NASA/CR–2015-218702

Certification Considerations for Adaptive Systems

Siddhartha Bhattacharyya and Darren Cofer
Rockwell Collins, Inc., Cedar Rapids, Iowa

David J. Musliner, Joseph Mueller, and Eric Engstrom
Smart Information Flow Technologies, Minneapolis, Minnesota

March 2015

NASA/CR–2015-218702

Certification Considerations for Adaptive Systems

Siddhartha Bhattacharyya and Darren Cofer
Rockwell Collins, Inc., Cedar Rapids, Iowa

David J. Musliner, Joseph Mueller, and Eric Engstrom
Smart Information Flow Technologies, Minneapolis, Minnesota

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NNL13AC67T

March 2015

Acknowledgments

This work was supported by NASA under contract NNL13AC67T. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NASA or the U.S. Government. Several individuals contributed to the study described in this report. Siddhartha Bhattacharyya of Rockwell Collins provided general oversight, outline of the sections and characterization. Darren Cofer of Rockwell Collins developed sections on certification challenges and also on the general oversight. Dave Musliner, Joseph Mueller and Eric Engstrom provided insight into intelligent and adaptive control algorithms. Kelly J. Hayhurst from NASA Langley Research Center provided valuable input regarding adaptive systems and certification. Many researchers from academia, industry and government agencies provided insight into adaptive systems.

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA STI Program/Mail Stop 148
NASA Langlet Research Center
Hampton, Virginia 23681-2199
Fax: 757-864-6500

ABSTRACT

Advanced capabilities planned for the next generation of aircraft, including those that will operate within the Next Generation Air Transportation System (NextGen), will necessarily include complex new algorithms and non-traditional software elements. These aircraft will likely incorporate adaptive control algorithms that will provide enhanced safety, autonomy, and robustness during adverse conditions. Unmanned aircraft will operate alongside manned aircraft in the National Airspace (NAS), with intelligent software performing the high-level decision-making functions normally performed by human pilots. Even human-piloted aircraft will necessarily include more autonomy.

However, there are serious barriers to the deployment of new capabilities, especially for those based upon software including adaptive control (AC) and artificial intelligence (AI) algorithms. Current civil aviation certification processes are based on the idea that the correct behavior of a system must be completely specified and verified prior to operation. This report by Rockwell Collins and SIFT documents our comprehensive study of the state of the art in intelligent and adaptive algorithms for the civil aviation domain, categorizing the approaches used and identifying gaps and challenges associated with certification of each approach.

Contents

1 Introduction 1

1.1 Terminology 2

1.2 Background 6

2 Motivating Applications 8

2.1 Post-Stall Upset Recovery 9

2.2 Catastrophic Damage Landing 9

2.3 Autonomous Operation of UAVs 10

3 Certification Overview 12

3.1 Airworthiness Requirements 13

3.2 Certification Process 13

3.3 Safety Assessment 14

3.4 System Development 15

3.5 Software Assurance 16

3.6 Example Certified Control Application 18

4 Adaptive Control Algorithms 22

4.1 Introduction 22

4.2 Gain Scheduled Control 25

4.3 Indirect Adaptive Control 26

4.4 Direct Model Reference Adaptive Control 27

4.5 L1 Adaptive Control 29

4.6 Adaptive control with Neural Networks 31

4.7 Summary of Key Characteristics for Adaptive Control 32

5 Artificial Intelligence Algorithms 35

5.1 Introduction 35

5.2 Machine Learning 35

5.3 Neural Networks 37

5.4 Expert Systems 39

5.5 Fuzzy Logic 41

5.6 Cognitive Architectures 43
5.7 Planning and Scheduling 43
5.8 Computer Vision / Machine Vision 44
5.9 Qualitative Physics 47
5.10 Evolutionary algorithms 48
5.11 Natural Language Processing 49
5.12 Summary of Key Characteristics 50

6 Certification Challenges 53
6.1 Comprehensive requirements. 53
6.2 Verifiable requirements 53
6.3 Documented Design 55
6.4 Transparent Design 55
6.5 Summary 56

7 Mitigation Strategies 59
7.1 Education 59
7.2 Modified Certification Standards 60
7.3 New Certification Methods 60
7.4 New Verification Approaches 60
7.5 Architectural Mitigations with Certification Support 61
7.6 Paradigm Shift: Licensing 62

8 Roadmap and Future Directions 65
8.1 Related Workshops 65
8.2 Roadmap for Certification of Adaptive Systems 66

9 List of Acronyms 68

10 Glossary 69

11 References 70

A Background **78**

A.1 Recent History of Research Programs with Adaptive Flight Control 78

A.2 Previous Studies 80

B Application Areas in Civil Aviation **86**

B.1 Air Traffic Management 86

B.2 Route-Planning 87

B.3 Automatic Flight Control 88

B.4 Flight Management 88

B.5 Collision Avoidance 88

B.6 Fault Detection and Diagnostics 89

B.7 Multi-UAV mission 89

1 Introduction

Advanced capabilities planned for the next generation of aircraft, including those that will operate within the Next Generation Air Transportation System (NextGen), will necessarily include complex new algorithms and non-traditional software elements. These aircraft will likely incorporate adaptive control algorithms that will provide enhanced safety, autonomy, and robustness in the presence of failures and adverse flight conditions. NextGen systems will encompass both airborne and ground-based nodes with significant computational elements acting in coordination to maintain a safe and efficient airspace. Unmanned aircraft will operate alongside manned aircraft in the National Airspace (NAS), with intelligent software performing the high-level decision-making functions normally performed by human pilots. Even human-piloted aircraft will necessarily include more autonomy to achieve desirable characteristics such as flight into increasingly congested areas, airspace coordination with UAVs, and fuel- and time-optimized operations in free-flight.

However, there are serious barriers to the deployment of new capabilities, especially for those based upon software including adaptive control (AC) and artificial intelligence (AI) algorithms. Current civil aviation certification processes are based on the idea that the correct behavior of a system must be completely specified and verified prior to operation. While systems based on artificial intelligence and adaptive algorithms can be found in military and space flight applications, they have had only limited use in civil airspace due to the constraints and assumptions of traditional safety assurance methods. These barriers will delay or prevent the deployment of crucial safety functions and new capabilities that could be of great value to the public.

This report by Rockwell Collins and SIFT documents our comprehensive study of the state of the art in intelligent and adaptive algorithms for the civil aviation domain, categorizing the approaches used and identifying gaps and challenges associated with certification of each approach. The research effort involved understanding different adaptive control and artificial intelligence algorithms being applied to civil and military aviation. This required surveying published literature, as well as direct interactions with known experts in this field. We organized a workshop with researchers involved in developing adaptive control and artificial intelligence approaches, especially those focused on aerospace applications. The workshop helped us identify the spectrum of different approaches and algorithms, and characterize their features that are relevant to certification considerations.

In the remainder of this introduction, we discuss basic terminology and provide a summary of recent research programs and similar studies. Section 2 provides an overview of application areas within civil aviation that could benefit from the adaptive technologies in our review. Section 3 reviews the current approach to certifying software for use in these applications. Section 4 summarizes a wide spectrum of adaptive algorithms from the controls community, while Section 5 discusses algorithms and approaches that stem from the AI field. In each of these sections, we identify the unique characteristics of each algorithm that may pose a challenge to the current certification processes. Section 6 summarizes those characteristics and associated certification challenges. Section 7 presents a set of suggested evolutionary changes to the certification processes that will enable adaptive technologies to be certified and deployed with confidence. We conclude in Section 8 with a roadmap summarizing how different adaptive system approaches may be gradually incorporated into civil aviation applications.

1.1 Terminology

This report focuses on two broad categories of advanced technologies that may be useful in civil aviation: adaptive control systems and AI-based methods. In this section we briefly define these terms; however, distinguishing between them is not strictly important to the intent of this survey. The main point of the survey is to identify challenges to verifying and certifying advanced technologies, and suggest a roadmap to overcoming those challenges.

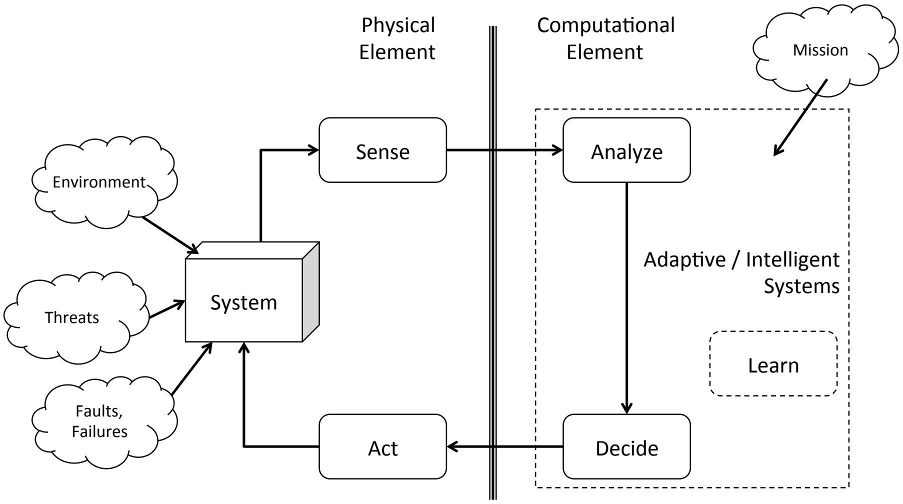

Figure 1: The roles of an adaptive intelligent system in closed-loop control.

Consider the active feedback loop shown in Figure 1. The four main steps in the loop are 1) sense, 2) analyze, 3) decide, and 4) act [82]. For most aviation applications, the “action” is applied to a system that we can describe reasonably well when it is operating nominally. If we have a good model of the system, then we can use that model along with the observations we have made from analyzing our sensory input to make a decision. However, it is very difficult to develop a model that fully captures all of the ways that a system may behave. The characteristics of the system may change significantly in response to changes in the mission, environment, threats, or failures— all of which are uncertain.

Traditionally, the computational element designed to control the system is fixed. In other words, the instructions that implement the “analyze and decide” steps are developed in some software which is inherently deterministic, and that software is subsequently certified relative to the original system requirements.

Adaptive System: A system in which the computational element of the active feedback process changes in order to maintain desired performance in response to failures, threats, or a changing environment.

With the introduction of adaptive controls or AI-based methods, the instructions that implement the “analyze and decide” steps are not fixed, but rather update over time in response to the system behavior. In general, when intelligence is added to the computational element, it 1) observes how it interacts with the system, 2) learns from those observations, and 3) updates itself to improve its interaction. This variability is the primary distinguishing feature that enables adaptive and intelligent methods to outperform their static counterparts. It is also the root cause of the certification challenges, as following subsections elaborate.

1.1.1 Definitions of Adaptive Control

Adaptive Control (AC): A control policy with 1) parameters that may be adjusted, and 2) some mechanism for automatically adjusting those parameters online based on measured performance.

Research in adaptive control dates back to the 1950’s, as flight control engineers worked to design automatic control policies for a new breed of experimental aircraft, most notably the X-15 [5]. As these new aircraft were capable of reaching higher altitudes and faster speeds, they stretched the flight envelope beyond the point where a single control policy could be used throughout. This experimental work sparked research in adaptive control, where the controller gains adapt in some way to the changing flight condition. The most practical solution was to simply pre-compute the controller gains over a grid of flight conditions, and update them online using a table-lookup. This practice is called gain-scheduled control, and has been widely used in flight controls as well as the broader controls community for the last several decades.

In general, adaptive control methods involve two basic pieces: 1) a control policy with parameters that may be adjusted, and 2) some mechanism for adjusting those parameters. Continued research in adaptive control theory has led to a rich mathematical framework to support the design and analysis of adaptive control algorithms. Several distinct methods are discussed in Section 4.

1.1.2 Definitions of Artificial Intelligence

Traditionally, the definition of AI has centered around the goal of computers emulating human intelligence. In 1981, Barr and Feigenbaum [6] defined AI as: "the part of computer science concerned with designing intelligent computer systems, that is, systems that exhibit the characteristics we associate with intelligence in human behavior – understanding language, learning, reasoning, solving problems, and so on." In 1983, Elaine Rich [61] described the field of AI as "the study of how to make computers do things at which, at the moment, people are better."

In 1998, Poole, Mackworth and Goebl provide a more useful definition, adopting the term *computational intelligence*: [60]

> Computational intelligence is the study of the design of intelligent agents. An agent is something that acts in an environment— it does something. Agents include worms, dogs, thermostats, airplanes, humans, organizations, and society. An intelligent agent is a system that acts intelligently: What it does is appropriate for its circumstances and its goal, it is flexible to changing environments and changing goals, it learns from experience, and it makes appropriate choices given perceptual limitations and finite computation.

For the purpose of this report, we define AI as a technological umbrella covering the broad class of methods that ultimately support the operation of an intelligent agent, as described above.

Artificial Intelligence (AI): A broad class of computational methods that are designed to operate with intelligence, primarily by 1) learning from experience, and 2) making decisions based on learned information to achieve a goal.

This notion of intelligence is appropriate as we consider the utility of AI in aviation, as well as a roadmap to certification. In particular, we are interested in the capabilities of 1) being flexible, or adaptive, to changes in the goals or environment, and 2) learning from experience. These two characteristics, **learning** and **adaptation**, are the two main ingredients for both AC and AI methods, providing the functional basis for enhancing performance and robustness over traditional methods. In short, a system that can learn from and adapt to its environment may be more capable than one that cannot.

The unique attributes of an intelligent, adaptive system are also evident when we examine the difference between automation and autonomy. The Air Force Research Laboratory uses the following definitions: [4]

Automation: The system functions with no/little human operator involvement, however the system performance is limited to the specific actions it has been designed to do. Typically these are well-defined tasks that have predetermined responses, i.e. simple rule-based responses."

Autonomy: Systems which have a set of intelligence-based capabilities that allow it to respond to situations that were not pre-programmed or anticipated in the design (i.e. decision-based responses). Autonomous systems have a degree of self-government and self-directed behavior (with the human's proxy for decisions).

Clearly, automation exists in various forms in civil aviation already. On-board instrumentation and GPS automatically provide navigation; the control system automatically tracks the pilot's commands; and the auto-pilot automatically guides the aircraft to its next waypoint. These functions are not intelligent, but rather consist of completely repetitive, pre-scripted responses that result in bounded and predictable behavior.

While the above characteristics are amenable to certification, they also constrain the performance and therefore the safety of the aircraft, because they prevent the overall system from continuing to function as desired when it departs from its nominal design conditions. Increasing adaptability, autonomy, and intelligence is intended to enhance performance and safety by enabling desired functionality to be maintained over a broader set of operating conditions.

1.1.3 Nondeterminism

One other term that we should deal with at the outset is *nondeterminism*. Adaptive systems are sometimes characterized as being nondeterministic. Critics may do this as a way to dismiss them as being impractical, unsafe, or impossible to certify. However, this is an incorrect generalization. There can be nondeterministic aspects to certain adaptive algorithms, but we need to be precise about the mechanism involved and whether or not it impacts safety and certification considerations.

There is no explicit requirement for determinism in current certification standards. The only explicit mention of determinism found in DO-178B was a requirement in Section 12 that "only deterministic tools may be qualified; that is, tools which produce the same output for the same input data when operating in the same environment." However, determinism is certainly a desirable characteristic for verification, and is assumed in current standards and processes. We have identified the following types of nondeterminism that may be relevant for our discussion:

Environmental nondeterminism— Most systems that take input from the outside world and make decisions are inherently dealing with nondeterministic data: we may be able to bound the input values and predict their behavior with mathematical models, but we cannot know in advance all of the possible sequences of inputs. For example, any control system that includes an integrator (e.g., a PID controller) may be considered nondeterministic at some level, because the internal state of the algorithm depends on its past inputs from the environment. The internal state of the integrator may reach very different values for very small changes in the input signals. Rigorous control design techniques can ensure that these issues are not a concern in practice. However, other methods such as neural networks may not be able to guarantee convergence or other desirable properties in the face of environmental nondeterminism.

Probabilistic algorithms— This includes algorithms that are based on sampling a random process or probability distribution. Mathematical techniques to bound the behavior of these algorithms and prove their convergence would be necessary if they were to be used in a certified system.

Uncertain existence of solutions— The existence of a solution to a problem may be unknown, or the algorithm may fail to find a solution within a fixed amount of time. Many planning and optimization algorithms fall into this category. For high-confidence applications, they may be used in conjunction with alternative mechanisms to guarantee the existence of a viable (though suboptimal) solution within a given deadline.

Concurrency— Multi-threaded computations where execution order impacts the result can lead to nondeterministic outputs. Multi-threaded computations should either be proven to be invariant to reordering, or synchronization mechanisms should be used to constraint the ordering as needed.

1.2 Background

The field of adaptive control has matured over the last 50+ years, with important contributions from a large number of researchers. Several different research programs were active in the 1990's, including the AFRL Self-Designing Controller [84], and the Boeing-led RESTORE [85] and JDAM [68] programs. These programs culminated in successful flight tests of indirect adaptive control on the NASA X-36 tailless aircraft, and multiple forms of adaptive augmentation on the JDAM missile platform. More recently, the NASA DACS and IRAC projects developed various adaptive control methodologies to provide onboard control resilience in the face of damage and other adverse flight conditions [9, 24]. More details on these modern research programs are provided in Appendix A.

As the core technologies for adaptive and intelligent flight control have steadily matured, focus has recently began to shift toward the topic of certification [27, 74, 82]. At the same time, general interest in using unmanned aerial vehicles (UAVs) has grown rapidly, extending beyond the military domain to countless applications in the civil, scientific and commercial sectors. The autonomous operation of UAVs will require intelligent methods to ensure safe and efficient flight, and so the topics of autonomous and intelligent systems go hand in hand. Within the last year, multiple workshops have been sponsored by AFRL on the test, evaluation, verification and validation of autonomous systems [4], and the National Research Council has issued a report on autonomy research for civil aviation [55].

All of the recent studies have underscored a few common points. It is widely recognized that existing certification criteria and processes do not properly account for the unique characteristics of adaptive, intelligent methods. Non-determinism and learning behavior are seen as presenting the greatest obstacles to certification. More specifically, the verification and validation process is faced with technical challenges, primarily due to the inherent difficulty in generating test-cases that provide full coverage.

In this report, our goal is to first identify the unique characteristics present in various types of adaptive control and AI methods, and then to examine why these characteristics lead to certification challenges. With this understanding in place, we provide a set of potential new methods for certification, and a suggested roadmap for progressive implementation. Our study validates the findings of the FAA report [82] conducted by Honeywell and NASA Langley, along with broadening the scope of the adaptive systems considered to include a more detailed survey of AI methods. Additionally, our report discusses the following new insights regarding certification of adaptive systems:

- We provide a mapping of the different adaptive methods to related certification challenges.

- We discuss the need for advanced verification approaches for adaptive systems.

- We provide a mapping of the adaptive methods and respective potential applications to the associated software certification level that may be required.

- We present a roadmap for adaptive technologies with categories defining the necessary changes in certification processes.

2 Motivating Applications

Technological advances in aviation, as with most applications, are driven by a few fundamental goals: improve safety, enhance performance, and reduce cost. In pursuit of these ideals over the past century, the overall design of the airplane, its systems, and how we regulate its flight have experienced a significant evolution. Progress has been made on numerous fronts, including stronger and lighter materials, more efficient engines, and enhanced aerodynamics. Advances in automatic control, in particular, have vastly improved safety by ensuring the aircraft remains stable and responsive to pilot commands in the face of atmospheric disturbances and imperfect sensor measurements.

As the aviation industry continues to evolve today, many have recognized the potential of adaptive and intelligent systems to further improve various aspects of aircraft safety and performance. A 1994 FAA study on AI in aviation [27] cited several areas that could potentially benefit from the application of intelligent systems, including: support for emergency procedures, navigation, diversion planning, diagnostics and monitoring. Another recent report written by NASA Langley and Honeywell for the FAA [82] identified planning, monitoring, parameter estimation and control reconfiguration as important application areas for adaptive technologies.

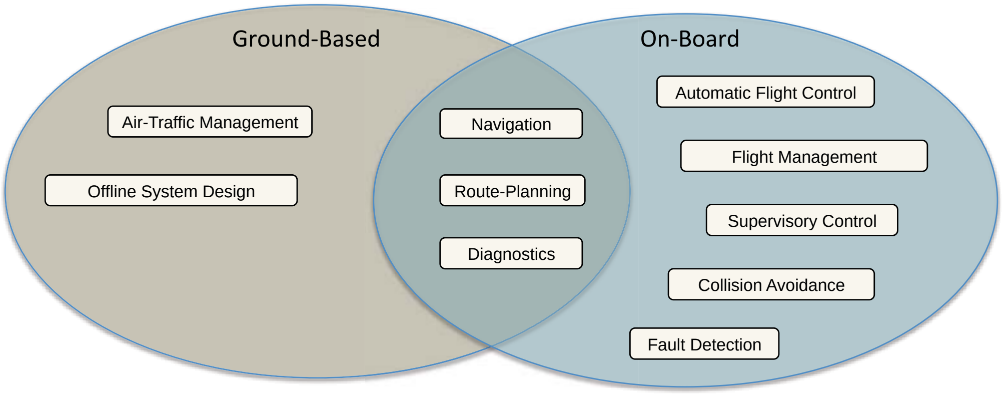

Figure 2: Applications for adaptive and intelligent systems in civil aviation.

