Certification Considerations for Adaptive Systems

S. Bhattacharyya[1], D. Cofer[1], D. Musliner[2], J. Mueller[2], and E. Engstrom[2]

Abstract— **Advanced capabilities planned for the next generation of unmanned aircraft will be based on complex new algorithms and non-traditional software elements. These aircraft will incorporate adaptive and intelligent control algorithms that will provide enhanced safety, autonomy, and high-level decision-making functions normally performed by human pilots, as well as robustness in the presence of failures and adverse flight conditions.**

This paper discusses the characteristics of adaptive algorithms and the challenges they present to certification for operation in the National Airspace System (NAS). We provide mitigations strategies that may make it possible to overcome these challenges.

I. Introduction

There are serious barriers to the deployment of unmanned aircraft having the advanced capabilities necessary to operation safely and autonomously in the NAS, 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.

We begin with 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 DO-178C [12] as 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.

[1]Rockwell Collins Advanced Technology Center
[2]Smart Information Flow Technologies

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.

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.

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.

II. Adaptive Systems

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.

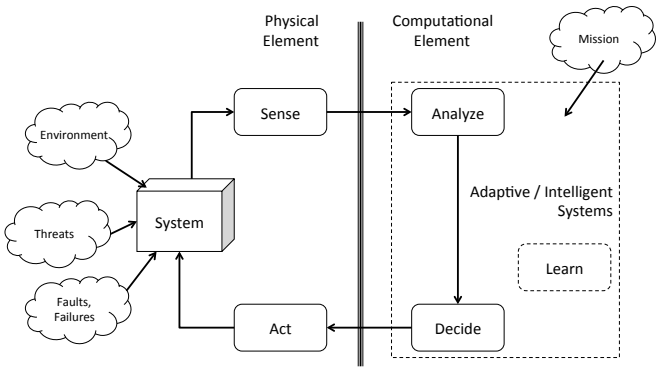

Fig. 1. The roles of an adaptive intelligent system in closed-loop control.

Traditionally, the computational element designed to control the system is fixed. The characteristics of the system may change significantly in response to changes in the mission, environment, threats, or failures— all of which are uncertain. However, the instructions that implement the "analyze and decide" steps in Figure 1 are developed in some software which is inherently deterministic, and that software is subsequently certified relative to the original system requirements. This is not the case for AC and AI systems.

1) Adaptive Control: Adaptive Control (AC) refers to 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 [2]. 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.

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 III.

2) Artificial Intelligence: Artificial Intelligence (AI) refers to 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. Two of the characteristics, *learning* and *adaptation*, are the main ingredients for both AC and AI methods, providing the functional basis for enhancing performance and robustness over traditional methods.

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. We have identified the following types of nondeterminism that may be relevant for our discussion:

a) 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.

b) 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.

c) 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.

d) 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.

In the next two sections, we describe key characteristics of various adaptive control and AI methods, and then examine why these characteristics lead to certification challenges.

III. Adaptive Control Algorithms

An adaptive controller, according to Astrom and Wittenmark [2], 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.

A. Gain Scheduled Control

A traditional feedback control structure with gain-scheduling is shown in Figure 2. With gain-scheduling, the adaptation method is a simple table lookup. In practice, it is multi-dimensional space discretized into a grid of distinct flight conditions, and a unique set of controller gains 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.

Gain scheduling is widely used in certified avionics applications. However, there are still potential assurance issues associated with it. For example, there are no performance guarantees in between design points. Furthermore, it is impractical to capture all possible model variations with a finite set of grid points.

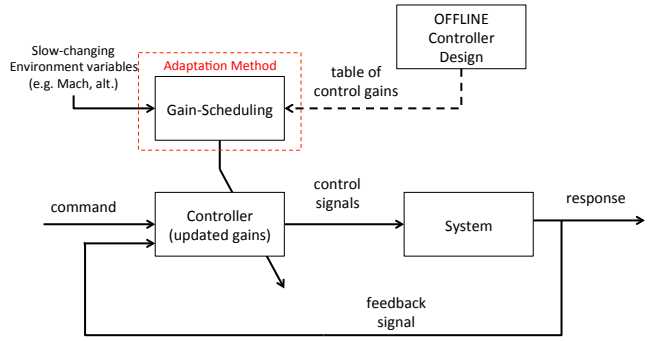

Fig. 2. Feedback Control with Gain Scheduling

B. Indirect Adaptive Control

A block diagram for indirect adaptive control (IAC)/model identification adaptive control (MIAC) is shown in Figure 3. 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.

