, no. 3, 2000.

[28] G. Rocha, “AF 447 - Flight Director with V-Bar Would Have Been Helpful,” *Aviation Troubleshooting*, Google, 2012. Web.

[29] W. Sweet, “The glass cockpit [flight deck automation],” *IEEE Spectrum*, vol. 32, no. 9, pp. 30-38, Sept. 1995.

[30] N. Sarter, D. Woods, and C. Billings, “Automation surprises,” *Handbook of Human Factors and Ergonomics*, NY: John Wiley and Sons, Inc., pp. 1295-1327, 1997.

[31] J. Tomayko, *Computers Take Flight: A History of NASA’s Pioneering Digital Fly-by-Wire Project*, Washington, D.C.: NASA, 2000.

[32] R. Vidal, A. Chiuso, S. Saotto, and S. Sastry, ”Observability of linear hybrid systems,” *Hybrid systems: Computation and control*, Springer Berlin Heidelberg, pp. 526-539, 2003.

[33] E. Wiener and R. Curry, “Flight-deck automation: promises and problems,” *Ergonomics*, vol. 23, no. 10, pp. 995-1011, 1980.