Figure 2 provides a broad summary of civil aviation applications that may benefit from adaptive and intelligent software. The application areas listed here are grouped into ground-based and on-board applications. This is an important distinction with regards to certification, as ground-based software is subjected to a unique set of certification guidance [36]. Applications that are completely ground-based include air traffic management, which is administered by a network of ground-based computer systems and personnel, as well as offline control system design. Applications that are completely on-board include those tasks that must occur immediately: automatic flight control, flight management, supervisory control, collision avoidance, and fault detection. Other applications may be implemented on-board or offloaded to a ground-based system as appropriate. This includes navigation, route-planning, and diagnostics.

In the sections that follow, we discuss a few specific motivating applications within civil aviation where the unique capabilities of adaptive and intelligent methods can provide important benefits.

2.1 Post-Stall Upset Recovery

One of the most common dangers in civil aviation is the loss of control due to some form of aircraft upset. This occurs when the aircraft enters into an extreme and dangerous flight condition, which could be the result of a system failure, extreme turbulence, or pilot error. An example is when the aircraft stalls, a condition in which the aerodynamic lift suddenly drops after the angle of attack grows too high. Loss of control immediately follows, as the aerodynamic control surfaces lose their effectiveness. Recovering from stall is a challenging task for pilots, particularly because the aircraft is nearly unresponsive during stall, and then responds much differently than normal at the onset of recovery. While conventional flight control systems are designed to provide desired handling qualities and robust control over the operational flight envelope, they cannot cope with the highly nonlinear and uncertain aerodynamic properties that occur at high angles of attack.

Adaptive control methods are well-suited for upset recovery. Rather than relying on a static, pre-designed control law based on a poor model of the aerodynamics, the adaptive control system changes dynamically based on the measured response, effectively tuning itself. Various types of adaptive control have been flight-tested on the NASA AirSTAR sub-scale transport aircraft [9, 24]. Adaptive control methods are discussed in Section 4.

2.2 Catastrophic Damage Landing

In the upset recovery example described above, the aircraft is only at risk while it is flying outside of the safe operating envelope. Once control is regained, routine flight may proceed. In the event of catastrophic damage or failure, however, the aircraft remains in danger as long as it is in the air, and the goal is to reach a safe landing as quickly as possible.

A prime example of catastrophic damage landing occurred in January of 2009, when US Airways Flight 1549 struck a flock of geese, lost both of its twin jet engines, and landed safely in the Hudson River. In this case, the successful landing was attributed to a well-trained and experienced pilot who recognized the safest possible location to land. With both engines out, the aircraft was far less maneuverable but it remained stable and provided sufficient control authority for the pilot to turn and manage the rate of descent.

In more extreme damage events, such as the partial loss of a wing, or other severe structural damage, the aircraft dynamics can change radically. In cases like these, the aerodynamic properties resulting from the damage can be highly nonlinear and extremely difficult to model. Even if a precise model could be made for each damage event, this approach is impractical due to the sheer number of different models that would be required, as well as the challenge of identifying in real-time which model is the correct one to use. Therefore, a control system that can automatically adapt to the abrupt change in dynamics may be the only way to restore control and save the aircraft.

Rockwell Collins Control Technologies has developed an automatic supervisory adaptive control architecture to enable this type of damage tolerant control [39].

2.3 Autonomous Operation of UAVs

Operation of remotely piloted vehicles by hobbyists, universities, and businesses has soared over the last several years. Quad rotors in particular have become a popular platform for may applications due in part to their relative ease of operation and their ability to hover and roam at low speed. Current FAA regulations severely restrict the use of unmanned vehicles in controlled airspace, but the inherent market incentives for using them will likely lead to a relaxation of these rules. As an example, both Amazon and Google have recently announced their plans to use unmanned "drones" for automated package delivery. Although the reality of these visions may still lie far into the future, they do give evidence to the potential economic benefits of UAVs.

If autonomous aerial vehicles are to be integrated safely into the airspace, several technical and regulatory challenges must be worked out. Both visual flight rules (VFR) and instrument flight rules (IFR) govern the appropriate actions of pilots in different classes of airspace. One of the key responsibilities of a pilot is to see and avoid other aircraft, ensuring that they do not endanger their safe flight. The same rules will extend to the autonomous control system flying the UAV. Therefore, onboard hardware and software enabling a "see and avoid" capability is likely to be a requirement for UAVs operating in regulated airspace.

Of the many potential applications, agricultural monitoring appears to be the most attractive place to start. There are clear market incentives for farmers, who could use light UAVs as a low-cost tool for quickly monitoring crop health over a large area. In addition, flying over private property in unpopulated rural areas has added benefits, eliminating privacy issues and reducing the safety implications. One remaining concern is the need to bound the flight of the UAV over a designated area. The concept of “geo-fencing” has been used on remotely piloted vehicles for automatic flight termination, and is a candidate solution to safely bound the flight of autonomous UAVs. Extending autonomous flights into urban areas is of course another major area of interest due to the broader market potential, but it introduces more challenges. In particular, safe landing paths must be identified ahead of time to ensure vehicles can safely terminate flight from any point in their trajectory [54].

3 Certification Overview

In this section we provide an overview of the current certification process for civil aircraft. As we later consider the characteristics of adaptive and intelligent algorithms, this will provide the background for understanding the challenges that the certification process might present for deployment of these algorithms.

Certification is defined in [53] as follows:

> **Certification:** Legal recognition by the certification authority that a product, service, organization or person complies with the requirements. Such certification comprises the activity of technically checking the product, service, organization or person and the formal recognition of compliance with the applicable requirements by issue of a certificate, license, approval or other documents as required by national laws and procedures.

The requirements referred to in this definition are the government regulations regarding the airworthiness of aircraft in the NAS. Note that software itself is not certified in isolation, but only as part of an aircraft.

This definition goes on to say that certification of a product (i.e., an airplane) involves: (a) the process of assessing the design of a product to ensure that it complies with a set of standards applicable to that type of product so as to demonstrate an acceptable level of safety; (b) the process of assessing an individual product to ensure that it conforms with the certified type design; (c) the issuance of a certificate required by national laws to declare that compliance or conformity has been found with standards in accordance with items (a) or (b) above.

In the context of commercial aircraft, *type certification* is legal recognition by the relevant certification authority (the FAA in the U.S. or EASA in Europe) that an aircraft design complies with specified governmental regulations. In practice, certification consists primarily of convincing representatives of a government agency that all required steps have been taken to ensure the safety, reliability, and integrity of the aircraft.

Certification differs from verification in that it focuses on evidence provided to a third party to demonstrate that the required activities were performed completely and correctly, rather on performance of the activities themselves. Also note that certification connects a product or design to legal requirements for its safety. Therefore, it is possible for a design to be safe but not certifiable if it is not possible to produce the type of evidence required by the certification process or if the certification authority is for some reason not convinced of the adequacy of the evidence provided.

Military aircraft fall under an entirely different legal framework. Civil aircraft certification differs significantly from military airworthiness certification due to differences in public safety expectations, operational requirements, and procurement processes. Airworthiness of military aircraft is defined in MIL-HDBK-516B. Military aircraft certification is part of the procurement process, since the military branch that is buying the aircraft is also responsible for certifying its airworthiness as part of accepting the product. This partially explains why some adaptive and intelligent algorithms have been deployed in military aircraft for many years, but not commercial aircraft.

3.1 Airworthiness Requirements

In the U.S., the legal requirements for aircraft operating in the NAS are defined in the Code of Federal Regulations, Title 14 (14CFR), Aeronautics and Space. The purpose of certification is to ensure that these legal requirements have been met.

Airworthiness standards for transport class aircraft are specified in Part 25 and standards for smaller aircraft are specified in Part 23. Parts 27 and 29 apply to rotorcraft and Part 33 to engines. Part 25 covers topics including Flight, Structure, Design and Construction, Powerplant, Equipment, Operating Limitations, and Electrical Wiring. Some of the requirements are quite detailed. For example, Subpart B (Flight) provides formulas and a detailed procedure for computing reference stall speed. It also provides requirements for controllability, trim conditions, and stability. Subpart D (Design and Construction) includes requirements for Control Systems related to stability augmentation, trim systems, and limit load static tests. Some requirements cover items that no longer apply to modern aircraft (cables and pulleys).

The important observation here is that any changes to the certification process that we may eventually want to consider to facilitate deployment of adaptive or intelligent systems must still ensure compliance with 14CFR. A detailed review of these requirements will be necessary to be sure that there are no legal barriers to deployment of adaptive or intelligent systems, in addition to any barriers related to the current certification process.

3.2 Certification Process

The stakeholders in the civil aviation domain (FAA, airframers, equipment manufacturers) have developed a collection of documents defining a certification process which has been accepted as the standard means to comply with federal regulations. The process includes system development, safety assessment, and design assurance. These documents and their relationships are shown in Figure 3.

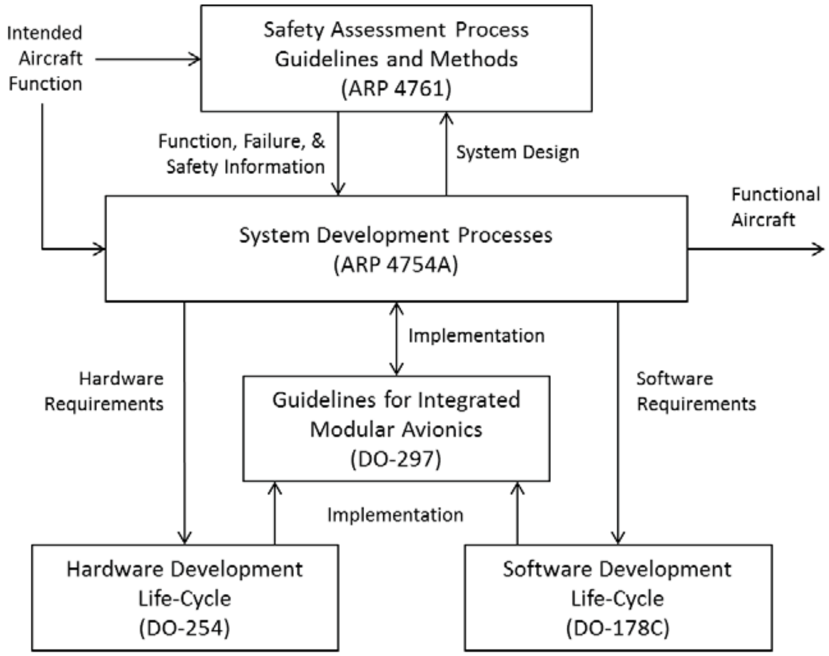
Intended Aircraft Function
Safety Assessment Process Guidelines and Methods (ARP 4761)
Function, Failure, & Safety Information
System Design
System Development Processes (ARP 4754A)
Functional Aircraft
Implementation
Hardware Requirements
Guidelines for Integrated Modular Avionics (DO-297)
Software Requirements
Implementation
Hardware Development Life-Cycle (DO-254)
Software Development Life-Cycle (DO-178C)

Figure 3: Relationship among key documents in the certification process

The intended function, or requirements, for a new aircraft are the starting point for the process. These requirements are the basis for the aircraft system design that is produced in accordance with ARP4754A, the guidelines for the system development process. The system design along with the aircraft requirements and its operating context are used to conduct a safety assessment in accordance with ARP4761.

The safety assessment determines, among other things, the criticality of system components as they contribute to the safety of the overall system. The system development process allocates functions and requirements to hardware and software components in the system, along with their assigned criticality from the safety assessment process. This information is used to develop the individual components and functions. The design assurance documents DO-178C (for software), DO-254 (for programmable hardware), and DO-297 (for integrated modular avionics) provide guidance for ensuring that these components satisfy the requirements that come from the system development process.

3.3 Safety Assessment

Safety assessment is performed in accordance with ARP4761, Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and

Equipment. This document describes guidelines and methods of performing the safety assessment for certification of civil aircraft, and is a means of showing compliance with the safety requirements of 14CFR. These requirements are hidden in Subpart F (Equipment) section 25.1309 with the unlikely title "Equipment, systems, and installations."

This section states that the equipment, systems, and installations required in an aircraft must be designed to ensure that they perform their intended functions under any foreseeable operating condition. The airplane systems and associated components, considered separately and in relation to other systems, must be designed so that:

- The occurrence of any failure condition which would prevent the continued safe flight and landing of the airplane is extremely improbable, and
- The occurrence of any other failure conditions which would reduce the capability of the airplane or the ability of the crew to cope with adverse operating conditions is improbable.

The section goes on to state that warning information must be provided to alert the crew to unsafe system operating conditions, and that systems, controls, and associated monitoring and warning means must be designed to minimize crew errors which could create additional hazards. Compliance must be shown by analysis or testing that considers possible modes of failure (including malfunctions and damage from external sources), the probability of multiple failures and undetected failures, the resulting effects on the airplane and occupants, and the crew warning cues, corrective action required, and the capability of detecting faults.

It should be obvious, but it is worth emphasizing that any aircraft design that contains adaptive or intelligent systems will include a safety assessment that identifies requirements for those systems and determines how they must function to provide for safe operation of the aircraft. This must include the ability to function in the presence of failures, and without creating conditions that could cause crew errors and introduce additional hazards. Any system that cannot provide the functionality required by the safety analysis must not be deployed in a civil aircraft, and barriers to certification in this instance are appropriate.

3.4 System Development

Aircraft system development is described in ARP4754A, "Guidelines for Development of Civil Aircraft and Systems." This document discusses the development of aircraft systems, taking into account the overall aircraft operating environment and functions. This includes validation of requirements and verification of the design implementation for certification and product assurance. It provides practices for showing compliance with the regulations.

ARP4754A provides guidance for creating plans for the system development, and eight *integral processes* which span all of the system development activities. These include safety assessment, assurance level assignment, requirements capture, requirements validation, implementation verification, configuration management, process assurance, and certification and regulatory authority coordination. The system development process allocates functionality and defines requirements for components, both hardware and software. It invokes the safety assessment process ensures that the system design satisfies safety requirements for the aircraft, and guides developers in allocating system requirements to hardware and software components and in determining the criticality level for those components.

An adaptive or intelligent system needs to be able to provide the level of safety required by the system design. It must be defined in terms of a complete set of system-level requirements that can be implemented and verified by the software assurance process.

3.5 Software Assurance

The software assurance process makes sure that components are developed and verified to meet their requirements *without any unintended functionality.* This means that the process will include activities specifically designed to provide evidence that the software does only what its requirements specify, and nothing else.

For software in commercial aircraft, the relevant guidance is found in DO-178C, "Software Considerations in Airborne Systems and Equipment Certification." Certification authorities in North American and Europe have agreed that an applicant (aircraft manufacturer) can use this guidance as a means of compliance with the regulations governing aircraft certification.

The original version of the document, DO-178, was approved in 1982 and consisted largely of a description of best practices for software development. It was revised in 1985 as DO-178A, adding definitions of three levels of software criticality, with development and verification processes described in more detail. DO-178B, approved in 1992, defined five levels of software criticality (A – E) with specific objectives, activities, and evidence required for each level. The processes and objectives in the document assume a traditional development process with test-based verification.

In 2005, the publishers of DO-178 initiated work on a revision to be known as DO-178C. A committee was chartered to draft the new document, with the objectives of minimizing changes to the core document, yet updating it to accommodate approximately 15 years of progress in software engineering. Guidance specific to new software technologies was to be contained in supplements which could add, modify, or replace objectives in the core document. New supplements were developed in the areas of object oriented design, model-based development, and formal methods, as well as an additional document containing new guidance on tool qualification. DO-178C and its associated documents were published in 2011 and accepted by the FAA as a means of compliance in 2013.

As a consequence if its latest revisions, DO-178C makes greater allowance for, and emphasis on, using analysis and model-based development techniques for software development. This should help with certification of aircraft that include adaptive and intelligent software algorithms.

3.5.1 Software Development

DO-178C does not prescribe a specific development process, but instead identifies important activities and design considerations throughout a development process, and defines objectives for each of these activities. It assumes a traditional development process that can be decomposed as follows:

- Software Requirements Process. Develops High Level Requirements (HLR) from the output of the system design process.
- Software Design Process. Develops Low Level Requirements (LLR) and Software Architecture from the HLR.
- Software Coding Process. Develops source code from the software architecture and the LLR.
- Software Integration Process. Combines executable object code modules with the target hardware for hardware/software integration.

Each of these processes produces or updates a collection of lifecycle data items, culminating in an integrated executable.

3.5.2 Software Verification

The results of these processes are verified through the verification process. The verification process consists of review, analysis, and test activities that must provide evidence of the correctness of the development activities.

In general, verification has two complementary objectives. One objective is to demonstrate that the software satisfies its requirements. The second objective is to demonstrate with a high degree of confidence that errors which could lead to unacceptable failure conditions, as determined by the system safety assessment process, have been removed.

One of the foundational principles of DO-178C is requirements-based testing. This means that the verification activities are centered around explicit demonstration that each requirement has been met.

A second principle is complete coverage, both of the requirements and of the code that implements them. This means that every requirement and every line of code will be examined in the verification process. Furthermore, several metrics are defined which specify the degree of coverage that must be obtained in the verification process, depending on the criticality of the software being verified.

A third principle is traceability among all of the life cycle data items produced in the development process. This means that:

- Every requirement must have one or more associated test cases. All testing must trace to a specific requirement.
- Every requirement must be traceable to code that implements it. Every line of code must be traceable to a requirement.
- Every line of code (and, in some cases, every branch and condition in the code) must be exercised by a test case.

Together, these objectives provide evidence that all requirements are correctly implemented and that no unintended function has been implemented. Note that the Formal Methods Supplement (DO-333) generally allows the testing described above to be replaced by a comparable formal analysis. However, even when formal methods are used some on-target testing is required.

One constraint imposed by ARP4754A (and DO-178C) is that requirements must be *verifiable*, which in the past has meant *testable*. This meant that in practice there could be no negative requirements such those related to safety (e.g., the system can never enter an unsafe state). However, such requirements can be addressed analytically and may be very useful in demonstrating the safety of an adaptive or intelligent algorithm.

In summary, under the current standards, any adaptive or intelligent system will need to have explicit, verifiable requirements that completely specify its behavior, and for which every line of the code implementing it can be traced back to those requirements through an appropriate coverage metric.

3.6 Example Certified Control Application

To give an example of how requirements are presently defined for existing control systems, this section describes the requirements for a typical certified autopilot system. The following are the requirements listed:

- Identify all the system inputs from other systems or subsystems: Roll, Pitch, Airspeed, Pilot Commands, Aileron commands.
- Evaluate validity of the system by monitoring Autopilot data: Identify conditions under which the Autopilot is invalid.
- Indicate how the system is activated: Identify the signal/condition that leads to activating the system.
- Resolve conflict about which Flight Guidance System is in control: Identify command source, Roll command data, Pitch command data.
- How to engage and disengage the autopilot: How does the pilot do it.
- Copilot trying to override autopilot on ground.
- Takeoff with Autopilot engaged.
- Autopilot Engagement: Conditions when the autopilot is engaged.
- Autopilot Disengagement: Conditions when the autopilot is disengaged.
- Autopilot fail: Indicate autopilot failed when the identified conditions are met.
- Autopilot monitor: Trigger the monitor when performance is outside the envelope.
- Autopilot Limiter/ Interface function: Autopilot controllers designed to interface with the Flight Guidance system and the Autoland systems. It provides the required attitude roll and pitch.
- Autopilot Pitch Interface:
  - Pitch Interface shall control aircraft pitch attitude in response to the pitch command input.
  - Autopilot pitch damping: The requirement is to meet a narrow fixed control bandwidth, by applying damping on the error signal.
  - Autopilot Pitch Command Selection and Limiting: The autopilot response must be relatively stable and robust to variation of other components it interacts with. Autopilot response must meet acceptable ride quality and passenger comfort characteristics. Transition characteristics between engaging and disengaging the Autopilot should be smooth and limited small signal oscillatory modes that result from non-linear elements in the aircraft control systems.
  - Autopilot pitch forward gain: Control response changes in response to airspeed and aircraft control configuration. The requirement is to meet a narrow fixed control bandwidth by applying forward gain on the error signal.
  - Autopilot Pitch Command Fade: Autopilot shall fade command when conditions are met about selection of the pitch.
  - Autopilot pitch performance: The autopilot performance requirements are based on achieving a desired aircraft response to step inputs that

ensure that the control bandwidth and accuracy of the pitch attitude controller is sufficient to support the outer loop control performance requirements.

- Autopilot Roll Interface:
  - Roll Interface shall control aircraft roll attitude in response to the roll command input.
  - Autopilot roll damping: The requirement is to meet a narrow fixed control bandwidth, by applying damping on the error signal.
  - Autopilot Pitch Command Selection and Limiting: The autopilot response must be relatively stable and robust to variation of other components it interacts with. Autopilot response must meet acceptable ride quality and passenger comfort characteristics. Transition characteristics between engaging and disengaging the Autopilot should be smooth and limited small signal oscillatory modes that result from non-linear elements in the aircraft control systems.
  - Autopilot roll forward gain: Control response changes in response to airspeed and aircraft control configuration. The requirement is to meet a narrow fixed control bandwidth by applying forward gain on the error signal.
  - Autopilot roll performance: The autopilot performance requirements are based on achieving a desired aircraft response to step inputs that ensure that the control bandwidth and accuracy of the roll attitude controller is sufficient to support the outer loop control performance requirements. Amount of roll, prevent oscillation.
- Autopilot Pitch Command Fade: Autopilot shall fade command when conditions are met about selection of the pitch.
- Autopilot output signals.