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. In order for the adaptive system to provide the desired performance characteristics, the parameter estimation step must converge sufficiently fast. Characteristics of indirect adaptive control include:

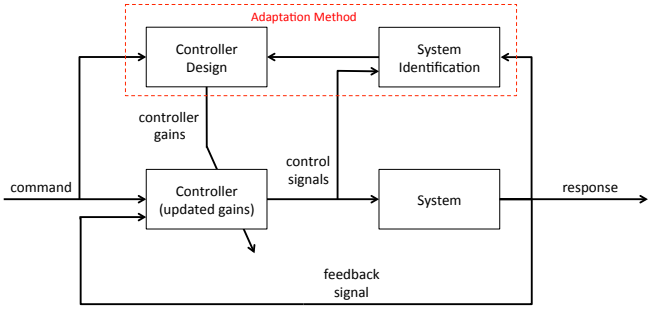

Fig. 3. Indirect Adaptive Control (IAC) / Model Identification Adaptive Control (MIAC)

Convergence rate: For indirect 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 [3]

Persistent excitation: The input signal must have properties for parameters to converge. Otherwise, the error goes to zero but the parameters do not necessarily converge to their correct values.

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.

C. Direct Model Reference Adaptive Control

Figure 4 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. Characteristics of MRAC include:

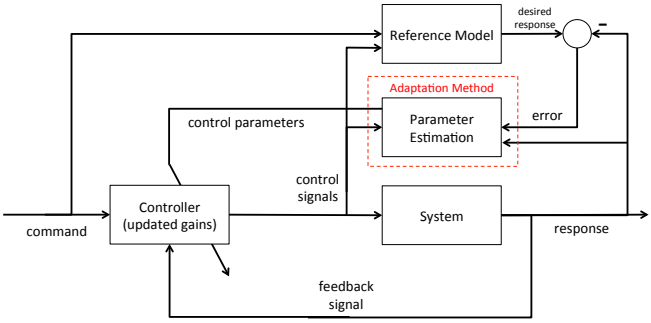

Fig. 4. Direct Model Reference Adaptive Control (MRAC)

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 [3].

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 specific properties for parameters to converge.

D. L1 Adaptive Control

A relatively new methodology called "L1 adaptive control" (or L1AC) addresses the challenge of simultaneously achieving both desired transient performance and guaranteed robustness properties [4]. One representation (though not representative of every possible architecture) of L1AC is shown in Figure 5. 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 at an appropriate location in the architecture, and the estimated control parameters are fed to the reference model, making it a state predictor.

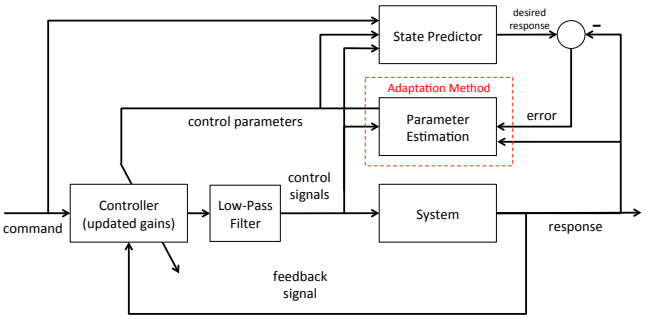

Fig. 5. L1AC - L1 Adaptive Control Structure

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. 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.

One possible issue associated with L1AC is related to its compuational 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.

E. Adaptive control with Neural Networks

The adaptation method for indirect adaptive control architectures can be executed with a neural network [5]. Neural networks (NN), which are discussed in Section IV-B, 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. Characteristics of NN include:

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.

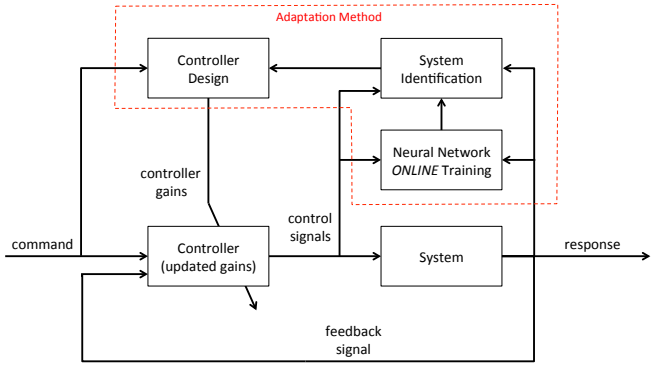

Fig. 6. Indirect Adaptive Control using a Neural Network with Online Training.

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

Adequacy of design The existence of solution is not guaranteed.

F. Adaptive Control Summary

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 several open problems [6], some of which have been addressed by modifying the architecture of some of the existing adaptive control technologies. The resulting approaches are L1-adaptive control, and combinations of direct and indirect approaches.

Remaining issues for adaptive control 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 *convergence* and *values of adaptive gain*. Online monitoring methods need to be deployed to check the safety bounds and assure stable transient response. Furthermore, the stability criteria for traditional control do not apply to adaptive control due to the nonlinear characteristics of adaptive control. 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 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. However, at a recent autonomy workshop hosted by the AIAA Intelligent Systems Technical Committee [7], it was observed that 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.

IV. Artificial Intelligence Algorithms

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. In this section, we provide an overview of several different AI methods and algorithms.

A. 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.

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. 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. Reinforcement learning 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. Characteristics of machine learning may include:

Stochastic processes 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.

B. Neural Networks

Neural networks (NN) are computational models that are inspired by the operation of the nervous system.

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.

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. Characteristics include:

Overfitting This is a general phenomenon that can impact any type of function approximator, including NN. A useful analogy is using a higher order polynomial to fit data when the trend is adequately captured by the lower order model.

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.

C. 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. Structurally, an ES is composed of 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 applies rules from propositional logic and uses search methods, such as forward chaining or backward chaining, to compute solutions based on information in the knowledge base. Characteristics of ES include:

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 for the inference engine to work with. *Subjectivity of knowledge base* Multiple experts may 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* 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. *Stochastic behavior* Some algorithms used to implement the inference engine explicitly use random numbers to search for satisfying solutions.

D. 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. According to fuzzy logic, elements can have a partial "degree of membership" within a 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]. The application of fuzzy logic in control systems has been proposed for a variety of applications, in aviation [8].

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.

E. Planning and Scheduling

Automated planning and scheduling systems are designed to find strategies or action sequences that achieve set 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. Planning problems can often also be reduced to equivalent satisfiability and model checking problems, so solvers from those research areas can also be used.

Planning and scheduling systems have been successfully deployed in numerous real-world applications, including control of satellites and space telescopes. However, they generally include critical aspects that would make them challenging to certify. 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. Many planning problems have no solution so it is not possible to know whether a solution exists without actually solving the problem.

F. 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 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. Some examples of applications for civil applications are trajectory optimization, departure and arrival scheduling in air traffic control, and large scale coordinated route planning by airlines. Relevant characteristics of EA include:

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.

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.

G. Artificial Intelligence Summary

There are many other AI methods that we have not covered, including Cognitive Architectures, Machine Vision, Qualitative Physics, and Natural Language Processing. These are addressed in the full technical report for the project.

Each of these AI methods have characteristics that make them challenging to use in the operational setting of civil aviation, which has requirements for safety assurance, resource constraints, and time-critical performance.

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.

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.

Is the solution correct? Ambiguity is built in to in a number of methods. The inability to precisely prescribe correctness or incorrectness to the information used in the algorithm or the results of the algorithm makes it difficult to test and validate.

Can the solution be repeated? Several methods explicitly inject randomness as part of the algorithm to improve convergence and discovery, but the inherent drawback is that it prevents the algorithms from being repeatable. This can lead to challenges in testing and validation.

V. 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 [8].

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.

A. 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.

1) 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.

2) 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.

B. 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.

1) 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.

2) 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.

3) 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.

4) 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.

C. 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.

1) 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.

2) 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.

D. 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.

1) 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.

2) 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.

3) 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.

4) 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.

VI. Mitigation Strategies

In this section we discuss several potential approaches to dealing with these certification challenges. 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 systems we have evaluated. Certification of adaptive systems may require a combination of these strategies.

A. Education

In our experience, 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.

B. 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.

C. New Certification Methods

Current standards impose a fixed and implicit rationale for system safety. For example, [10] describes work to identify and make explicit the assurance case argument underlying DO-178C. Researchers in the UK [9] 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.

D. 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.

E. 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 [11]. 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).

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.

F. 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. 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. 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.

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

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. 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, even if it does not eliminate 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.

VII. Conclusions

Certification of adaptive systems is currently an active area of research, motivated in part by the demand for UAVs with greater autonomy in both military and civil applications. 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.

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. Approaches that are based upon architectural mitigations may be realizable with relatively minor modifications to the certification process. Some approaches that we have discussed will require more extensive changes to the certification process, such as the use of run-time verification to guarantee that the outputs of a planning algorithm are safe for the guidance system to enact. A license-based approach to autonomous control systems is clearly the most radical departure from the current certification regime. However, recent developments in autonomous cars show that this approach may find public and governmental acceptance in the near future.

ACKNOWLEDGMENT

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.

VIII. References

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

[2] Åström, Karl Johan and Wittenmark, Björn, Adaptive control, Courier Dover Publications, 2008.

[3] Nguyen Nhan T., Robust Optimal Adaptive Control Method with Large Adaptive Gain, Aug, 2009.

[4] Cao, Chengyu and Hovakimyan, Naira, Design and analysis of a novel L1 adaptive controller, Part I: Control signal and asymptotic stability, American Control Conference, 2006.

[5] McFarland, M. and Calise, A., Adaptive nonlinear control of agile antiair missiles using neural networks.IEEE Trans. Contr. Sys. Techn., 2000.

[6] Wise, Kevin and Lavretsky, Eugene and Hovakimyan, Naira, Adaptive Control of Flight: Theory, Applications, and Open Problems, American Control Conference, 2006.

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

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

[9] 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.

[10] 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.

[11] 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.

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