This list indicates the activities and requirements in a typical control application. The most important activity is establishing the requirements, as the designed system needs to be traced back to these requirements. These requirements, are to some extent, linked to the safety analysis for the system. Some requirements are linked to airworthiness requirements in 14CFR. However, many are based on experience and best practices, drawing on the expertise of control engineers. Adaptive and intelligent software can be certified in the same way as long as we can come up with a convincing set of requirements to be implemented and verified.

Many of the adaptive and intelligent algorithms that we might consider for near-term deployment in civil aircraft are not substantively different from avionics systems. They consist of software that implements some mathematical algorithm running on standard computing hardware. For these algorithms, the main questions will be:

- Can we define a complete set of requirements that describes the function of the system and behavior that is necessary to ensure the safety of the aircraft?
- Can we accomplish a complete verification of the system, showing that in all scenarios it satisfies its requirements with no unintended behaviors?

In the following sections we will examine a number of specific algorithms with respect to these and other potential barriers to certification.

4 Adaptive Control Algorithms

4.1 Introduction

Feedback control laws are designed to ensure stability, reject disturbances, and improve specific response characteristics of a dynamic system. If the dynamics of the system are well-known and the disturbances well-understood, then a single control policy may be designed that is robust to all possible variations in the system dynamics. On the other hand, if the system carries significant uncertainty or spans a large range of dynamic behavior, then the potential variations in the model may be so large as to prevent a single robust control policy. In this case, it is necessary that the control system be adaptive to changes that emerge in the system.

A concise definition of an adaptive controller, according to Astrom and Wittenmark [5], is simply one "with adjustable parameters and some mechanism for adjusting the parameters." This includes the traditional approach of gain-scheduled control which has been used in commercial aviation for decades. With gain-scheduled control, multiple control policies are designed *a priori* at specific points within a multi-dimensional operating space. For aircraft, this space is called the flight envelope. Based on the aircraft's flight condition within this envelope, an appropriate set of control parameters (or gains) is selected from the pre-designed set of control policies.

In general, adaptive control methods can be implemented either directly or indirectly. With Indirect Adaptive Control, the plant parameters are estimated online and the controller is then synthesized from the estimated plant model. In contrast, with Direct Adaptive Control, the estimated parameters are used directly in the controller.

An overview of key adaptive control concepts and methods is given below:

Adaptation Law — All adaptive control methods require some form of online parameter estimation. This is the adaptive part of adaptive control, sometimes referred to as the adaptation law. Various methods, including some AI-based techniques, may be used to perform this step, and their performance has important implications on the stability and robustness of the closed-loop system. Several different types of adaptation laws have been developed by various researchers over the years. Several widely used adaptation laws are summarized below: [5,34]

- Gradient Based. This is also known as the MIT Rule. Here, a cost function is defined based on the error, and the adaptation law is derived by setting the time-derivative of the control parameter to be proportional to the negative gradient of the cost function. This ultimately yields an expression for the time-derivative of the control parameter that is a function of both the error signal and the sensitivity derivative. The proportional constant γ is a design parameter called the adaptive gain. The MIT rule, though widely used, does not ensure convergence and its performance is often very sensitive to the selection of γ.

- Least-Squares Based. Here, the plant is modeled as a linear combination of set functions, and the coefficients are chosen to minimize the sum of the squares of the components of the error signal. Performance is sensitive to the model structure. Using too few parameters will not provide a good fit to the data. However, using too many parameters can result in overfitting, where the model matches the measured data points very well, but is a poor model for other data sets.
- Lyapunov Based. The adaptation law here is based on a Lyapunov function of the error and state signals which is designed to give bounded asymptotic convergence of the control parameters. Lyapunov methods with projection operators have been used recently in L1 adaptive control to provide guaranteed robustness properties with fast estimation (high adaptive gain) [23, 31].

Persistent Excitation - In many parameter estimation schemes, guaranteeing convergence requires persistent excitation (PE) of the control signal. In recursive designs, the tracking error signal may reach zero *before* the learning/estimation filter is able to converge. This is not a fundamental problem, but can lead to poor transient behavior until the parameter estimation converges. Nonrecursive designs, although always convergent by definition, are still sensitive to the control signal excitation. In a weighted least squares method, for example, the estimation requires a matrix inversion where the matrix is only invertible (full-rank) if the control inputs span a set of linear independent vectors over the sampling period. In both cases, adding a PE signal to the control signal can ensure convergence, but will, in turn, lead to poor steady-state performance [35]. For civil aviation, introducing PE would significantly impact pilot handling qualities and ride comfort. For this reason, estimation methods that do not require PE are generally preferred.

Certainty Equivalence - Most adaptive control designs are based on the *certainty equivalence* principle. In general, the gains of the adaptive controller are derived from a set of uncertain parameters whose true values are not known. Using the certainty equivalence principle, the estimates of these parameters are treated as if they were the true parameters. This requires sufficiently fast parameter convergence in order for the adaptive controller to exhibit the desired performance characteristics. If certainty equivalence is not used, the design is referred to as *cautious*. In this case, the uncertainty is modeled explicitly in the parameter estimates [34, 86].

Self-Tuning Regulator (STR) - This is one of the earliest terms used for adaptive control [5]. The term “self-tuning” refers to the property that the control parameters converge to the values that would be designed explicitly if the uncertain process were known. This is now referred to as the certainty equivalence principle, discussed above. The term “regulator” refers to the control structure, which attempts to regulate the state of the plant to a set point. Such a structure does not include a reference model [5].

Model Reference Adaptive Control (MRAC) - Here, a reference model is used to define the desired response of the system given a command input. Using the error between the observed plant output and the desired response, the adaptation mechanism computes the control parameters (either directly or indirectly) to drive that error to zero.

Model Identification Adaptive Control (MIAC) - This is a form of indirect adaptive control. The parameter estimation step is used to compute uncertain parts of the plant model. The control parameters are then computed in a separate step, based on the estimated plant model [82].

Adaptive Feedforward Control / Compensation - A non-adaptive controller is augmented with an adaptive feedforward filter, modifying the control signal to achieve desired closed-loop properties. This is also known as Adaptive Augmentation. This approach is typically used to improve disturbance rejection when the nominal plant is well known, but the disturbances can be large and uncertain.

L1 Adaptive Control - Application of a relatively new L1 control design methodology enables the robustness properties of the system to be decoupled from the adaptation rate, enabling fast adaption with guaranteed robustness.

Adaptive Pole Placement Control - Here, the control parameters are computed to place the closed-loop system poles to give desired dynamics. It has been shown that MRAC is a special case of Adaptive Pole Placement Control [34]

Adaptive Augmentation - A non-adaptive baseline control law is used to provide tracking and robustness properties for the nominal system. An adaptive element is added to augment the baseline controller by estimating some measure of the plant uncertainty and driving the residual tracking error to zero. This is often implemented with an adaptive feed-forward filter to improve disturbance rejection when the nominal plant is well known, but the disturbances can be large and uncertain. Adaptive augmentation in general is a popular architectural approach that has been used by a variety of researchers on several different aircraft and missile systems [9,24,57,76,77].

Adaptive Dynamic Inversion (ADI) - Dynamic Inversion (DI) applies a fast inner loop control that inverts the dynamics, forcing the nonlinear system to exhibit a desired (linear) response to reference commands. Some form of an adaptive outer loop is added in order to compensate for the nonlinear inversion errors. [31,45,50].

Stochastic Adaptive Control - Here, the process and environment are described with a stochastic model. The control law objective is to minimize the expected value of a loss function. This is also referred to as *dual control*, as it requires a balance of minimizing the tracking error as well as estimation errors. The problem may be solved numerically using dynamic programming, but solutions do not scale well and are therefore practically limited to relatively simple models [5,86].

4.1.1 Adaptive Control Structures

In the sections that follow, we provide a brief discussion of different control system structures, and point out how various forms of adaptive control differ from one another. In the diagrams that follow, the "System" block represents the aircraft dynamic system, or plant, that is being controlled. The system *response* is fed back to the "Controller" block, which computes the *control signals* to drive the system according the *command* signal from the pilot. In each diagram, a different type of *Adaptation Method* is shown. The output of the adaptation method in each case is an updated set of control parameters that are provided to the Controller block. [1]

4.2 Gain Scheduled Control

A traditional feedback control structure with gain-scheduling is shown in Figure 4. With gain-scheduling, the adaptation method is a simple table lookup. The full flight envelope of the aircraft is typically characterized by a range of altitudes and Mach numbers, though additional parameters may also be used, such as dynamic pressure and rate of climb. In practice, this multi-dimensional space is discretized into a grid of distinct flight conditions, and a unique set of controller gains are designed for each grid point. As the aircraft state gradually moves around within the flight envelope, the pre-computed gains for the nearest set of flight conditions are used to determine the appropriate gain values at the current flight condition. This operation is typically done using a table lookup with interpolation. This methods is widely used and certified.

But it still has issues

No performance guarantees in between design points.

[1] The control parameter update is shown using standard notation in adaptive control, with a diagonal arrow crossing the Controller block.

Impractical to capture all possible model variations with a finite set of grid points.

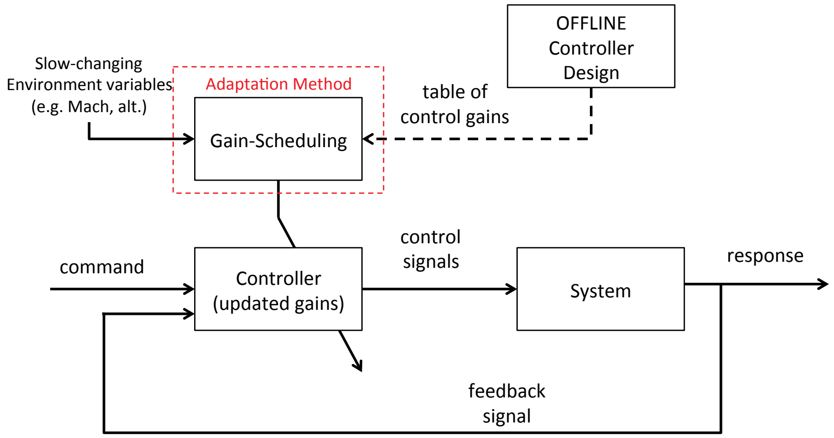

Figure 4: Feedback Control with Gain Scheduling

4.3 Indirect Adaptive Control

A block diagram for indirect adaptive control (IAC) is shown in Figure 5. This is also sometimes referred to as a model identification adaptive control (MIAC) [82]. Here, the control signals and measured response of the system are first used to perform system identification; this is where the online parameter estimation occurs. In a MIAC structure, the estimated parameters describe the system. Using the updated model of the system, a separate control design step computes the new controller gains that provide the desired performance characteristics for it.

As the diagrams in Figure 4 and Figure 5 clearly point out, the main difference between gain scheduled and adaptive control is that in the adaptive system, the controller design step is automated. An obvious advantage of the adaptive system, therefore, is that it removes the burden of having to design and store a large set of control gains. Rather than pre-computing several “point-design” controllers at discrete flight conditions, the appropriate gains are instead computed automatically for the updated system model, based on some chosen control synthesis procedure (e.g. pole placement, LQR, or $\mathcal{H}_\infty$). In this context, adaptive systems can be (and have been) adopted to facilitate offline development of control designs [82].

In order for the adaptive system to provide the desired performance characteristics, the parameter estimation step must converge sufficiently fast. The consequence of slow convergence is poor transient behavior, as the controller is essentially trying to compensate for the wrong dynamics while parameter estimates are converging to the correct values. This is a common challenge for all adaptive systems.

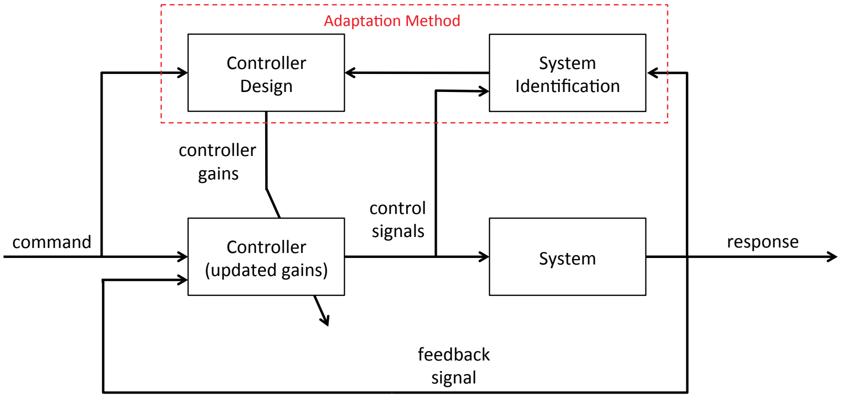

Figure 5: Indirect Adaptive Control (IAC) / Model Identification Adaptive Control (MIAC)

Key Characteristics:

Convergence rate For indirect/identification methods parameter convergence is more complicated as the correct value to which it converges to keeps changing as the system identification continues to identify the model. Thus the rate is unpredictable. Additionally, presence of noise or inadequate excitation signal may make convergence slower.

High frequency dynamics A large adaptive gain has the potential to lead to high-frequency oscillations which can excite the unmodeled dynamics that could adversely affect robustness/stability of an adaptive control law [73]

Persistent Excitation The error goes to zero but the parameters do not necessarily converge to their correct values. The input signal must have properties for parameters to converge. For convergence to occur the input signal to the process should be sufficiently exciting and the structure of the model should be compatible with the process.

Transient effects Choices of underlying design methodology that leads to oscillation or ringing should be avoided. Also, initial transients depends upon the initial values of the estimator. These characteristics vary based on the algorithm used for parameter adjustment and the amount of excitation.

4.4 Direct Model Reference Adaptive Control

Figure 6 shows a direct form of model reference adaptive control (MRAC). The command input is supplied to a reference model, which represents the desired dynamic behavior of the system. The output of this reference model is the desired response, which is compared to the actual measured response of the system to give the error. If the controller is tuned correctly, it will produce control signals that drive the error to zero. However, the presence of uncertainty in the system prevents us from knowing how to tune the controller a priori. The adaptation method for a direct MRAC is to estimate the system uncertainty parameters and use those parameters directly in the control law to cancel the error. When the estimated uncertainty parameters converge to their true values, the error is canceled completely.

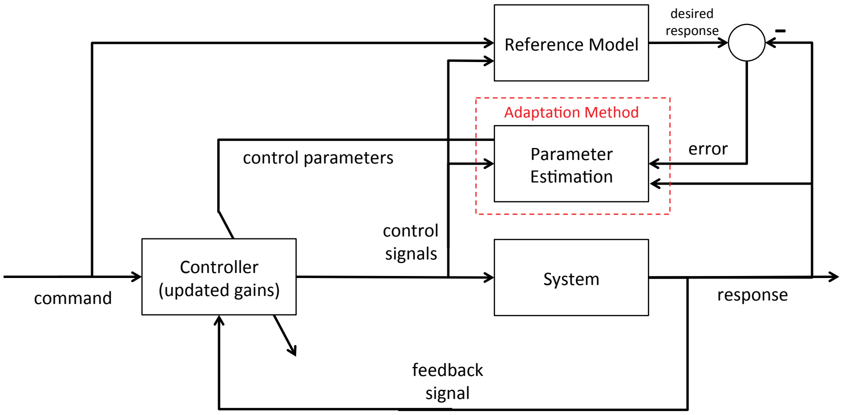

Figure 6: Direct Model Reference Adaptive Control (MRAC)

In general, the performance of an MRAC system is improved with a higher adaptive gain. If the adaptive gain is too small, the parameter estimation loop is too slow to converge, and the adaptive control is simply incapable of tracking the faster dynamics. Increasing the adaptive gain enables the parameter estimation to converge faster, reducing the tracking error and giving a better transient response. However, if the adaptation gain is too high in a traditional MRAC architecture, it leads to high-gain feedback which can cause undesirable oscillations or even instability. [84]. This tradeoff between robustness and transient performance is a fundamental design challenge with adaptive control. Rockwell Collins, Automatic Supervisory Adaptive Control (ASAC) [39] provides an example of adaptive algorithm, which gets enabled only in off-nominal conditions. In the specific case of adapting to a wing loss, ASAC tracks aileron trim offset, which normally should be very close to neutral (zero). When trim exceeds a certain threshold, then ASAC employs a PI control law, which commands a sideslip angle to offset the aileron trim. When aircraft is not damaged, ASAC output is zero. When damage occurs, it does the right thing by design, without caring what kind of damage occurred.

Key Characteristics:

Convergence rate Parameters converge to the correct value based on the reference model either gradually or rapidly. This rate of convergence depends upon the value of gain. For low values of gain the rate increases with gain but for higher values of gain the behavior is unpredictable.

High frequency dynamics Faster convergence requires a large adaptive gain which has the potential to lead to high-frequency oscillations which can excite the unmodeled dynamics that could adversely affect robustness/stability of an adaptive control law [73]

Lack of parameter convergence The error goes to zero but the parameters do not necessarily converge to their correct values. The input signal must have properties for parameters to converge.

4.5 L1 Adaptive Control

A relatively new methodology called "L1 adaptive control" (or L1AC) was introduced in 2006 to address the challenge of simultaneously achieving both desired transient performance and guaranteed robustness properties [11, 12]. The L1AC approach was initially developed from a conventional MRAC architecture. One representation (though not representative of every possible architecture) of L1AC is shown in Figure 7. Comparing this to the MRAC diagram, the two main architectural differences with L1AC can be seen clearly. Namely, a low-pass filter is applied to the control output, and the estimated control parameters are fed to the reference model, making it a state predictor.

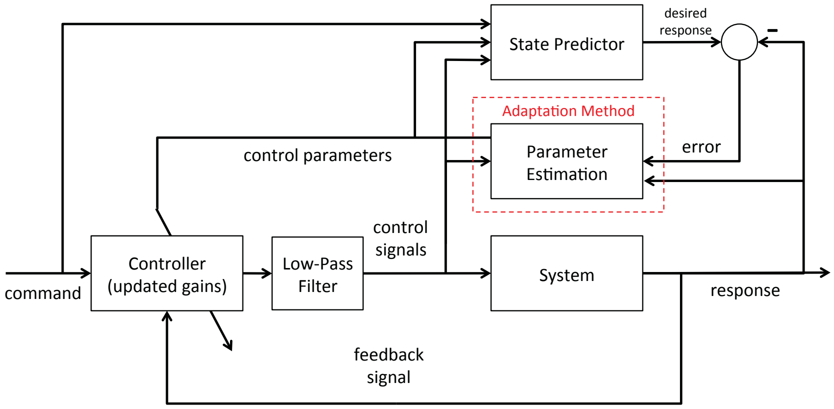

Figure 7: L1AC - L1 Adaptive Control Structure

The authors of L1AC point out that the main distinction from MRAC is the problem formulation, which seeks to determine a control law that ensures: 1) predictable transient performance for all $t \geq 0$, and 2) a bounded control signal that does not exceed the control system bandwidth. The key to achieving these objectives is to synthesize an appropriate control architecture that decouples the estimation loop from the control loop. In some cases, this can be achieved with minimal deviation from the conventional MRAC architecture, for example, by just inserting a low-pass filter on the controller output. In particular, this approach is shown to be satisfactory for systems that have *known* input gain [32]. For systems with *unknown* input gain, the control law must take on a slightly different form to provide the necessary decoupling between the estimation and control loops.

With the decoupling of estimation and control provided by the L1AC scheme, the adaptive gain can be increased to improve transient performance without introducing high-gain feedback in the control channel. In this sense, the adaptive gain may be increased arbitrarily; theoretically, it is only limited by the processing capability of the computer hardware implementing the filter. For a conventional MRAC, the time-delay margin of the closed-loop system vanishes to zero as the adaptive gain is increased. In contrast, the time-delay margin for L1AC remains bounded away from zero, guaranteeing a certain level of robust stability. The usual design tradeoff between performance and robustness is still present in L1AC, but it is no longer sensitive to the rate of adaptation. Instead, the performance / robustness tradeoff is isolated in the selection of the control filtering structure, which may be designed using traditional control design methodologies.

Notably, L1AC methods have been used successfully on several flight tests of the NASA subscale Generic Transport Model (GTM) [9, 23, 24]. These flights were conducted to support test and evaluation of control laws beyond the edge of the normal flight envelope, such as post-stall, high-alpha conditions, where there is significant nonlinearity and uncertainty in the aerodynamic model. L1 adaptive can be utilized for online tuning of gain schedules. Imagine an experimental software load, which runs a fixed control law with configurable parameters, where parameters are changed in real-time by L1 adaptive algorithm, running on another processor. The processor running the control law can impose restrictions on parameter changes, rate of change of parameters, etc, to make it safe. Once gain tuning has been accomplished throughout the flight envelope and a set of aircraft configurations covered by intended type certification, the schedules are frozen, and loaded into the autopilot computer memory. Again, this does not require any changes to current cert process. This method can be used when plant models are either unavailable, or so poor that the offline optimization is not tenable.

Key Characteristics:

Computational demands The L1-adaptive control method seeks to decouple the estimation loop from the control loop, enabling the adaptive gain to be arbitrarily high. However, the large adaptive gain leads to high bandwidth channels in the estimation loop which requires fast processing. This can potentially place unrealistically high demands on the computer hardware.

4.6 Adaptive control with Neural Networks

The adaptation method for indirect adaptive control architectures can be executed with a neural network [49]. Neural networks (NN), which are discussed in Section 5.3, are effective at approximating continuous nonlinear functions, which make them well-suited for system identification. This represents the learning component of indirect adaptive control.

When implementing a NN for adaptive control, the network weights may be trained offline, or they may be dynamically updated online. In the case of offline training, the network weights are computed prior to flight based on an extensive set of simulation data. When the NN is later implemented online, the weights remain fixed. In this case, the NN simply provides another functional way of performing nonlinear regression to estimate parameters. It is an alternative to a recursive filter.

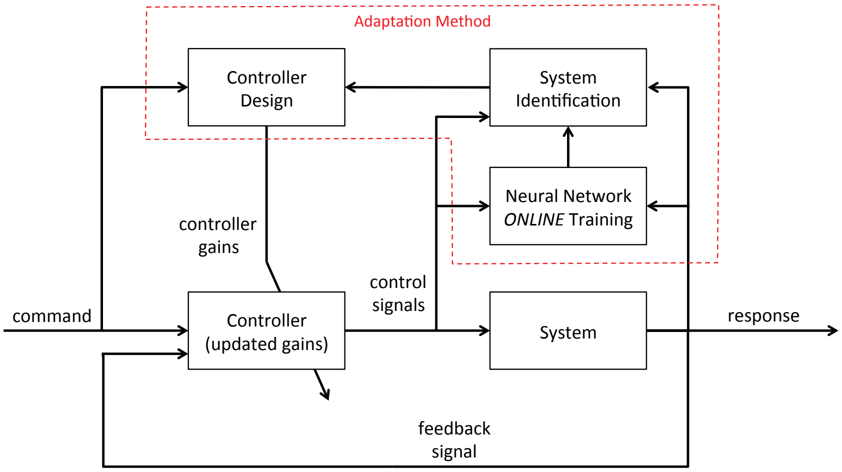

Figure 8: Indirect Adaptive Control using a Neural Network with Online Training.

Figure 8 illustrates an indirect adaptive control structure in which the NN is dynamically updated *online*. In this case, the NN weights are actively trained during flight based on the reference commands, control signals, and system response that are provided as inputs. The training of the NN becomes an integral part of the adaptation method.

Key Characteristics:

Convergence For online neural networks, the convergence rate for training has to be sufficiently fast in order to make performance guarantees. Different types of optimization algorithms may be used to train neural networks, depending on the size and structure of the problem, such as gradient descent, evolutionary algorithms, and simulated annealing. While these methods have different convergence properties, neural network training typically suffers from slow convergence due primarily to the prevalence of flat regions of the objective function and the large multi-dimensional space over which the search must take place.

Transient Effects Using a neural network that has not yet converged as a means of performing system identification can introduce undesired transient effects.

Overfitting Poor generalization to new inputs, a common issue with neural networks. Can be overcome by training with diverse data sets that capture all dynamic conditions.

Adequacy of Design Existence of solution is not guaranteed

4.7 Summary of Key Characteristics for Adaptive Control

Although the field of adaptive control has matured significantly, there are still a number of issues that require further research. In 2006, Wise et. al. identified the following open problems [84]:

Reference Model Design The appropriate selection of the model for the aircraft is critical to respond to the situation. This is because the error which is the difference between the actual model and the reference model is driven to zero.

Parameter Tuning Guidelines In the design process several parameters and matrices are used that need tuning. Tuning of these could lead to transients which needs to be addressed.

Adaptive Dead zone and Learning Rates The sizing of the dead zones and the learning rate is critical. This is because fast learning rate can prevent departures due to gust rejection. High learning rates introduce unwanted high frequency oscillations and transients.

Adaptive Structural Mode Suppression compensate for structural mode interaction with the control system. Presently the filters are often conservative and reduce stability margins more than necessary to account for change in mass properties and flight envelope.

Gain and Phase Margins for Adaptive Systems Simulation analyses have shown that adaptive system with high learning rates are not robust to time delays in the feedbacks. This begs the question as to how much margin is present in the design, and how best to analyze such a nonlinear problem.

The approaches to resolve these issues were developed by modifying the architecture of some of the existing adaptive control technologies. These approaches are L1-adaptive control, combination of indirect-direct approaches.

In 2008, Jacklin et. al. identified five major gap areas [35]

- Defining adaptive controller requirements
- Simulation models for adaptive control
- Proving learning stability and convergence
- On-line monitoring tools
- Adaptive controller certification plans

Presently, requirements for certified controllers are well defined. A similar well defined approach to define requirements for designing adaptive controllers needs to be established. Along with defining the requirements it is key to understand the characteristics of adaptive control technologies that pose certification challenges. These key characteristics can be attributed to the execution of methods in real time that deal with the estimation and update of values, and they can be grouped into two main categories: **convergence**, and **values of adaptive gain**. Convergence has several issues associated with the uncertainty to converge, and delay in converging. This leads to an unpredictable behavior that needs to be well understood and bounded. During the process of convergence, the value of adaptive gains dynamically change. Larger values of adaptive gain leads to unwanted behavior by most of the adaptive algorithms. This means that online monitoring tools need to be deployed to check the safety bounds and assure stable transient response. Furthermore, the stability criteria for traditional control does not apply to adaptive control due to the nonlinear characteristics of adaptive control. As such, it should be reasonable to replace the phase margin with a time delay margin as one measure of stability for adaptive control.

Finally, it is important to consider the issue of non-determinism. Because the adaptive gains are computed as a result of a recursive feedback process with stochastic inputs, their values are inherently non-deterministic. For many, this is immediately perceived as a negative characteristic of the algorithm, and a stumbling block for safety assurance. Is this necessarily true, or is it a misperception? At a recent autonomy workshop hosted by the AIAA Intelligent Systems Technical Committee, Dr. Kevin Wise of Boeing addressed this question with a useful analogy [83]. In a traditional PID controller, the integrator state reaches a steady value at a trim flight condition. Due to small variabilities between aircraft and the stochastic nature of the input signals provided to the controller, the internal value of this integrator state is also non-deterministic. The reason this is not perceived as a problem is because the internal state is computed as a result of a stabilizing feedback law. Arguably, the same claim can be made for the gains in an adaptive controller.

5 Artificial Intelligence Algorithms

5.1 Introduction

The field of artificial intelligence consists of a diverse set of computational methods whose common trait is that they are either grounded in or inspired by human intelligence. The advent of modern AI occurred in the 1950's shortly after the arrival of the digital computer, as several researchers set out to build a machine as intelligent as a human being. Since then, the field has grown in multiple directions and has sparked the development of several new approaches that transcend AI and have become disciplines in their own regard, such as statistical inference and data mining.

In this chapter, we provide an overview of several different AI methods and algorithms. Its important to note that some of the methods have significant overlap with others, such as qualitative physics and fuzzy logic. In addition, many of the methods provide complementary functions, leading to hybrid implementations that incorporate multiple technologies. For example, a fuzzy controller may use a neural network to implement its membership functions, and a neural network may be trained using reinforcement learning algorithms.

As will be seen, AI methods pose unique capabilities that can provide important benefits to civil aviation. However, all civil aviation applications occur in an operational setting with requirements for safety assurance, resource constraints, and time-critical performance. It is therefore necessary to consider the key characteristics of each AI method that present challenges to meeting these requirements.

In each of the sections that follow, we consider a different category of AI methods. We first provide a general description of the overall method, and then discuss key characteristics.

5.2 Machine Learning

Machine learning deals with the design and study of systems that learn from data. The process by which learning is achieved can take on many different forms. In general, though, the purpose of learning is to improve the performance of a task based on experience. In this sense, the claim that learning has occurred is evidenced by the fact that performance has improved after the gathering of experience.

All learning systems must be able to 1) represent data, and 2) generalize from data. A training data set may first be provided when the learning system is constructed. The learning system is designed to adapt so that it performs well (provide some desired response) when given the training data set as an input. The learning system is also intended to *generalize* from the training data, so that it will continue to perform well when given new, previously-unseen data. Some learning systems continue to adapt while deployed.

There are three main sub-divisions of machine learning:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

In supervised learning, algorithms are trained on specific input data sets that have a corresponding desired output. The objective in this training phase is to generalize a function or mapping from inputs to outputs which can then be used to predict outputs for new, previously unseen inputs. When operating online, the trained system essentially applies what it has already "learned" (e.g. the mapping function) so that it can draw inferences from new data.

With unsupervised learning, the input data sets do not have a corresponding desired output. Instead, the learning objective is to discover structure in the data. In other words, the goal is not to return the "right" answer for given data, but rather to identify certain features or statistical properties that it exhibits. One widely-used form of unsupervised learning is called clustering, where multiple sets of input data are organized according to common characteristics into distinct clusters.

Reinforcement learning (RL) is focused on the use of intelligent agents to accumulate knowledge and develop policies. As the agent takes action it changes its state within the environment, producing an outcome and an associated reward. Through a discovery process, the agent can learn the approximate the value of taking an action in a given state. As more information is gained, the accumulating set of action / result observations are used to learn an "action-value" function. With finite states and actions, this can be done in a discrete table-lookup fashion using algorithms such as active dynamic programming (ADP), temporal-difference (TD) learning, and Q-learning.

Further generalization can be achieved by using some form of functional approximation for the tabulated data, such as a neural network. In fact, neural networks are often used to implement all three of the learning paradigms discussed above. More details about neural networks are provided in Section 5.3.

Different forms of reinforcement learning have been used to carry out some extremely challenging problems in aerospace vehicle control. In particular, RL techniques have been applied to different types of helicopter control problems. In [46], RL is used to execute autorotation maneuvers in response to an engine failure; in [2], a controller is generated from RL using a reward function that was derived from piloted flights, and is then used to autonomously perform aerobatic maneuvers; and in [52], a RL algorithm was used to learn a controller for autonomous inverted hovering. Others have applied Q-learning to the problem of morphing control, in which aerodynamic control is achieved by actively deforming the shape of the vehicle [43, 75].

Key Characteristics:

Stochastic Process — The algorithms that implement the discovery process can be stochastic in nature, explicitly assigning random values to algorithm parameters. The purpose of injecting random numbers is to promote exploration of new parts of the state space, and thereby improve learning. The consequence, however, is that it prevents the algorithm from being perfectly repeatable for the same set of input data.

Convergence — The convergence rate in machine learning depends strongly on the stochastic properties of the input sources, as well as the learning algorithms themselves. Some algorithms that are used in reinforcement learning can exhibit poor convergence properties, especially if the input signals are noisy. Others, such as Q-learning and value-iteration, have stronger convergence properties.

5.3 Neural Networks

Neural networks are computational models that are inspired by the operation of the nervous system. Some of the earliest work in artificial intelligence was aimed at developing artificial neural networks in order to emulate the behavior of the brain - specifically, the stimulation of interconnected neurons. The resulting structure is a versatile mathematical tool that has proven to be extremely effective for a wide range of statistical analysis applications.

The network structure is composed of nodes (neurons) connected by links. Each node has multiple inputs and a single "activation" output. A representative mathematical model for node j is given as:

$$a_j = g\left(\sum_{i=0}^{n} w_{i,j} a_i\right)$$

Here, a_i is the activation value from a prior node, and a_j is the activation value of the current node j. The function g is called the activation function, which is typically a hard threshold or a logistic function. The input to the activation function is a weighted sum of all of n inputs to the neuron, where $w_{i,j}$ is the weight for using input i at node j.

Two important distinctions in network structure are that of feed-forward versus recurrent, and single-layer vs multi-layer. In a feed-forward network, the connections between neurons always go in the same direction, from the input side of the network to the output side - there are no loops. As such, the structure of a feed-forward network forms a directed acyclic graph, and it is arranged in one or more layers. Multi-layer networks have at least one layer that is internal to (or hidden within) the network. A recurrent network uses feedback loops explicitly, returning node outputs back as inputs to nodes in previous layers.

Neural networks are known to be good at pattern recognition, classification, clustering, and data fitting. Essentially, the network can be designed by adding layers and defining the weight values in order to approximate an arbitrary nonlinear function. In a pattern recognition problem, for example, the task would be to recognize known patterns (for example, printed letters) given a set of pixel intensity values from a camera. The number of inputs would be equal to the number of pixels in the image, and there would be 52 output neurons, representing all possible upper and lower case letters in the alphabet. The neural network would be trained on a set of example data, and then (if all goes well) when an image of the letter "g" is supplied, for example, only the output neuron associated with that letter would be activated.

The letter recognition example described above is a case where a supervised learning process would be used to train the network. Supervised learning is appropriate when there exists a desired mapping of inputs to outputs, and we want the network to emulate this mapping. Neural networks can also be developed with an unsupervised learning process. In this case, the network is given input data sets but with no accompanying outputs. Rather than learning a function that approximates an input/output map, as was the case with supervised learning, the goal here is to detect patterns, or statistical relationships within the data. Unsupervised-learning neural networks are often used for clustering tasks, where the input sets are grouped into clusters based on different measures of how similar they are.

Within the framework of reinforcement learning, neural networks have been used as function approximators, generalizing the utility function to go beyond the finite limits of a discrete table-lookup. The use of neural networks in reinforcement learning has proven particularly useful for complex, high-dimensional environments, including systems with nonlinear continuous dynamics.

If a NN's weights are fixed once trained, the NN becomes highly deterministic system that can probably be certified using existing methods, analogous to a gain-scheduling table. However, the more likely use of NNs would be in dynamic applications where they would be used to continuously learn/update their model of some system behavior (e.g., aircraft dynamics), and thus they face the challenges of all online learning methods.

Key Characteristics:

Overfitting — A well-known drawback of neural networks is the possibility of overfitting. This is a general phenomenon that can impact any type of function approximator. A useful analogy is using a higher order polynomial to fit data when the trend is adequately captured by the lower order model. The higher order polynomial can reduce the error on the data sets that are given, but it may lead to larger errors on points outside of these data sets. The same holds true for a neural network, where more nodes and layers can provide a better fit for the training data sets, but may result in poor generalization on new input data.

Convergence — With most NN structures, there is no time-bounded guarantee that the training process will converge to an effective classifier or other NN system; designing NNs remains essentially an art.

5.4 Expert Systems

Expert systems (ES) attempt to capture the knowledge and experience of human experts and then use this information to form rules that suggest the best response for a given state of the world. Historically, expert systems were one of the first successful methods of applied artificial intelligence.

Structurally, an ES is composed of two main parts: a knowledge base and an inference engine. The knowledge base stores facts and rules about the environment that the system is expected to operate within. The inference engine uses the knowledge base to find a solution (or answer) for the given problem (or question). Different types of inference engines may be used, including rule-based and case-based methods.

A rule-based system applies a set if-then rules of the form: if A and B and C then X and Y, where A, B, C are the conditions and X, Y are the consequents. If all of the conditions are true, then the rule is *triggered* and the consequents are *fired*. The consequents may be new facts that are stored to the working memory inside the system, or actions of some form that are sent outside of the system.

The reuse of a generic ES shell enables domain experts to directly author and revise rules, an advantage that enables rapid prototyping of designs and makes the code base easier to maintain. However, knowledge acquisition remains the most intensive and time consuming aspect of an ES. As a result, ES designs tend to be limited to domain-specific applications with a narrow scope, in order to reduce the amount of knowledge acquisition that is required.

Expert systems can take on many different forms, and may be integrated with other AI methods. For example, the knowledge base may be represented as a

Bayesian Network (BN) in order to support probabilistic reasoning, and fuzzy logic may be used to implement the rules in the inference engine. The system could also be integrated with a learning component, where the knowledge base is updated automatically through a separate learning process. BN is a multi-use tool for reasoning with uncertainty. It provides a systematic way to incorporate probabilistic dependencies for events of interest or provides a formalism for reasoning with uncertainty when partial beliefs are available. It is a tree structured network. A BN can be used to learn causal relationships, and hence can be used to gain understanding about a problem domain and to make predictions.

The inference engine applies rules from propositional logic and uses search methods, such as forward chaining or backward chaining, to compute solutions. With forward chaining, the algorithm starts with available input data, applies it to the rules in the knowledge base, then eventually concludes with one or more resulting statements about the data set. These results may then be added to the knowledge base. In contrast, backward chaining is a goal-directed method. It begins with a goal, and applies logical inference to the knowledge base in order to determine whether there exists any data to support goal.

The general problem of applying propositional logic to determine whether a goal is met is commonly referred to as a *satisfiability* problem - or SAT problem. The SAT problem is a fundamental problem in AI and computer science in general, as many different types of combinatorial problems, including planning and model checking, can be reduced to checking SAT for a propositional logic sentence. All SAT problems are in the computational complexity class *NP-Complete*, and in the worst case require a combinatorial run time. For example, the worst case for a problem with n binary variables requires full enumeration, which has 2^n combinations.

A substantial amount of research has been dedicated to developing more efficient SAT solvers. One of the most widely used algorithms is the Davis-Putnam-Logemann-Loveland (DPLL) algorithm, which applies a form of complete backtracking [65]. DPLL uses a recursive, depth-first enumeration of all possible models. Several variations have been applied to DPLL in order to enhance its ability to scale up to extremely large problems. Some techniques make explicit use of randomness. For example, a *random restart* can be done if the current run is not making progress, randomly generating a new candidate solution, while avoiding conflict clauses that have already been learned. Another popular method is called *WalkSAT*. At each iteration, the algorithm chooses an unsatisfied clause and randomly picks one variable in that clause to flip. WalkSAT is best applied to problems where a solution is expected to exist, as it is not capable or determining that the problem is unsatisfiable.

Key Characteristics:

Completeness of Knowledge Base — In order for the ES to be effective, the knowledge base must be provided a sufficient amount of information, in the form of primitive rules, in order for the inference engine to

Subjectivity of Knowledge Base — Parts of the knowledge base may be subjective. Multiple experts may in fact give contradictory inputs to form the knowledge base. In such cases, it is an open question of how to properly identify which set of rules is correct, because the notion of correctness is not clearly defined.

Long Run Times with Large Knowledge Base— Inference engines effectively apply some type of SAT solver to check whether the input data is supported by the rule base. For large knowledge bases, the solution could take extremely long run times. This could render the ES ineffective for operational settings in which a timely result is needed.

Randomness in the Inference Engine – Some algorithms used to implement the inference engine explicitly use random numbers to search for satisfying solutions.

5.5 Fuzzy Logic

The concept of "fuzzy logic" was born in the 1960's along with the introduction of fuzzy set theory. In classical set theory, the membership status of an element is binary – it is either a member of the set, or it is not. With a fuzzy set, however, elements can have a partial "degree of membership" within the set. The extent to which an element is in the set is defined by a membership function, which is valued in the interval [0,1].

For example, the element *Honda Accord* firmly belongs in the set *Car*, and so classical set theory is sufficient. However, is *Honda Accord* a member of the set *Foreign* or *Domestic*? The answer is unclear. Honda is a foreign-owned company, but it manufactures parts and assembles vehicles in the United States. Fuzzy set theory is appropriate here, because it can have partial membership in both sets. We could reasonably assign it a membership value of 0.6 for *Foreign* and 0.4 for *Domestic*. It is important to point out that the choice of these numeric assignments is arbitrary.

The notion of "fuzzy" does not represent probability or uncertainty in data. Rather, it models the ambiguity that exists in the meaning or interpretation of the data or event. As such, fuzzy sets prove useful when the information describing a model is inherently imprecise.

Fuzzy logic is an extension of classical boolean logic that uses fuzzy set variables to express truth values. The application of fuzzy logic grew in popularity through the 1980's and 1990's, primarily as a means for controlling complex systems that were difficult to model, and that had limited sensing or actuation. The application of fuzzy logic in control systems has been proposed for a variety of applications, in aviation [27]. A fuzzy control system implements three main steps: fuzzification, inference, and defuzzification.

Fuzzification The numeric values associated with sensory inputs are converted to fuzzy values. For example, a velocity of 50 miles per hour could be converted to fuzzy value of "fast", whereas 80 mph would be "very fast".

Inference Fuzzy inputs are applied to a rule base to determine the appropriate fuzzy actions. For example, the fuzzy inputs of "fast" and "neighborhood road" might have an associated rule that leads to the action "slow down". The combination of "very fast" and "neighborhood road" would infer a stronger "slow down immediately".

Defuzzification Fuzzy actions are converted into crisp, well-defined output values. For example, a fuzzy action of "slow down" could correspond to a numeric deceleration command of 1 mph per second, but "slow down immediately" might produce a 5 mph per second deceleration.

As the above examples indicate, classifying data is an important aspect of any fuzzy logic controller. In fact, neural networks can be used in an unsupervised learning mode to perform data clustering for a fuzzy system. The grouping of related data sets into clusters would provide the basis for defining membership functions.

Key Characteristics: Some implementations of fuzzy logic use machine learning to learn the membership functions. These systems inherit the same key characteristics of those learning algorithms. Another characteristic unique to fuzzy logic is its ambiguous definition of degrees of membership.

Ambiguity — The whole point of fuzzy logic is to provide an effective way to deal with imprecision. Consequently, fuzzy logic algorithms themselves are also grounded in ambiguity. The fuzzification and defuzzification steps involve an arbitrary mapping between precise numeric values and imprecise conceptual states. These steps rely on the use of membership functions, which could be fully defined a priori, or learned during execution via machine learning. In either case, there is no universal way of assessing the merit and quality of the membership functions. In other words, we cannot produce a measure of how good the membership function is, because there is no "ground truth" to which we can compare.

5.6 Cognitive Architectures

A cognitive architecture [56] specifies the underlying infrastructure for an intelligent system. The architecture supports the creation and understanding of synthetic agents that emulate the same capabilities as humans. A cognitive architecture includes aspects of cognitive agents which are: memory of an agent that stores the goals, beliefs and knowledge, the representation of the elements, their organization of the mental structure and the functional processes that operate on these structures. The functional processes can be either utilizing the information or updating it based whatever was learned.

With the increase in demand for enhancing autonomy for autonomous applications there is the need to integrate systems that exhibit intelligent behavior not just improve components within the system. Thus there is a need for increased focus on system level architectures that support complex cognitive behavior across a broad range of relevant tasks.

These cognitive architectures have differences in the way they implement capabilities or features. In some of the research efforts the architecture is static and in others the architecture is allowed to grow. Learning is not part of all cognitive architectures. Also some approaches allow several components to execute concurrently or one at a time. These are different ways cognitive architecture is implemented based on understanding of certain aspects of human capability. Some of the cognitive architectures that have been applied more frequently in different applications are: ACT-R, Soar, ICARUS, PRODIGY and MIDAS.

Key Characteristics:

Convergence— Depending upon the learning mechanism implemented the algorithms might not converge to a solution.

Uncertain behavior— Based on the learning mechanism used and the source data used the response behavior will vary.

5.7 Planning and Scheduling

Automated planning and scheduling systems are designed to find strategies or action sequences that achieve some goals. Given a description of the possible initial states of the world, a description of the desired goals, and a description of a set of possible actions, the planning problem is to find a plan that is guaranteed (from any of the initial states) to generate a sequence of actions that leads to one of the goal states. If the duration of actions or other temporal aspects are important, the problem may include elements of scheduling (i.e., determining not just what actions should be taken in what order, but when).

There are many classes of planning problems, depending on whether characteristics such as observability, controllability, nondeterminism or probabilistic behaviors, durative actions, etc. are included. Numerous planning algorithms have been developed, most relying on some form of search through the space of possible future states of the modeled system. Planning problems can often also be reduced to equivalent satisfiability and model checking problems, so solvers from those research areas can also be used.

Key Characteristics: Planning and scheduling systems have been successfully deployed in numerous real-world applications, including control of satellites ([13]) and space telescopes ([38]). However, they generally include several critical aspects that would make them challenging to certify:

Uncertain execution time — Most planners use deterministic algorithms but solve problems that are at least NP-complete, which means that their execution time on new problems is not predictable. Unless all of the problems that a particular system would ever face can be tested in advance, traditional certification methods would not apply.

No guaranteed solution — Many planning problems have no solution, and because they are NP-complete or worse, it is not possible to know whether a solution exists without actually solving the problem.

5.8 Computer Vision / Machine Vision

The field of computer vision (or machine vision) deals with the automated processing and analysis of electro-optical sensory data. In a general sense, computer vision may be viewed as a special case of data mining. Whether the specific task is edge detection, object recognition, feature extraction, or centroiding, the common objective is to extract useful information out of the raw data. Much of the development in this field has been guided by an attempt to mimic human vision, which is extraordinarily good at quickly processing and understanding visual scenes.

With the introduction of digital cameras, images have become a readily available sensor input for many systems. In manufacturing, for example, visual inspection is routinely performed at points along the production line to spot defects. In spacecraft operations, automated rendezvous and docking uses images as a primary source of navigation data.

There are number of specialized sub-fields within computer vision:

Feature Extraction The data is searched to find specific features such as lines, edges, ridges, and ovals.

Detection / Segmentation Specific points or regions in the image are selected for further processing. This selection is often based on the outcome of a lower level feature extraction step.

High-Level Processing The focus shifts to pre-selected regions of the image along with identified features in order to make a higher-level determination about the image. Examples are:

- Comparison to a model database for anomaly detection (e.g. for quality assurance in manufacturing)
- Estimation of image feature characteristics, such as incidence angle or size (e.g. for navigation)
- Classification of the image into one or more categories (e.g. for surveillance)

An assortment of different algorithms are used to implement the various aspects of computer vision listed above. The brunt of the workload tends to occur in the feature extraction step. Finding geometric features within a large space of pixel data, and then repeating that process to perform more complex pattern recognition, has proven to be a stubbornly difficult task for computers. As a result, a great deal of research has been devoted to finding efficient algorithms for this class of problems.

There are two many different image processing techniques to give an exhaustive list. Instead, we will briefly discuss a few different approaches that are used for different aspects of image processing.

Canny edge detection is a widely used method for identifying edges in images. It first applies a Gaussian filter to smooth the image so that spikes from noisy pixel regions do not trigger false edges. It then applies a gradient operator to compute the intensity and direction of changes in pixel strength. A non-maximum suppression technique to determine whether pixels are better candidates to form an edge than their neighbors. This amounts to a prioritized search over a subset of the image. Finally, edges are traced through a hysteresis thresholding procedure. A similar technique is Canny-Deriche edge detection, which uses a different smoothing filter that can be more easily tuned.

The Hough transform is a different technique of feature extraction, used primarily to find circles or ellipses. Given a target shape to find, candidate objects in the image are obtained from applying the Hough transform, and those candidates are then voted on to produce a final solution. The transformation takes (x, y) pixel coordinates and puts them into the equation $r = x \cos\theta + y \sin\theta$. Circles are found, for example, when edges are identified along (x, y) coordinates for a roughly constant r value and $\theta \in [0, 2\pi)$.

The above two methods are popular techniques for low-level image processing: detecting basic geometric shapes. Going a step further and identifying more complex shapes, such as animals, vehicles, faces, and their orientation, is a much more challenging problem in machine vision.

Recently, a research team from the University of Toronto developed an algorithm called *SuperVision*, which decidedly won the ImageNet Large-Scale Visual Recognition Challenge in 2012 with a recognition error rate of 16% - about twice as good as the next best entry [64]. The challenge includes both image-level annotation, determining whether objects of a certain class are in the image or not, and object-level annotation, finding known objects in the image and drawing a box around them. To give an idea of the size and scope of this work, SuperVision used a deep convolutional neural network, consisting of about 650,000 neurons in five convolutional layers, with 60 million tunable weights. The network was trained on 1.2 million images, which took 6 days to complete running in parallel on two NVIDIA GPUs [41]. Since that competition, several other researchers have used the deep neural network design and improved upon it. The recent winner of the 2014 contest is the Google-led project called GoogLeNet, which now achieves an error rate of just 6.7%.

Key Characteristics:

Resources – The recent trend in machine vision is to use deep neural networks, as they have demonstrated superior performance and scalability in the most challenging vision problems. However, the remarkable improvement in accuracy with these methods comes at the cost of some hefty resource requirements. Network sizes tend to be extremely large, with millions of tunable weights and several hundred thousand neurons. This places stringent requirements on computational resources, often resulting in parallel execution of multiple processors in order to tackle the problem.

Training — Long training times and large data sets are required for deep neural nets. This effectively prevents online training from being a viable option in most operational settings.

Overfitting – Due to the extremely large size of the deep neural nets, overfitting is a common problem.

5.9 Qualitative Physics

Qualitative physics is a form of knowledge representation designed to reason about physical systems using only qualitative abstractions. Essentially, the model of the world is expressed at an abstract level using qualitative attributes of behavior, rather than numerical simulations.

The field of qualitative physics was originally motivated in the 1970's with Fahlman's BUILD planner, which found ways to assemble complex towers out of blocks. The planner spent most of its time modeling the physics of various block configurations. However, the full numerical model used a level of precision that was deemed unnecessary for the purposes of the planner.

One advantage of qualitative physics is that its models contain explicit explanations of the macro-level behavior. Contrast this to a numerical simulation, where any macro-level explanations must be separately deduced from the bottom-up, a non-trivial task requiring significant computational effort. The human brain is very good at dealing with and properly characterizing imprecise quantities, especially when it comes to pattern recognition and example-based learning. Computers, on the other hand, are not well-suited for these tasks. The computer chips we have built are much better (and significantly more capable than the human brain) at storing large amounts of data in terms of precise numeric values, and quickly performing successive operations on those numbers.

Many of the mathematical models we use to describe physical systems involve a great deal of approximation and simplification. Even when a complete, high-fidelity model is available, the simulated behavior is often not informative and requires further interpretation, which in many cases is represented qualitatively. From this perspective, the precision used in the numerical simulation can turn out to be completely unnecessary.

The application of qualitative physics invokes three fundamental pieces: 1) qualitative variables, 2) behavior rules, and 3) an inference mechanism. Qualitative variables take on a finite set of values, usually in the ordered set of symbols: $+, 0, -$. The concepts of addition, subtraction, multiplication and division are explicitly defined for all combinations of these values. The description of the system being modeled is expressed in terms of behavior rules that involve one or more qualitative variables. These rules represent all of the knowledge about the system. The inference mechanism applies the behavior rules to the current set of qualitative variables to predict the future state of one or more variables in the system. When the basic set of qualitative variables is insufficient to capture the full behavior, additional "landmark" variables are introduced. For example, a system may introduce new landmark variables at each time point in order to capture the time-varying nature of a system, as opposed to a system that is time-invariant.

Key Characteristics: Qualitative physics is similar to fuzzy logic in many ways. Both methods are used to efficiently dealing with imprecise or abstract quantities, and both methods replace traditional high-precision numeric variables with a less descriptive counterpart. At the same time, qualitative physics also resembles some aspects of an expert system, with a knowledge base of rules and an inference engine to operate over those rules in order to reason about (draw a conclusion or predict an outcome for) the future behavior. As such, this class of methods inherits the same key characteristics that are found in fuzzy logic and expert systems. In addition, qualitative physics has some other characteristics unique unto itself:

Large Search Space — Because finite values (e.g. $+, 0, -$) are assigned to the qualitative variables, problems involving search over the space of these variables suffer from a combinatorial explosion when the number of variables is large.

Completeness of Rules — The behavior of the system modeled in qualitative physics is limited to the set of behavior rules that are given to it. As these rules are defined in a qualitative way, it is difficult to determine whether they adequately capture the full range of behavior for the system being modeled.

Ambiguity — The attribution of values to the qualitative variables is arbitrary. Because the behavior models and variable representations are abstract rather than numeric, there is no "correct" or "incorrect" way of assigning these values. The possibility of multiple different acceptable values for a variable leads to ambiguity, and creates a challenge for testing and algorithm validation.

5.10 Evolutionary algorithms

Evolutionary algorithms (EA) are a class of heuristic methods that have proven very effective at solving global optimization problems. The general process of solving an optimization problem involves finding a candidate solution that satisfies constraints, evaluating the utility of that solution, and then progressing to a better solution. In numerical optimization, gradient based methods choose the next candidate solution by moving along the direction of the cost function gradient. However, these techniques are limited in that they can only be applied to continuous differentiable functions, and they are not globally optimal – they only converge to a local minimum of the function, which may not be the global minimum.

The term "evolutionary" refers to the manner in which EA methods compute new candidate solutions. Just as neural networks are inspired by our understanding of how the nervous system works, evolutionary algorithms are inspired by the process of biological evolution. In the context of an EA, candidate solutions represent individuals in a population, and the quality of each solution is measured by a fitness function. The process of the EA begins by generating an initial population (a set of candidate solutions), which may be done randomly or through some deterministic method – this represents the first generation. The fitness of each individual in this generation is then evaluated, and the best-fit individuals are selected for "reproduction". In the reproduction process, new individuals are created by applying crossover and mutation operations to the fit parents. Finally, the next generation is created by replacing the least-fit individuals with the newly created offspring, and the process repeats.

Applications within civil aviation that could utilize evolutionary algorithms include the types of problems that have a large number of variables, complex constraints and multiple objectives, especially those with non-smooth functions. Some examples are trajectory optimization, departure and arrival scheduling in air traffic control, and large scale coordinated route planning by airlines. Evolutionary algorithms, coupled with distributed (e.g. cloud) computing, can be enormously powerful for offline optimization problems. One such application area is tuning gain schedules offline, given plant model with uncertainties.

Key Characteristics: Evolutionary algorithms are typically used on large scale problems in an offline setting. They provide a useful tool for finding globally optimal solutions in a space of large variables, especially with complex and non-smooth functions describing the objectives and constraints.

Existence of Solution – Some optimization problems are constrained to the point where no feasible space exists. The constraints are sufficiently complex, though, that the non-existence of a solution cannot be known until a solution is attempted. For high-dimensional problems with non-convex and/or non-smooth constraints, global methods must be used to search the existence of a valid solution. In these cases, a significant amount of computational effort may be required only to conclude that no solution exists.

Injection of Randomness — The crossover and mutation operations are inherently random in nature. Random perturbations are explicitly introduced in order to create new candidate solution sets.

Premature Convergence — EAs are susceptible to premature convergence, where the algorithm converges with a suboptimal solution before reaching the global optimum.

5.11 Natural Language Processing

Natural language processing (NLP) is composed of computational methods that read or listen to human language, and analyze the input to interpret its meaning. This process requires NLP methods to recognize both the syntax and the semantics of a language. Understanding the syntax involves the identification of token words or phrases from whichever medium the information is conveyed in, whether it be digital characters, handwriting, or speech. Understanding the semantics involves interpreting the ordered collection of those words and phrases to develop higher order concepts and associations.

Most modern NLP algorithms are based on statistical machine learning, in which rules governing the language are automatically learned through large example data sets. Of the large number of different applications within the field of NLP, two are potentially useful to civil aviation.

Speech Recognition Determine the textual representation of speech. This might be used to enable pilots to speak messages that can be transmitted to ground control in the form of text messages.

Machine Translation Automatically translate text or spoken words from one human language to another. This could be used in international flights to ease the language barrier between the pilot and ground control.

Information Extraction Extract semantic information from text. This technique could be used to enable voice operation of certain ATM or onboard aircraft functions. It could also provide an automated mechanism for rapidly checking the validity and safety of voice-issued directives from air traffic controllers.

Key Characteristics: Most of the algorithms used to perform NLP are based on machine learning, and so they inherit the same set of characteristics from those methods. Another characteristic unique to NLP is the possible ambiguity of interpretations:

Ambiguity — The extraction of semantics from symbolic text can lead to many possible interpretations. In the absence of more context, it is not possible to determine which interpretation is correct, leaving ambiguous results. This creates a challenge for testing and validation, as the notion of correctness is not clearly defined.

5.12 Summary of Key Characteristics

In each of the AI methods discussed, we have called out a few key characteristics. We focused on those characteristics that may pose challenges to using each method in the operational setting of civil aviation, which has requirements for safety assurance, resource constraints, and time-critical performance. The full set of identified characteristics can largely be organized into the following main categories.

Existence of Solution – Is there a solution?

With planning, model-checking, and optimization problems, the existence of a solution is not guaranteed. For larger, more complex problems, it may take a significant amount of computational resources and time to reach this conclusion.

Convergence – How long will it take to reach the solution?

Many of the AI methods are recursive or iterative in nature. The algorithm must converge to a usable solution within a reasonable amount of time in order to be useful in an operational setting. Some machine learning techniques have better convergence properties than others, and so this becomes an important design criteria. Neural networks are powerful tools for function approximation and classification, and can be used to implement the learning or inference component of other AI methods. However, neural networks do not have time-bounded guarantees that they will converge to an effective solution.

Ambiguity – Is the solution correct?

Ambiguity is built in to in a number of methods. In an expert system, the knowledge base consists of rules defined by experts, but they are often subjective. If two experts were to define rules or thresholds differently, which version is correct? Fuzzy logic and qualitative physics explicitly deal with ambiguous or imprecise concepts. They assign values to fuzzy or qualitative variables based on observed input data, but this assignment is subjective and arbitrary. In natural language processing, the interpretation of words and phrases can often have multiple semantic meanings, leading to ambiguous results. In all of these cases, the inability to precisely prescribe correctness or incorrectness to the information used in the algorithm makes it difficult to test and validate.

Randomness – Can the solution be repeated?

Several methods explicitly inject randomness as part of the algorithm. Evolutionary algorithms for optimization use a random process to perform mutations on individuals in the population that represent candidate solutions. Algorithms that implement the discovery process in reinforcement learning often apply random perturbations to ensure new parts of the search space are explored, in order to promote learning. SAT solvers, used in planning and sometimes as the inference engine for expert systems, can employ methods that utilize random numbers in an attempt to more efficiently navigate the search space. All of these methods use randomness as a tool to improve convergence, but the inherent drawback is that it prevents the algorithms from being repeatable. This can lead to challenges in testing and validation.

In the next section, we shift our focus to certification, and discuss how the key characteristics in both adaptive control and artificial intelligence can lead to specific certification challenges.

6 Certification Challenges

Having examined the characteristics of adaptive systems, we will next consider how these characteristics can lead to challenges in the certification process. Current civil certification processes are based on the idea that the correct behavior of a system must be completely specified and verified prior to operation. The fact that adaptive systems change their behavior at run-time is contrary to this idea in many ways. In general, many AI methods have unique characteristics that do not fit naturally within context of existing certification guidelines. This is due to the fact that the certification policies, conceived decades ago and still in use today, were not written with the needs and capabilities of AI in mind [27].

In this section we will use the characteristics of adaptive systems that have been described to identify categories of challenges that arise from those characteristics.

6.1 Comprehensive requirements.

One persistent challenge presented by adaptive systems is the need to define a comprehensive set of requirements for the intended behavior. The dynamic nature of these systems can make it difficult to specify exactly what they will do at run-time.

Complete description of desired behavior. Requirements must be measurably complete in the sense that execution of the test cases derived from them will provide complete structural coverage of the source code (up to the coverage metric required by the software assurance level). Creating a set of requirements that completely describes the desired system behavior and provides structural coverage can be difficult even for conventional systems. Defining such requirements for adaptive systems is likely the most common and difficult challenge that we have identified in this study.

Decomposition of requirements. System requirements must be decomposed and allocated to hardware and software in the system design. These allocated requirements must together guarantee that the system-level requirements are satisfied. Defining the necessary software requirements to guarantee the stability, convergence, and boundedness is a challenge that must be addressed for adaptive control algorithms.

6.2 Verifiable requirements

Assuming that an applicant is able to define a comprehensive set of requirements for an adaptive system, it may be difficult to verify those requirements.

Availability of verification method. One necessary characteristic of a good requirement is that it must be verifiable. This means that there must be some verification method available that is appropriate for the artifact and the requirements that are to be verified. Some of the components in AI algorithms may present challenges of this sort. What are the requirements for an inference engine and how should its behavior be verified? How can we show that a rule database in an expert system is correct? Furthermore, adaptive systems often involve discrete and non-linear operations that may now need to be verified analytically due to their complexity. Areas where new verification approaches may be required include:

- Verification of hybrid systems. This involves modeling the discrete and continuous domains together to verify the correctness of the overall system. It exercises the interaction of these two domains so a more realistic model of the overall system can be evaluated. Model checking of hybrid systems has the potential to address trust issues with certain types of non-linear behavior, and current research is continuing to improve scalability.

- Verification of non-linear systems. New decision procedures to reason about non-linear operations are being developed and incorporated into advanced model checking tools.

Well-defined behavior. Another important characteristic of a good requirement is that it must be possible to determine if a test (or analysis) produces the correct result. This may be difficult to define *a priori* for some adaptive systems.

Implementation language. The implementation language for AI software may present another challenge. Current certification guidance assumes the use of imperative languages like C/C++, while many AI algorithms are implemented in functional languages like Lisp or ML. We are not aware of any tools for computing coverage metrics for functional languages. In fact, it is not clear that structural coverage is a meaningful measure of completeness for functional languages.

Structural coverage. Structural coverage metrics are a key certification objective to demonstrate completeness of requirements, adequacy of test cases, and absence of unintended behaviors. If structural coverage cannot be obtained for the software in an adaptive system, some other methods will be needed to satisfy the underlying objectives. DO-333 provides some direction as to how this might be done using formal analysis in place of testing, but this has not been demonstrated for an adaptive system.

6.3 Documented Design

Many certification requirements amount to providing detailed documentation of the system. This may present difficulties for some adaptive systems if they were not initially developed with certification in mind.

Control of source code. Many AI systems are based on large open source libraries produced and revised over time by many different organizations. This software would need to be brought under active configuration control to prevent unauthorized changes that could invalidate verification results or add unintended functionality.

Traceability. Traceability (among requirements, test cases, and code) is an important element of the current certification process. Traceability may be difficult to implement when software is based on open source libraries developed by many different people and organizations.

6.4 Transparent Design

The certification process assumes a that perspicuous, transparent design and software implementation is being presented for evaluation. Since the certification authorities must be convinced that requirements have been met, they must (at least to some reasonable degree) be able to understand the artifacts and evidence that is brought before them. This may be challenging in several ways.

Deterministic behavior. As noted earlier in this report, we have to be careful how we use this term. In many cases, the adaptive algorithm itself is not nondeterministic. It is just handling uncertainty in the environment or the vehicle operating condition, in the same way that any control system responds to its sensed environment. On the other hand, there are some algorithms that make use of stochastic processes as part of their computations. This may even be acceptable if there are accompanying proofs of convergence or correct behavior. However, current certification process is based on an expectation that algorithms are deterministic; that is, the computed results, computation time, and resource utilization are predictable at design time. Any real lack of determinism will present challenges for current certification processes.

Conventional design artifacts. Many AI algorithms include unconventional design artifacts. By this we mean computing methods, data structures, or languages that are unfamiliar to certification authorities. Thus, there can be an expertise gap between system developers and regulators. This gap will make it difficult to convince regulators that an adaptive system based on unconventional methodologies satisfies certification requirements.

Complexity. The complexity of conventional software designs already stretches the capabilities of current verification technologies, and can be very difficult for a reviewer (or developer) to completely understand. Adaptive systems include embedded models, solvers, and new algorithms that greatly increase their complexity compared to conventional systems. Just the amount of software involved can present a significant challenge to understanding and verification, independent of any more sophisticated measures of complexity.

No unintended functionality. Another consequence of complexity is that it is more difficult to demonstrate the absence of unintended functionality. This is an important objective of the verification process.

6.5 Summary

These certification challenges are summarized in Table 1. The table also lists some of the adaptive algorithms that are likely to be impacted by each challenge.

The certification challenges faced by adaptive algorithms will also be related to their application within the aircraft. Based on an understanding of the characteristics of each algorithm we can attempt to categorizie them as to their expected function and then assess their criticality. Table 2 provides some possible applications for different algorithms and assigns a possible software level based on that functionality.

Table 1: Summary of Certification Challenges

Certification Challenges	**Description**	**Impacted Algorithms**
Complete description of the desired behavior (Comprehensive Requirements (CR))	Requirements that completely describe the behavior and provides structural coverage	Adaptive Control (AC), Learning methods, Evolutionary Algorithms(EA), Planning and Scheduling(PS), Qualitative Physics (QP)
Guaranteeing performance and good behavior (CR)	Requirements that guarantee convergence, stability and boundedness	MRAC,L1-AC, Machine Learning, Cognitive Architecture (CA), PS , EA, Neural Nets(NN)
Availability of Verification method (Verifiable Requirements (VR))	Requirements should be verifiable by some method (review, analysis, or testing)	Expert Systems (ES), EA, ACT-R, Soar
Well defined behavior (VR)	Determining by test or analysis that the system produces the correct result	Fuzzy Logic, QP, Natural Language Processing (NLP)
Implementation Language (VR)	The language used for implementation is suitable for verification activities	NLP
Structural Coverage (VR)	Structural coverage metrics are used to demonstrate completeness of requirements, adequacy of test cases, and absence of unindented behaviors.	AC, ES, EA
Control of source code (Documented Design (DD))	Control over changes to and integration of open source software	Expert Systems
Traceablity(DD)	Tracing design, code, test cases to requirements	AC, ES, EA, NN
Deterministic behavior(Transparent Design (TD))	Behavior that does not involve any randomness in computing the future state of the system	NN, EA,
Conventional design artifacts(TD)	Design artifacts that have been traditionally used and known to regulators	MRAC, MIAC, L1-AC, PS, ES, EA, CA
Complexity(TD)	The sophisticated ways in which the adaptive algorithms interact with the system	Expert Systems, EA, NN,
No unintended functionality(TD)	Guaranteeing that the algorithms do not interact with the system leading to unintended functionality	IAC, Machine Vision, NN

Table 2: Example applications for adpative algorithms

Algorithm	**Example Application**	**Possible Software Level**
L1-Adaptive Control, Direct MRAC	Flight Control and Fault Management	Level A
Indirect adaptive control, Adaptive control with Neural Networks	Flight Control	Level A
Cognitive architecture with no learning	Rule-based automated procedures	Level C
Reinforcement Learning	Pilot advisory system (weather, etc.)	Level D
Expert Systems/Cognitive architectures with learning	Pilot advisory system	Level D
Evolutionary algorithms	Flight planning and optimization of activities	Level B or C
Mission Planning	Flight planning	Level B or C
Neural Network	Pilot advisory system	Level D
Computer vision	Synthetic vision	Level A or B
Qualitative Physics	Flight planning and conflict avoidance	Level A or B
Natural Language Processing	Pilot and ATC interface	Level A

7 Mitigation Strategies

The general issues surrounding certification of adaptive systems technologies for aviation were discussed in Section 6. Here we identify several potential approaches to dealing with these issues.

There are three questions to be asked in approaching certification of an adaptive system:

1. Does the algorithm provide a significant benefit (in terms of safety or performance) compared to current approaches? In other words, is it providing a benefit that cannot be accomplished using current approaches? If not, an implementation based on traditional (non-adaptive) technology is preferable and there is no need to proceed further.

2. Is the algorithm dependable? Can we actually demonstrate the safety of an aircraft that relies on this algorithm? There must be some rational basis for believing that the technology does what it claims to do. If not, then there are fundamental issues with use of this technology in a safety-critical application apart from certification considerations.

3. Can we produce a convincing argument that the algorithm is dependable? There may be barriers in the current certification process to producing or presenting this evidence, but in principle, we should be able to modify our processes to accommodate this if there are clear benefits to doing so.

If we have answered the first two questions affirmatively, then it is worth considering what strategies might allow us to achieve certification for an that aircraft includes adaptive systems. The following sections describe mitigation strategies that may provide a way forward for some of the adaptive / intelligent systems we have evaluated. Certification of adaptive / intelligent system may require combination of these strategies.

7.1 Education

We have seen that there can be an expertise gap between developers and regulators when it comes to adopting new technologies. In fact, the commercial aviation industry is itself very conservative and (for good reason) usually reluctant to switch to the latest technology. However, we are convinced that for some adaptive algorithms this reluctance is unwarranted. Some approaches to adaptive control (such as those based on the L1 methodology) have been proven to be dependable and predictable in flight tests, and there seem to be no actual barriers to their certification. In fact, the very purpose of the adaptive component in these cases is to make the aircraft *safer to fly* when unsafe conditions are encountered. The roadblocks appear to consist of misunderstandings and misapplied descriptions (such as “nondeterministic”). In this case, no changes to the certification process are required. We would overcome the *perceived* barrier with a combination of education and demonstrations that the new technology can be certified using current processes.

7.2 Modified Certification Standards

Current standards assume static behavior specifications for aircraft functions. It may be possible to relax this assumption and other constraints in a principled way. The goal here would be to modify our existing standards in a way that retains the underlying safety principles, but also permits a more dynamic software structure. In some ways, this can be seen as analogous to the work undertaken in developing DO-178C to accommodate new technologies such as formal methods and model-based software development. The same approach could be used to provide new guidance in the form of a Technology Supplement directed toward certain classes of adaptive systems.

7.3 New Certification Methods

Current standards impose a fixed and implicit rationale for system safety. For example, [30] describes work to identify and make explicit the assurance case argument underlying DO-178C. Researchers in the UK [21] have previously explored ways to justify the substitution of one technology for another while maintaining the same level of safety provided by DO-178B. The main idea was to show that the new technology provided evidence that was at least as convincing as the previous technology in terms of the underlying (implicit) safety case. Certification approaches based on the development of a safety case for the aircraft (including its adaptive components) would in principle provide more flexibility to use advanced algorithms, demonstrating the safety of the adaptive algorithm by using the most appropriate evidence, while not sacrificing safety. However, there is much work to be done before applicants would have sufficient expertise to produce an accurate and trustworthy safety case, and regulators would be prepared to evaluate one.

7.4 New Verification Approaches

Current test-based verification processes will never be sufficient to assess the behavior of adaptive systems. Factors such as software size, complexity, unconventional artifacts, probabilistic computations, and large state spaces have been discussed as reasons for the difficulty of testing. Testing will have to be replaced or augmented by analysis based on formal methods or other mathematical techniques from the control theory or computer science domains. An example of a verification approach not currently allowed would be a probabilistic analysis that shows that an algorithm is sufficiently likely to converge to a solution within a given deadline.

7.5 Architectural Mitigations with Certification Support

Suppose that we are to certify an adaptive function that provides some advanced capability related to improved performance or recovery from a failure or upset, but we are unable to verify the behavior of the function with the required level of assurance. In some cases it may be possible to bound the behavior of the adaptive function by relying on a carefully designed system architecture. The key idea is to be able to treat the adaptive system differently based on when it executes (e.g., during different phases of flight).

One approach is called the *simplex* architecture [14]. It relies on three smaller, high-assurance functions: a system status monitor, a simpler backup for the adaptive function, and a switching function. During normal operation, outputs from the adaptive function are used by the rest of the system. If the monitor detects that the adaptive function is not behaving correctly (e.g., it has not converged, or computed new output before its deadline) or the system as a whole is approaching a state in which correct behavior of the adaptive function has not been verified, then the system will switch to using outputs from the simpler backup function. There are a number of technical challenges in this approach related to defining the switching boundary and blending smoothly from the adaptive function to the backup. For example, unwanted and potentially unsafe transient effects could be introduced if the transition mechanism were not designed properly. However, the inherent advantage in this approach is that, due to the architecture design, the safety of the vehicle never depends soley upon the adaptive function. The adaptive function is used during “normal” operating conditions and switched off during “abnormal” conditions when it might not be dependable.

An alternative approach uses a complex adaptive function to recover the vehicle in the case of a catastrophic failure or upset condition. In this case there is a conventional system that is used during normal flight operation, and a high-assurance monitor and switch that only invokes the adaptive system when the vehicle would otherwise be destroyed. The function of the monitor is to guarantee that the adaptive function is never used during normal operations. This is similar in concept to an airbag system in an automobile. In contrast to the first approach, the adaptive function is switched off during “normal” operating conditions and only switched on during “abnormal” conditions (when the vehicle would be lost anyway).

The same methodology is employed in control systems with adaptive augmentation. Using this control architecture, a traditional robust inner-loop controller is designed and effectively controls the system when the dynamics are close to the expected model. When the observed dynamics deviate significantly from the expected model, the traditional controller is no longer capable of providing the desired performance, resulting in noticeable tracking errors. At this point, the adaptive controller begins to provide a more dominant contribution to restore desired performance.

While such architectures have been demonstrated and could be implemented with high-assurance, the current certification process does not allow for this type of reasoning about time-varying levels of assurance.

7.6 Paradigm Shift: Licensing

Perhaps the most revolutionary approach to certification for advanced, adaptive techniques is to depart entirely from the current paradigm and instead emulate the techniques used to train and approve human pilots or operators. Pilots are licensed to fly based on demonstrating knowledge and skill through hundreds of hours of training and evaluation [4]. Similarly, humans performing other critical tasks such as air traffic control are trained and tested extensively before they enter an operational role. Extending this licensing procedure to autonomous software would lead to an analogous system of gained trust. Certification would be eventually attained through extensive, though not exhaustive, demonstration of knowledge and skill by the advanced software systems.

This approach has several key advantages:

Performance focus. A licensing approach would focus more resources on the actual proven performance of a system than its development methodology/process. One of the criticisms of DO-178C is that it focuses more on the the development process and producing evidence of compliance, than on evaluation of the resulting software. According to this argument, the current process invests thousands of person-hours in creating reams of costly, deeply intertwined documentation that may or may not have a direct impact on system safety. In a licensing approach, the investment emphasis would shift towards more extensive training, testing, and revision of the actual safety-critical system. High-fidelity simulations are already used extensively to test both human and synthetic systems; a licensing approach would just increase this focus.

Reduced cost. Once high-fidelity simulations are developed to a sufficient level to support the bulk of training and testing autonomous systems, the cost of retesting and re-licensing a new or revised system would become dramatically lower than current certification costs.

Reduced stagnation/concretization. Currently, certified avionics systems are essentially cast in stone the moment they are certified. It can cost more than $1M just to “open the box” and consider making a change to a certified system, because of the costs associated with re-certification. Furthermore, the implied fault-free guarantee of “certification” provides a strong incentive for developers to *not* continue testing or validating a deployed system— any fault they find could potentially make them liable for prior incidents: *knowledge* of the fault is worse than ignorance. General Motors’ experience with ignition switches provides a recent example.

Realistic expectations and reduced liability. A licensed system would not have an implied guarantee of perfection, it would have a proven track record of performance. Properly legislated, this could relieve a system developer from the legal liability that threatens all advanced technologies, much as parents are not held liable for the driving accidents that their licensed children cause. Already, self-driving cars are proving safer than human-driven cars [70]. Every year, more than 35,000 people are killed in America alone in car accidents. If deploying autonomous cars could cut that in half, it would be a huge societal gain. But that will not happen if the car manufacturers can be held liable for the remaining 17,000 deaths. Errors by human pilots are at least a contributory factor in most airplane crashes. How can we make it safe for a company to deploy an autonomous system that could avoid most of those crashes, but not all of them?

The licensing barrier could be much higher for an autonomous system than a human, of course, because the costs are non-recurring. Once a piece of software is class-certified to fly a particular aircraft, for example, it can be copied essentially for free into all other similar aircraft. This is simply not possible with humans– we cannot duplicate a skilled senior pilot, and it takes decades to grow another one. So instead of requiring a few hundred hours of simulation training and live flight before licensing, as with people, an autonomous system might be required to fly for hundreds of thousands of simulated hours, and thousands of real hours, encountering thousands or millions of faults and contingencies, demonstrating competency far beyond what any human could possibly show in a lifetime.

As with any testing-based validation, one key problem with a licensing approach is that any test-based evidence of acceptable behavior may be completely invalidated by a change to the system. Human abilities are remarkably robust to changing inputs– a person may experience a momentary muscle spasm or a power glitch in a control system, and will still retain the skills and abilities he had earlier. Only major medical difficulties might interfere with a pilot’s ability to fly. For example, you cannot disable a pilot simply by saying something to him or her. In contrast, we can reach into the “brain” of a software system and tweak it, and the consequences can be vast and (in the worst case) unintentional. A single line change to a software system may have broad, inadvertent consequences. Consequently, evolving or changing software is much more hazardous than exposing a human to changes, or even asking a human to change. Hence there is a need for relatively inexpensive, high-fidelity simulation testing environments that can be used to essentially re-test a full range of desired behaviors in both nominal and off-nominal situations, prior to deployment of any revised system.

8 Roadmap and Future Directions

We have surveyed current work in adaptive and intelligent systems, including both adaptive control schemes and those based on AI techniques. Based on the survey we have identified different characteristics of the adaptive algorithms and linked them to aspects of the current certification process that could prevent deployment of these systems. We have proposed several possible mitigation strategies that could provide alternative paths to eventual certification (or some equivalent demonstration of compliance with safety regulations) if that is determined to be desirable and appropriate.

In this section, we conclude with comments on two related workshops and a roadmap that puts these adaptive systems and mitigation strategies on a timeline. The goal is to suggest a way forward, targeting the "low-hanging fruit" first and identifying longer range research needs to address more challenging problems.

8.1 Related Workshops

Certification of adaptive systems is currently an active area of inquiry, motivated in part by the demand for UAVs with greater autonomy in both military and civil applications. AFRL is working on roadmap specific to autonomy certification as part of a series of workshops they began in 2013 called "Test and Evaluation, Verification and Validation of Autonomous Systems (TEV&V)." [4]. The stated goal of this workshop, which was split across industry, academia, and government, was to "identify, understand and categorize the unique challenges to the certification of safety critical autonomous systems by identifying the Verification and Validation (V&V) approaches needed to overcome them." Members of our project team participated in the first workshop focused on industry. The workshop dealt with a number similar concerns to those we have documented here.

Some interesting components of the vision from the TEV&V workshop include:

- Fast certification and recertification, including system changes
- Self-testing systems: monitor and expand own performance envelope
- Integrated safeguards and active and continuous V&V (embedded in instrumentation)
- Autonomy that can be "trained" and certified like a pilot

Additional information about this workshop is provided in Appendix A.

We organized a special session on Certification Considerations for Adaptive Systems as part of the March 2014 meeting of the Aerospace Control and Guidance Systems Committee (ACGSC). The objective of the session was to gather input from researchers and practitioners in the aerospace guidance and control community on adaptive systems being planned or developed, and to discuss the challenges that these new systems might face related to certification. The session was a great success with lively discussion among the participants. The ACGSC workshop helped shape our thinking about a roadmap to investigate certification of adaptive systems. Key outcomes from the workshop include:

- Enhanced awareness of certification concerns in the aerospace guidance and control community.
- Examples of relevant adaptive control and AI algorithms that are being developed for aerospace applications.
- Idenitification of L1 adaptive control as a good candidate for a near-term proof-of-concept demonstration.
- Discussion of licensing for autonomy algorithms as a viable alternative approach to certification.

8.2 Roadmap for Certification of Adaptive Systems

Based on our survey of adaptive and intelligent systems, we have grouped the different approaches into four categories based on the severity of changes to the certification process that they would require. We have ordered these categories ranging from presently certifiable to very challenging to serve as a roadmap for future investigation. The four categories are:

No Change to Certification Process. Based on discussions with a number of researchers and industry control engineers, we believe that L1 adaptive control can be implemented as part of an aircraft control system and certified with no changes to the current process. This methodology eliminates some of the barriers to certification that may be found in other adaptive control approaches, specifically by decoupling estimation and control, and ensuring that time delay margin is bounded away from zero. The barriers that remain appear to be related to misunderstandings based on incorrect generalizations about all adpative control algorithms, and the need to develop a complete set of requirements appropriate for an L1 control system. We recommend that a demonstration project be undertaken to produce the certification evidence for an example system based on the L1 methodology to illustrate what this might look like in practice.

Minor Modification of Certification Process. Approaches that are based upon architectural mitigations may be realizable with relatively minor modifications to the certification process. In these examples, the new components that guarantee adpative systems are only executed during the intended flight conditions could be verified and certified using traditional processes. The only change required would be acceptance of this approach to permit adaptive functions with lower levels of demonstrated assurance to operate as part of a system architected to provide high assurance. An example of such a system is an MRAC controller in which the excitation signal is only applied for a limited time after fault detection. Operational characteristics of the system while the excitation signal is being applied may not normally be acceptable, but in this proposed architecture it would be allowed if guaranteed to occur only under the prescribed fault detection and recovery scenario.

Major Modification of Certification Process. Some approaches that we have discussed will require more extensive changes to the certification process. An example that fits in this category is the use of run-time verification to guarantee that the outputs of a planning algorithm are safe for the guidance system to enact. The onboard use of complex verification software or inference engines is not anticipated at all by current standards. Assurance of this approach would apply the standards for onboard software development to components that would normally be dealt with using tool qualification guidance. It would also be necessary to determine how run-time verification failures are handled in the absence of human supervision.

Paradigm Shift: Licensing. A license-based approach to autonomous control systems is clearly the most radical departure from the current certification regime. However, recent developments in automous cars show that this approach may find public and governmental acceptance in the near future. Further study is needed to identify requirements and constraints that might be necessary to apply a license-based approach to automous aircraft, and to achieve a level of dependability and confidence that is at least as high as achieved by current processes.

This roadmap is illustrated in Figure 9 with examples of adaptive algorithms within these categories.

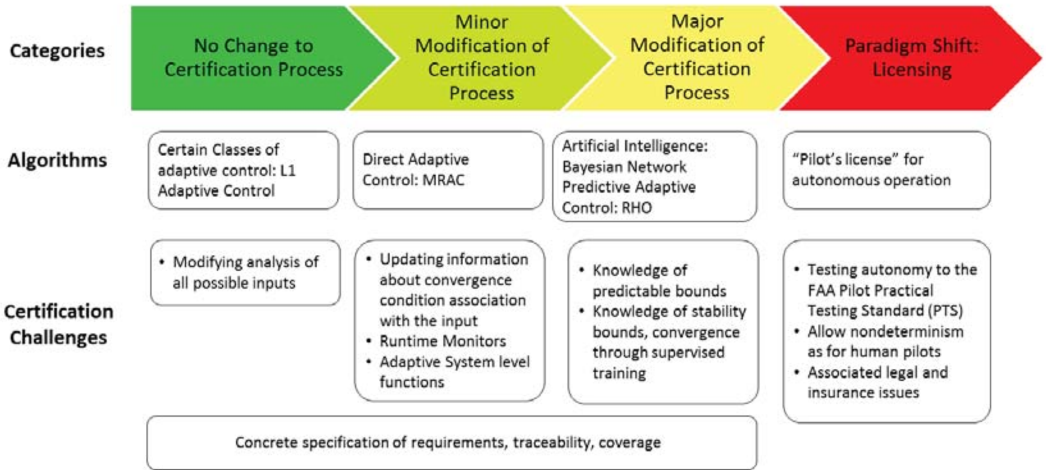
Categories
No Change to Certification Process
Minor Modification of Certification Process
Major Modification of Certification Process
Paradigm Shift: Licensing
Algorithms
Certain Classes of adaptive control: L1 Adaptive Control
Direct Adaptive Control: MRAC
Artificial Intelligence: Bayesian Network Predictive Adaptive Control: RHO
"Pilot's license" for autonomous operation
Certification Challenges
• Modifying analysis of all possible inputs
• Updating information about convergence condition association with the input
• Runtime Monitors
• Adaptive System level functions
• Knowledge of predictable bounds
• Knowledge of stability bounds, convergence through supervised training
• Testing autonomy to the FAA Pilot Practical Testing Standard (PTS)
• Allow nondeterminism as for human pilots
• Associated legal and insurance issues
Concrete specification of requirements, traceability, coverage

Figure 9: Roadmap

9 List of Acronyms

ADI — Adaptive Dynamic Inversion

ADS-B — Automatic Dependent Surveillance Broadcast

AI — Artificial Intelligence

BDI — Belief, Desire, Intention

CATMT — Collaborative Air Traffic Management Technologies

CSS-Wx — Common Support Services - Weather

DAI — Distributed Artificial Intelligence

EA — Evolutionary Algorithm

FMS — Flight Management System

GTM — Generic Transport Model

MIAC — Model Identification Adaptive Control

MRAC — Model Reference Adaptive Control

NAS — National Air Space

NGATS — Next Generation Air Transportation System

SIFT — Smart Information Flow Technologies is a small research company specializing in intelligent automation and human-centered systems.

SLAM — Simultaneous Localization and Mapping

SWIM — System Wide Information Management

TCAS — Terminal Collision Avoidance System

TEV&V — Test, Evaluation, Verification and Validation

V&V — Verification and Validation

10 Glossary

Adaptive Control — A control policy with 1) parameters that may be adjusted, and 2) some mechanism for automatically adjusting those parameters online based on measured performance.

Adaptive System — The computational element of the active feedback process changes, in order to maintain desired performance, in response to failures, threats, or a changing environment.

Artificial Intelligence — A broad class of computational methods that are designed to operate with intelligence, primarily by 1) learning from experience, and 2) making decisions based on learned information to achieve a goal.

Adaptation Method — The algorithm that governs how online parameter estimation is conducted in an adaptive control framework.

Direct Adaptive Control — The system uncertainty parameters are estimated and used directly in the control law to cancel the tracking error.

Indirect Adaptive Control — A system identification process estimates uncertain model parameters. These parameters are then used as inputs to a control design method that computes the controller gains.

Non-Determinism — In general, a nondeterministic algorithm is one in which the output cannot be repeated with certainty, given the same input. Section 1.1.3 describes four types of nondeterminism: Environmental nondeterminism, probabilistic algorithms, uncertain existence of solutions, and concurrency.

Overfitting — The set of independent basis functions used to fit a given set of data is too complex. This results in the model fitting random noise which is only present for the supplied data sets, causing it to be a poor approximation of new data.

11 References

[1] *Minutes from Future of Aerospace Workshop, Integration Discussion*, 2013.

[2] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," *Advances in neural information processing systems*, vol. 19, pp. 1–8, 2007.

[3] AFRL. *Test and Evaluation, Verification and Validation of Autonomous Systems: Industry Exploration Workshop, Meeting Report*, October 2013.

[4] AFRL. *Test and Evaluation, Verification and Validation of Autonomous Systems: Challenge Exploration Final Report*, May 2014.

[5] K. J. Åström and B. Wittenmark, *Adaptive control*, Courier Dover Publications, 2008.

[6] A. Barr, P. Cohen, and E. Feigenbaum, *The Handbook of Artificial Intelligence*, Addison Wesley, 1981.

[7] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta, "Artificial vision in road vehicles," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1258–1271, Jul 2002. doi:10.1109/JPROC.2002.801444.

[8] J. Boskovic and R. Mehra, "Performance analysis of a simple L1-adaptive controller," in *American Control Conference (ACC), 2013*, pp. 3370–3375, June 2013.

[9] S. F. Campbell and J. T. Kaneshige, "A nonlinear dynamic inversion L 1 adaptive controller for a Generic Transport Model," in *American Control Conference (ACC), 2010*, pp. 862–867. IEEE, 2010.

[10] C. Cao and N. Hovakimyan, "L1 adaptive output-feedback controller for non-strictly-positive-real reference systems: missile longitudinal autopilot design," *Journal of guidance, control, and dynamics*, vol. 32, no. 3, pp. 717–726, 2009.

[11] C. Cao and N. Hovakimyan, "Design and analysis of a novel L1 adaptive controller, Part I: Control signal and asymptotic stability," in *American Control Conference, 2006*, pp. 3397–3402. IEEE, 2006. doi:10.1109/ACC.2006.1657243.

[12] C. Cao and N. Hovakimyan, "Design and analysis of a novel L1 adaptive controller, Part II: Guaranteed transient performance," in *American Control Conference, 2006*, pp. 3403–3408. IEEE, June 2006. doi:10.1109/ACC.2006.1657244.

[13] S. Chien, R. Sherwood, D. Tran, B. Cichy, G. Rabideau, *et al.*, “The EO-1 Autonomous Science Agent,” in *Int’l Joint Conf. on Autonomous Agents and Multiagent Systems*, pp. 420–427, 2004.

[14] T. L. Crenshaw, E. L. Gunter, C. L. Robinson, L. Sha, and P. R. Kumar, “The Simplex Reference Model: Limiting Fault-Propagation Due to Unreliable Components in Cyber-Physical System Architectures,” in *Proceedings of the 28th IEEE Real-Time Systems Symposium (RTSS 2007), 3-6 December 2007, Tucson, Arizona, USA*, pp. 400–412, 2007.

[15] DOD. *Unmanned Aircraft System Airspace Integration Plan*, 2011.

[16] A. Dorobantu, P. Seiler, and G. Balas, “Time-delay margin analysis for an adaptive controller,” *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 5, pp. 1418–1425, 2012. URL: http://arc.aiaa.org/doi/pdf/10.2514/1.56719, doi:10.2514/1.56719.

[17] R. Enns and J. Si, “Apache Helicopter Stabilization Using Neural Dynamic Programming,”, vol. 25, no. 1, pp. 19–25, 2002. doi:10.2514/2.4870.

[18] FAA, “Integration of Civil UAS in the NAS Roadmap,” Technical report, FAA, 2013. http://www.faa.gov/about/initiatives/uas/media/UAS_Roadmap_2013.pdf.

[19] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” *AI magazine*, vol. 17, no. 3, pp. 37, 1996.

[20] M. Fisher, L. Dennis, and M. Webster, “Verifying Autonomous Systems,” *Communications of the ACM*, vol. 56, no. 9, pp. 84–93, 2013. doi:10.1145/2494558.

[21] A. Galloway, R. Paige, N. Tudor, R. Weaver, I. Toyn, and J. McDermid, “Proof vs testing in the context of safety standards,” in *Digital Avionics Systems Conference*, 2005. doi:10.1109/DASC.2005.1563405.

[22] E. Gat, “Autonomy software verification and validation might not be as hard as it seems,” in *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, volume 5, pp. 3123–3128, March 2004. doi:10.1109/AERO.2004.1368117.

[23] I. Gregory, C. Cao, E. Xargay, N. Hovakimyan, and X. Zou, “L1 adaptive control design for NASA AirSTAR flight test vehicle,” in *AIAA Guidance, Navigation, and Control Conference*, volume 5738, 2009. doi:10.2514/6.2009-573.

[24] I. Gregory, E. Xargay, C. Cao, and N. Hovakimyan, "Flight test of L1 adaptive controller on the NASA AirSTAR flight test vehicle," in *AIAA Guidance, Navigation and Control Conference*, 2010. doi:10.2514/6.2010-8015.

[25] J. Guo and G. Tao, "A feedback-based fault detection scheme for aircraft systems with damage," in *Control Conference (ASCC), 2011 8th Asian*, pp. 1431–1436, May 2011.

[26] J. Guo, Y. Liu, and G. Tao, "A multivariable MRAC design using state feedback for linearized aircraft models with damage," in *American Control Conference (ACC), 2010*, pp. 2095–2100, June 2010.

[27] L. Harrison, P. Saunders, and J. Janowitz, "Artificial Intelligence with Applications for Aircraft.," Technical report, DTIC Document, 1994.

[28] K. Hayhurst, J. Maddalon, P. Miner, M. DeWalt, and G. McCormick, "Unmanned aircraft hazards and their implications for regulation," in *25th Digital Avionics Systems Conference, 2006 IEEE/AIAA*, pp. 1–12. IEEE, 2006.

[29] J. Hinchman, M. Clark, J. Hoffman, B. Hulbert, and C. Snyder, "Towards Safety Assurance of Trusted Autonomy in Air Force Flight Critical Systems," *Computer Security Applications Conference, Layered Assurance Workshop*, 2012.

[30] C. Holloway, "Towards Understanding the DO-178C / ED-12C Assurance Case," in *7th International IET System Safety Conference, Incorporating the Cyber Security Conference*, pp. 15–18, October 2012.

[31] N. Hovakimyan, E. Lavretsky, and C. Cao, "Adaptive dynamic inversion via time-scale separation," *Neural Networks, IEEE Transactions on*, vol. 19, no. 10, pp. 1702–1711, 2008.

[32] N. Hovakimyan. *Paper with Unsubstantiated and Wrong Claims about L1 Adaptive Control*. http://naira.mechse.illinois.edu/paper-misperceptions-l1/, 2014.

[33] N. Hovakimyan and C. Cao, *L1 adaptive control theory: Guaranteed robustness with fast adaptation*, volume 21, SIAM, 2010.

[34] P. Ioannou and J. Sun, *Robust adaptive control*, Dover Publications, 2012.

[35] S. Jacklin, "Closing the certification gaps in adaptive flight control software," in *Proc. AIAA Guidance, Navigation and Control Conf*, 2008.

[36] S. Jacklin, M. Lowry, J. Schumann, P. Gupta, J. Bosworth, E. Zavala, J. Kelly, K. Hayhurst, C. Belcastro, and C. Belcastro, "Verification, validation, and certification challenges for adaptive flight-critical control system software," in *American Institute of Aeronautics and Astronautics (AIAA) Guidance, Navigation, and Control Conference and Exhibit*, pp. 16–19, 2004.

[37] S. A. Jacklin, "Certification of Safety-Critical Software Under DO-178C and DO-278A," in *AIAA Infotech Aerospace Conference 2012, Garden Grove, CA*. American Institute of Aeronautics and Astronautics (AIAA), 2012. URL: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20120016835_2012017661.pdf.

[38] M. D. Johnston, "Spike: AI Scheduling for Hubble Space Telescope After 18 Months of Orbital Operations," in *Proc. of the 1992 AAAI Spring Symposium on Practical Approaches to Scheduling and Planning*, pp. 1–5, Stanford, CA, 1992.

[39] D. Jourdan, M. Piedmont, V. Gavrilets, D. Vos, and J. McCormick, "Enhancing UAV Survivability through Damage Tolerant Control," in *AIAA Guidance, Navigation and Control Conference*, number AIAA 2010-7548, August 2010. doi:10.2514/6.2010-7548.

[40] A. Kornecki and J. Zalewski, "Software Certification for Safety-Critical Systems: A Status Report," in *Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on*, pp. 665–672. IEEE, 2008.

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[42] K. Kumar and G. Thakur, "Advanced Applications of Neural Networks and Artificial Intelligence: A Review," *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 4, no. 6, pp. 57, 2012.

[43] A. Lampton, A. Niksch, and J. Valasek, "Reinforcement Learning of a Morphing Airfoil-Policy and Discrete Learning Analysis," *Journal of Aerospace Computing, Information, and Communication*, vol. 7, no. 8, pp. 241–260, 2010. doi:10.2514/1.48057.

[44] I. Landau, R. Lozano, M. M'Saad, and A. Karimi, *Adaptive Control: Algorithms, Analysis and Applications*, Springer, 2011.

[45] E. Lavretsky and N. Hovakimyan, "Adaptive dynamic inversion for nonaffine-in-control systems via time-scale separation: Part II," in *American Control Conference, 2005. Proceedings of the 2005*, pp. 3548–3553. IEEE, 2005.

[46] D. J. Lee and H. Bang, “Autonomous Autorotation of an Unmanned Helicopter Using a Reinforcement Learning Algorithm,” *Journal of Aerospace Information Systems*, vol. 10, no. 2, pp. 98–104, 2013. doi:10.2514/1.48253.

[47] L. Ma and Y. Zhang, “DUKF-based GTM UAV fault detection and diagnosis with nonlinear and LPV models,” in *Mechatronics and Embedded Systems and Applications (MESA), 2010 IEEE/ASME International Conference on*, pp. 375–380, July 2010. doi:10.1109/MESA.2010.5552042.

[48] D. Maiti, J. Guo, and G. Tao, “A discrete-time multivariable state feedback MRAC design with application to linearized aircraft models with damage,” in *American Control Conference (ACC), 2011*, pp. 606–611, June 2011.

[49] M. McFarland and A. Calise, “Adaptive nonlinear control of agile antiair missiles using neural networks.,” *IEEE Trans. Contr. Sys. Techn.*, vol. 8, no. 5, pp. 749–756, 2000. URL: http://dblp.uni-trier.de/db/journals/tcst/tcst8.html#McFarlandC00.

[50] M. B. McFarland and A. J. Calise, “Adaptive nonlinear control of agile antiair missiles using neural networks,” *IEEE Transactions on Control Systems Technology*, vol. 8, no. 5, pp. 749–756, 2000.

[51] A. Mili, B. Cukic, Y. Liu, and R. Ayed, “Towards the verification and validation of online learning adaptive systems,” in *Software Engineering with Computational Intelligence*, pp. 173–203, Springer, 2003.

[52] A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, “Autonomous inverted helicopter flight via reinforcement learning,” in *Experimental Robotics IX*, pp. 363–372, Springer, 2006.

[53] S. C. of RTCA. *DO-178C, Software Considerations in Airborne Systems and Equipment Certification*, 2011.

[54] I. J. Olson, A. J. Ten Harmsel, and E. M. Atkins, “Safe landing planning for an energy-constrained multicopter,” in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pp. 1225–1235. IEEE, 2014.

[55] C. on Autonomy Research for Civil Aviation; Aeronautics, S. E. B. D. on Engineering, and P. S. N. R. Council, *Autonomy Research for Civil Aviation: Toward a New Era of Flight*, The National Academies Press, 2014. URL: http://www.nap.edu/openbook.php?record_id=18815.

[56] J. E. L. Pat Langley and S. Rogers, “Cognitive architectures: Research issues and challenges,” *Cognitive Systems Research*, vol. 10, no. 2, pp. 141–160, 2009. doi:10.1016/j.cogsys.2006.07.004.

[57] V. Patel, C. Cao, N. Hovakimyan, K. Wise, and E. Lavretsky, "L1 Adaptive Controller for Tailless Unstable Aircraft," in *American Control Conference, 2007. ACC '07*, pp. 5272–5277, July 2007. doi:10.1109/ACC.2007.4283114.

[58] C. Pecheur. *Verification and validation of autonomy software at NASA*, 2000.

[59] T. Peni, B. Vanek, Z. Szabo, P. Gaspar, and J. Bokor, "Supervisory fault tolerant control of the GTM UAV using LPV methods," in *Control and Fault-Tolerant Systems (SysTol), 2013 Conference on*, pp. 655–660, Oct 2013. doi:10.1109/SysTol.2013.6693938.

[60] D. Poole, A. Mackworth, and R. Goebel, *Computational intelligence - a logical approach.*, Oxford University Press, 1998.

[61] E. Rich, *Artificial Intelligence*, McGraw-Hill, 1983.

[62] J. Rushby, "A safety-case approach for certifying adaptive systems," in *AIAA Infotech@ Aerospace Conference*, 2009.

[63] J. Rushby, "New Challenges in Certification for Aircraft Software," in *Proceedings of the Ninth ACM International Conference on Embedded Software*, pp. 211–218. ACM, 2011. URL: http://www.csl.sri.com/~rushby/papers/emsoft11.pdf.

[64] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *arXiv preprint arXiv:1409.0575*, 2014.

[65] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2010.

[66] Q. Sang and G. Tao, "Adaptive control of piecewise linear systems with applications to NASA GTM," in *American Control Conference (ACC), 2011*, pp. 1157–1162, June 2011.

[67] M. Sharma, E. Lavretsky, and K. Wise, "Application and Flight Testing of an Adaptive Autopilot on Precision Guided Munitions," in *AIAA Guidance, Navigation and Control Conference*, number AIAA-2006-6568. AIAA, August 2006. doi:doi:10.2514/6.2006-6568.

[68] M. Sharma and A. J. Calise, "Neural-network augmentation of existing linear controllers," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 1, pp. 12–19, 2005.

[69] H. Shuaib, R. Anthony, and M. Pelc, "A framework for certifying autonomic computing systems," in *ICAS 2011, The Seventh International Conference on Autonomic and Autonomous Systems*, pp. 122–127, 2011.

[70] T. Simonite, “Data Shows Google’s Robot Cars Are Smoother, Safer Drivers Than You or I,”, October 2013. URL: http://www.technologyreview.com/news/520746/data-shows-googles-robot-cars-are-smoother-safer-drivers-than-you-or-i/.

[71] J. Slaby, L. Welch, and P. Work, “Toward certification of adaptive distributed systems,” in *Real-time and Embedded Systems Workshop*, 2006.

[72] D. Spencer, “Applying artificial intelligence techniques to air traffic control automation,” *The Lincoln Laboratory Journal*, vol. 2, pp. 537–554, 1989.

[73] N. N. T., “Robust Optimal Adaptive Control Method with Large Adaptive Gain,” in *American Control Conference, 2009*, pp. 2581–2586, Aug 2009. doi:0.1109/ACC.2009.5159904.

[74] G. Tallant, J. Buffington, W. Storm, P. Stanfill, and B. Krogh, “Validation & verification for emerging avionic systems,” in *National Workshop on Aviation Software Systems: Design for Certifiably Dependable Systems*, oct 2006.

[75] J. Valasek, D. T. Monish, and J. Rong, “A Reinforcement Learning - Adaptive Control Architecture for Morphing,” *Journal of Aerospace Computing, Information, and Communication*, vol. 2, no. 4, pp. 174–195, 2005.

[76] J. Wang, V. Patel, C. Cao, N. Hovakimyan, and E. Lavretsky, “Novel L1 adaptive control methodology for aerial refueling with guaranteed transient performance,” *Journal of guidance, control, and dynamics*, vol. 31, no. 1, pp. 182–193, 2008.

[77] J. Wang, V. V. Patel, C. Cao, N. Hovakimyan, and E. Lavretsky, “L1 adaptive neural network controller for autonomous aerial refueling with guaranteed transient performance,” in *AIAA Guidance, Navigation, and Control Conference, Keystone, CO*, 2006.

[78] D. Ward, J. Monaco, R. Barron, R. Bird, J. Virnig, and T. Landers, “Self Designing Controller: Design, Simulation, and Flight Test Evaluation,” *BAI, Lockheed Martin, Calspan AFRL Final Report, WL-TR-97-3095, November*, 1996.

[79] S. Waslander, G. Hoffmann, J. S. Jang, and C. Tomlin, “Multi-agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning,” in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 3712–3717, Aug 2005. doi:10.1109/IROS.2005.1545025.

[80] M. Webster, N. Cameron, M. Jump, and M. Fisher, "Towards Certification of Autonomous Unmanned Aircraft Using Formal Model Checking and Simulation,", 2012.

[81] M. Webster, M. Fisher, N. Cameron, and M. Jump, "Formal Methods for the Certification of Autonomous Unmanned Aircraft Systems," in *Proceedings of the 30th International Conference on Computer Safety, Reliability, and Security*, SAFECOMP'11, pp. 228–242, Berlin, Heidelberg, 2011, Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=2041619.2041644.

[82] C. Wilkinson, J. Lynch, R. Bharadwaj, and K. Woodham, "Verification of Adaptive Systems," Technical report, FAA, 2013.

[83] K. Wise, "Autonomous Systems: The Future in Aerospace," in *2014 AIAA Intelligent Systems Workshop*, Aug. 2014.

[84] K. Wise, E. Lavretsky, and N. Hovakimyan, "Adaptive Control of Flight: Theory, Applications, and Open Problems," in *American Control Conference, 2006*, pp. 5966–5971. IEEE, 2006. doi:10.1109/ACC.2006.1657677.

[85] K. A. Wise, J. S. Brinker, A. J. Calise, D. F. Enns, M. R. Elgersma, and P. Voulgaris, "Direct adaptive reconfigurable flight control for a tailless advanced fighter aircraft," *International Journal of Robust and Nonlinear Control*, vol. 9, no. 14, pp. 999–1012, 1999.

[86] B. Wittenmark, "Adaptive dual control," *Control Systems, Robotics and Automation, Encyclopedia of Life Support Systems (EOLSS), Developed under the auspices of the UNESCO*, 2002.

[87] R. Wolfe. *MTSI's UAS Airspace Integration Competencies*, 2011.

[88] N. Wu, M. Ruschmann, J. Huang, and K. Huang, "Probing the NASA Generic Transport Aircraft in Real-Time for Health Monitoring," *Control Systems Technology, IEEE Transactions on*, vol. 19, no. 5, pp. 1288–1295, Sept 2011. doi:10.1109/TCST.2010.2067215.

[89] F. Zotes, *Application of Intelligent Algorithms to Aerospace Problems*, PhD thesis, Universidad Nacional de Educación a Distancia, 2011.

Appendix A

Background

A.1 Recent History of Research Programs with Adaptive Flight Control

The field of adaptive control has matured over the last 50+ years, with important contributions from a large number of researchers. The full body of work encompassing the historical development of adaptive and intelligent flight control is well beyond the scope of this book. However, we have identified a few specific programs where noteworthy advancements have been made, and summarize them here.

A.1.1 Damage Tolerant Control

Rockwell Collins developed Damage Tolerant Control (DTC) technology to mitigate common failures for combat aircraft, such as primary control surface damage, airframe damage, and complete propulsion system failure. Under a DARPA-sponsored program, a series of flight tests were performed to showcase four key aspects of the DTC technology. The all-attitude autopilot formulation was shown to enable aerobatic maneuvers and automatic recovery.from unusual attitudes, while enforcing flow angle and load limits. The model-reference adaptive control (MRAC) module, with short-term persistent excitation, demonstrated the ability of DTC to recover baseline controller performance after losing actuation (on the aileron, rudder, and/or elevator). Automatic Supervisory Adaptive Control (ASAC) returned an aircraft with catastrophic wing damage to trimmed and controllable flight within a few seconds, enabling the vehicle to complete its mission and perform an autonomous landing. Finally the Emergency Mission Management System (EMMS) provides the ability for an aircraft which suffered complete engine failure to glide back onto a feasible landing trajectory. The combination of these modules dramatically enhances the survivability of combat aircraft [39].

A.1.2 SDC - Self-Designing Controller

The SDC was an AFRL program with flight tests conducted in 1996. An indirect form of adaptive control was flown in the VISTA / F-16 aircraft. Real-time parameter identification was performed with a simulated failure of a missing horizontal tail. They performed modified sequential least squares for online parameter estimation. A receding horizon optimal control law was implemented, which was updated in real-time based on the model parameter estimates. Flight-tests showed that the handling qualities were sufficient to land the aircraft [84].

A.1.3 IFCS - Intelligent Flight Control System

The IFCS program, led by Boeing in the late 1990's, developed an indirect adaptive control system using neural networks to learn (or recursively estimate) several aerodynamic coefficients to form linear models of the aircraft dynamics. Optimal feedback gains were then computed based on these linear models by solving the Riccati equation online [84].

A.1.4 RESTORE - Reconfigurable Control for Tailless Fighter Aircraft

RESTORE was a program sponsored by the Air Force Research Laboratory in the 1990's aimed at developing reconfigurable flight software capable of adapting to unknown failures and damage [85]. The Boeing-led effort developed applied a dynamic inversion control law as the base control system, and used an on-line neural network to adaptively regulate the errors resulting from nonlinear plant inversion. In addition, an on-line system identification method was used to estimate the control derivatives used by the control allocation algorithms. The system identification used a least-squares estimation method, and applied filters to the pilot commands in order to ensure the measured signals contained sufficient information to estimate the plant parameters. This effectively provided persistent excitation, a necessary condition for convergence of the learning process in adaptive control. The algorithms developed under the RESTORE program were flight-tested on the NASA X-36 tailless aircraft with two successful flights in 1998.

A.1.5 JDAM - Joint Direct Attack Munition

The Joint Direct Attack Munitions program began in the late 1990's with the purpose of developing guidance kits to retrofit existing un-guided munitions. The same approach used in the RESTORE program was applied by Boeing to the JDAM MK-84 munition, which was demonstrated in at least two successful flight tests. The production LQR based flight controller was replaced entirely with a new scheme that used dynamic inversion for inner loop control, augmented with a direct adaptive controller using a neural network. This approach precluded the need for any gain-scheduling or wind-tunnel testing. Instead, the adaptive system was designed for a single point in the flight envelope using general aerodynamic data from the missile DATCOM data sheet. In addition, the neural network was only trained online, during flight [68, 84].

A.1.6 IRAC - Integrated Resilient Aircraft Control

The Integrated Resilient Aircraft Control (IRAC) project began at NASA in late 2006 as part of the broader and ongoing Aviation Safety Program. The purpose of IRAC is to research new flight control methods capable of providing "onboard control resilience" to ensure safe flight is maintained in the face of adverse conditions. Flight tests for various research flight control laws have been conducted on NASA Dryden's F/A-18A systems research aircraft.

Research objectives for IRAC include:

- Improve stability and maneuverability for safe landing using adaptive controls
- Develop validated and integrated aircraft control design tools for safe flight under adverse conditions
- Improve understanding of the dynamics of loss-of-control (LOC) incidents and design control software to help regain control
- Enhance engine models to improve engine response during LOC incidents

One of the key components of IRAC is to examine adaptive control technology as a solution for improving operational safety of the aircraft in off-nominal, adverse flight conditions. To this end, several different applications of adaptive control have been tested using the NASA Generic Transport Model (GTM) [9], and flight tests have been conducted on the AirSTAR [24].

A.1.7 AirSTAR - Airborne Subscale Transport Aircraft Research

The NASA AirSTAR system, operated at NASA Langley Research Center, is a 5.5% dynamically scaled version of the Generic Transport Model aircraft. It is powered with twin turbines, has a 6 foot wingspan, weighs approximately 54 lbs at takeoff, and is fully equipped with flight test instrumentation.

The AirSTAR platform has been used to flight-test several different adaptive control methods from different research groups. In particular, flight tests with L1 adaptive control demonstrated the ability to prevent loss of control at high angles of attack and maintain pilot handling qualities throughout a large portion of the flight envelope without gain scheduling [24].

A.2 Previous Studies

Several different studies have been conducted in recent years on the topics of applying artificial intelligence, adaptive methods and autonomous systems to civil aviation. Although a complete review of all such studies is beyond the scope of this report, we do provide an overview of the most relevant ones, most of which have been published within the last year.

A.2.1 Artificial Intelligence with Applications for Aircraft, 1994

In 1994, the FAA issued a report entitled "Artificial Intelligence with Applications for Aircraft" [27]. This report focused primarily on three types of methods: neural networks (NN), expert systems (ES), and fuzzy logic. At the time the report was written, the early 1990's, these were among the most popular techniques being studied in the rapidly evolving field of AI. Several specific applications of these methods are discussed, along with some attention to the issues of certification and human factors. Neural networks were also identified as a technology to support intelligent monitoring and diagnostics, but most of the report focuses on expert systems. In particular, various types of expert systems are described as candidate tools to support navigation, flight management, and diversion routing.

Even though the state of the art in AI and aviation has grown significantly in the last twenty years, we still face the same fundamental issues regarding safety, V&V, and certification. The authors point out that the unique characteristics apparent in developing and testing AI-based software are not addressed in RTCA/DO-178B, the governing document in aviation certification. Standardization across different areas of AI was recommended for easing the certification process in general. Specific recommendations include a common run time inference engine to standardize expert system shells, and the adoption of a standard language, such as Common LISP. In addition, a thorough discussion is provided on verification and validation of expert systems, and how the process compares to that of conventional software.

The key issues related to certification of AI technologies were identified as follows:

- Learning - The use of AI based systems with learning components is deemed unlikely in civil aviation due to the apparent difficulty to demonstrate that they will maintain flight within the safe operating envelope.

- Safety - The issue of safety is based primarily on how AI is perceived. The application of a technology that is generally perceived as being mature and safe will be viewed as contributing to safety, rather than degrading it.

- Verification and Validation - For expert systems, the knowledge base must be complete and error free. A more general challenge is identifying a suitable set of tests to demonstrate safety.

- Design Guidance - The current FAA document, RTCA/DO-178B, does not accommodate AI-based methods. More work is required to develop meaningful design guidance for AI-based systems.

- Human Factors - With AI supplementing or replacing cognitive tasks of humans, the potential loss of situation awareness, loss of proficiency, and overconfidence in the AI system should be understood and mitigated.

A.2.2 VVIACS - Verification and Validation of Intelligent and Adaptive Control Systems, 2005

This was an AFRL funded study performed by Lockheed Martin and Carnegie Mellon University that ended in 2005. [74] The main goals of this research were to 1) classify emerging safety-critical control systems based upon characteristics that may challenge traditional certification, and to 2) identify V&V tools and processes with the potential to efficiently support these systems.

The LM/CMU team identified several emerging fundamental properties that are required for new safety-critical control systems, all of which must tested, evaluated, verified and validated in order to be fully certified. As a result, they project that cost, schedule and risk will grow at faster rates (possibly exponential) as complexity of the control system increases, with the main drivers being software, simulation, test, and test tool development. A key finding is that advanced V&V tools focused on the above drivers should significantly reduce the overall V&V cost and effort needed to support advanced safety-critical emerging control systems.

A.2.3 Verification of Adaptive Systems, 2013

In 2013, NASA Langley and Honeywell together delivered a report to the FAA entitled "Verification of Adaptive Systems" [82]. This report provides an overview of adaptive algorithms, focusing primarily on adaptive control, neural networks and genetic algorithms.

To motivate an analysis of how certification would be applied to an adaptive system, they used an example adaptive control structure that augmented traditional gain-scheduled control with a trigger. The outcome of this analysis was a table listing DO-178B/C objectives and a corresponding discussion of why those objectives might be hard to satisfy in the chosen AS example. Several of the cited difficulties are due to the AS learned state space making it more difficult to perform certain tasks that are required to support DO-178B/C objectives. The qualification of development tools and verification tools are also cited, as well as developing adequate test cases, and verifying that system level stability and convergence properties are retained through requirements decomposition and implementation.

In order to properly derive and validate safety and functional requirements for adaptive systems, the report recommends the use of structured, objective, evidence based safety cases. It is also recommended that safety properties should be specified explicitly for when adaptation may be engaged, bounding the allowable learned state-space, and the fall-back to a backup response if those safety bounds are exceeded.

The report also provides several recommendations for achieving requirements verification of adaptive systems, including:

- Use model based design techniques with well-defined syntax and semantics, with requirements expressed explicitly in the model
- Use formal methods to provide adequate test coverage and proof of safety and performance properties
- DO-178C and its supplements are necessary, as DO-178B is inadequate to provide software design assurance
- The certification process will need to rely more on verification by analysis and formal proofs rather than test

A.2.4 Test and Evaluation, Verification and Validation of Autonomous Systems, 2013

In 2013, the Air Force Research Laboratory (AFRL) began hosting a three-part workshop on "Test and Evaluation, Verification and Validation of Autonomous Systems". [4]. The stated goal of this workshop, which was split across industry, academia, and government, was to "identify, understand and categorize the unique challenges to the certification of safety critical autonomous systems by identifying the Verification and Validation (V&V) approaches needed to overcome them." Upon completion of the workshop, the identified challenges and complementary approaches are to be organized into 3-6 thrust areas that will feed the AFRL Autonomy Strategy and Research goals, as well as the DoD TEV&V portfolio.

The industry workshop, which included 23 participants from 14 different companies, focused on high-level obstacles to the adoption of automated systems in civil and military aviation. A substantial portion of time and discussion was focused on how to certify such systems, with an AFRL official citing that non-determinism and learning behavior represents the leading risk of future autonomous systems. The final report noted that the biggest takeaway by far from this industry workshop was the collective discussion of a new "licensing paradigm", in which the certification process for an autonomous system is reformed to be more like the licensing process used for human pilots.

In the academic workshop, which included 21 participants from 14 different universities, the group identified the biggest challenges they face in TEV&V. The list was refined into six broad categories: 1) Requirements, Models and Design; 2) Human-Machine Interaction; 3) Modeling and Simulation, Testing; 4) Runtime Assurance, Verification; 5) Highly Complex Interactive Autonomy; 6) Policy - User Impact. One of the interesting takeaways from this workshop was the suggestion that the certification process will become more like the process used on humans as automated systems become more human-like.

The government workshop, which included 18 participants from AFRL, identified a 2030 vision for what is needed in TEV&V of autonomy, along with 14 technical objectives to support that vision. Some interesting components of the 2030 vision include:

- Fast certification and recertification, including system changes
- Self-testing systems: monitor and expand own performance envelope
- Integrated safeguards and active and continuous V&V (embedded in instrumentation)
- Autonomy that can be “trained” and certified like a pilot

A.2.5 Autonomy Research for Civil Aviation: Toward a New Era of Flight, 2014

In July of 2014, the National Research Council issued a report entitled “Autonomy Research for Civil Aviation: Toward a New Era of Flight”. [55]. This report explores the use of “increasingly autonomous” (IA) systems in civil aviation, identifying unique characteristics, potential benefits, and barriers to implementation. The vision for integrating autonomy is described for three distinct platforms: crewed aircraft, unmanned aircraft, and air traffic management.

The council underscores the belief that while the aviation industry is on the threshold of profound change, due to the rapid growth of autonomous systems, it will be a significant challenge to integrate these new platforms and algorithms safely and efficiently into the national airspace. The authors also point out that existing certification criteria and processes do not properly account for the unique characteristics of advanced, increasingly autonomous systems. Within their list of 14 potential barriers to the adoption / integration of IA systems, we find: verification and validation, the certification process, and trust in adaptive/nondeterministic systems.

The report concludes with a suggested research agenda to address the technical challenges that underlie the implementation barriers. Some of council's final recommendations that are most pertinent to our report are to:

- Develop methodologies to characterize and bound the behavior of adaptive/nondeterministic systems over their complete life cycle.
- Develop standards and processes for the verification, validation, and certification of IA systems, and determine their implications for design.
- Determine how IA systems could enhance the safety and efficiency of civil aviation.
- Develop processes to engender broad stakeholder trust in IA systems for civil aviation.

Appendix B

Application Areas in Civil Aviation

In this appendix, we discuss a few different areas in civil aviation where adaptive and intelligent methods have the potential to provide significant benefits.

B.1 Air Traffic Management

Air Traffic Management (ATM) encompasses all of the systems and operations that enable aircraft to be actively monitored and directed by ground-based facilities from departure to arrival.

From gate to gate, the aircraft is handed off from one controlling authority to another. At the airport, control is divided into four main segments: Clearance Delivery, Ground Control, Local Control, and Terminal Control. Prior to departure, the planned route of the aircraft is approved by Clearance Delivery, which ensures that it has the proper route and slot time. The Ground Control segment directs all non-runway ground traffic, including taxiing between the gate and the runway. The Local Control (or Tower Control) segment is responsible for clearing the aircraft for takeoff and landing. Local weather conditions and runway separation constraints may lead Local Control to hold aircraft prior to landing, or divert them to another airport. Tracking and all other in-flight air traffic services in the vicinity of the airport are handled by the Terminal Control segment. A single Terminal Radar Approach Control (TRACON) facility can track aircraft in a 30-50 nautical mile radius, which may service multiple airports. The primary function of the Terminal Control segment is to ensure safe separation in congested airspace.

In general, ATM consists of several interconnected segments of supervisory control, where humans are tasked with monitoring status, planning and scheduling time-ordered actions, identifying conflicts, and then resolving those conflicts as they arise. Many of these tasks can be improved to be made safer and more efficient with through automation. Intelligent systems, such as expert systems, planners, and automated agents, clearly have potential applications in this area.

The Next Generation Air Transportation System (NGATS), commonly referred to as NextGen, includes a broad set of enhancements to the United States ATM infrastructure, as well as technology upgrades on the aircraft themselves. In particular, the following ATM-related components of NextGen may benefit from the application of adaptive and intelligent software:

Collaborative Air Traffic Management Technologies (CATMT) A suite of enhancements to the decision-support and data-sharing tools used by air traffic management personnel. The goals are to reduce operator workload and provide a more collaborative environment among controllers and operators, thereby improving efficiency in the National Airspace System. AI methods, such as expert systems and algorithms for planning and optimization, may be used as decision aid tools.

System Wide Information Management (SWIM) A network structure that will carry NextGen digital information. SWIM will enable cost-effective, real-time data exchange and sharing among users of the National Airspace System (NAS). AI methods that enable emergent behavior from decentralized agents may be used to enable efficient and robust operation of the SWIM network.

Common Support Services-Weather (CSS-Wx) A system that will provide the FAA and NAS users with same-time access to a unified aviation weather picture via the SWIM network. This will enable collaborative and dynamic decision making among all users of the NAS. AI methods for machine learning and data mining may be used to identify weather conditions and patterns that can impact certain classes of aircraft.

B.2 Route-Planning

One of the ongoing objectives in air traffic management is the notion of free-flight. This refers to the idealistic future capability of aircraft to be granted the freedom to plan and fly their own routes, independent of a centralized authority. Clearly, this raises safety concerns as it introduces a greater potential for in-air collisions. However, given the advancements of onboard technology such as GPS receivers and ADS-B transponders, and the overwhelming burden placed on a centralized air traffic control paradigm, the eventual shift towards a more decentralized system is certainly plausible.

When individual aircraft have more autonomy to determine their own flight path, this opens up a large decision space that is amenable to route-planning. Trajectory optimization methods can be used online to find minimum-fuel or minimum-time trajectories. These may change in non-trivial ways based on weather and aircraft loading conditions. The big airlines are particularly interested in taking more control of their route-planning, and having the freedom to adjust them dynamically, as this can add up to enormous fuel savings and better performance in terms of on-time arrivals.

B.3 Automatic Flight Control

Automatic flight control is commonplace on all commercial and private planes today. Conventional methods use classical control designs to provide robustness and desirable pilot handling qualities, but these methods are limited to operation within the nominal flight envelope. As was discussed in Chapter 4, the application of adaptive control technology has the potential to improve safety and performance by maintaining control even when the aircraft experiences damage, a failure, or departs from the safe operating condition.

B.4 Flight Management

In modern avionics, the flight management system (FMS) is responsible for automating a wide range of in-flight tasks. Generally speaking, the goals of the FMS are to reduce pilot workload, improve aircraft safety, and increase fuel efficiency.

AI related applications of the FMS would include health monitoring, fault detection, and diagnostics, which are discussed in separate sections below. Another area of interest is decision aid tools to recommend courses of action in response to flight critical events. One example is recommending the best place to land given diminished flight capability if the aircraft suffers a catastrophic failure or damage.

B.5 Collision Avoidance

Ensuring collision avoidance has become a rising challenge as the density of air traffic has increased in recent years. The standard system regulated by the FAA is called the Traffic Collision Avoidance System (TCAS). Components of TCAS include communications hardware, a processor, and a cockpit display.

TCAS regularly interrogates all other aircraft within a set range. The other aircraft reply with their altitude, and the interrogating aircraft computes their range and bearing. The TCAS processor uses other sensor inputs to generate traffic advisories and/or resolution advisories when another aircraft enters the protected airspace.

The TCAS system is currently in place and will be merged with new transponders of the class Automatic Dependent Surveillance-Broadcast (ADS-B). ADS-B uses a GPS receiver to determine its precise position, and automatically shares this data along with additional flight information on a broadcast frequency. The TCAS and ADS-B systems are compatible, in that the processing component of TCAS can utilize the information provided via ADS-B messages, rather than relying on a two-step interrogation / response.

Collision avoidance will become an increasingly important and challenging topic as unmanned air systems (UAS) are integrated into the national airspace. Already, several reports have been cited of near collisions between civilian passenger aircraft and remotely piloted air vehicles. A particular issue is the use of quad rotors, which can easily maintain flight by hovering for extended durations, creating a situation in which the pilot can easily retain control of the vehicle even when it is beyond the line of sight. This severely reduces or in some cases completely eliminates the pilot's situational awareness of the vehicle's environment. Given that automatic sense and avoid technology will almost certainly be a requirement for any UAS to be allowed into the controlled airspace, AI-based methods for machine vision and planning may become enabling technologies.

B.6 Fault Detection and Diagnostics

Within the aerospace industry, data mining can be a particularly useful tool for identifying and predicting failures. For example, as part of Flight Operations Quality Assurance, airlines regularly download vehicle health data from the flight data recorders and store it in a database. Applying data mining methods to this database can enable the discovery of important trends and precursors to component failures or other hazardous events.

The same approach can be used to optimize maintenance schedules, so that the projected risk of component failures is minimized. These are examples of offline applications to discover new knowledge about historical data, techniques that employ different types of data mining, which is rooted in machine learning. Once this type of analysis has been done, representations of the knowledge (trends, relationships, etc.) can be used onboard the aircraft to support the real-time execution of vehicle health monitoring, fault detection, and diagnostics.

B.7 Multi-UAV mission

Multi-UAV or multi-autonomous agent missions require adaptive or intelligent networking behavior to maintain connectivity to accomplish a mission. This can be accomplished by Cognitive Network Management System (CNMS) which is responsible for automated, policy-based real time network management for complex Mobile Adhoc Network (MANET) networks. CNMS reduces the workload for the operator in positioning of mobile networked devices for the following:

- maintaining connectivity
- increasing/maintaining the connection bandwidth
- deal with interference issues

- increasing the coverage area.

The CNMS supports fully distributed policy learning mechanisms [?] that allow CNMS to adapt at run-time to unanticipated network conditions and application requirements by creating and distributing new locally learned policies. CNMS will play a significant role in the execution of multi-UAV missions. CNMS also provides support for if/how to mitigate interference. This is done by selecting one among several strategies to reliably send the information to avoid jamming. This will specially be helpful in dealing with high interference regions of operation or under jamming attacks.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
01- 03 - 2015

2. REPORT TYPE
Contractor Report

3. DATES COVERED *(From - To)*
September 2013 - October 2014

4. TITLE AND SUBTITLE
Certification Considerations for Adaptive Systems

5a. CONTRACT NUMBER
NNL13AC67T

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Bhattacharyya, Siddhartha; Cofer, Darren; Musliner, David J.; Mueller, Joseph; Engstrom, Eric

5d. PROJECT NUMBER

5e. TASK NUMBER
NNL13AA05B

5f. WORK UNIT NUMBER
357672.04.04.07.02

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, Virginia 23681

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)
NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NASA/CR-2015-218702

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 63
Availability: NASA STI Program (757) 864-9658

13. SUPPLEMENTARY NOTES
Final Report

Langley Technical Monitor: Kelly J. Hayhurst

14. ABSTRACT

Advanced capabilities planned for the next generation of aircraft, including those that will operate within the Next Generation Air Transportation System NextGen), will necessarily include complex new algorithms and on-traditional software elements. These aircraft will likely incorporate adaptive control algorithms that will provide enhanced safety, autonomy, and robustness during adverse conditions. Unmanned aircraft will operate alongside manned aircraft in the National Airspace (NAS), with intelligent software performing the high-level decision-making functions normally performed by human pilots. Even human-piloted aircraft will necessarily include more autonomy. However, there are serious barriers to the deployment of new capabilities, especially for those based upon software including adaptive control (AC) and artificial intelligence (AI) algorithms. Current civil aviation certification processes are based on the idea that the correct behavior of a system must be completely specified and verified prior to operation. This report by Rockwell Collins and SIFT documents our comprehensive study of the state of the art in intelligent and adaptive algorithms for the civil aviation domain, categorizing the approaches used and identifying gaps and challenges associated with certification of each approach.

15. SUBJECT TERMS

Adaptive systems; Artificial intelligence; Certification

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	99	(443) 757-5802