This paper is a preprint of a paper accepted by the Engineering & Technology Reference and is subject to Institution of Engineering and Technology Copyright. The final published version is available at the IET Digital Library
DOI: 10.1049/etr.2014.2015.0009

Safety and Certification of Unmanned Air Systems

Charles Patchett, Mike Jump, Michael Fisher

University of Liverpool

January 2015

Abstract

The problem of assessing and certifying *Unmanned Air Systems* (UAS) is addressed and an overview of their safety requirements provided. Whilst not a comprehensive study, current directions, references to key aspects of the literature, and opinions concerning the technologies and directions that might be used in the future are provided. Overall, this article covers what is known as “safe to fly” regulations, as captured by certification requirements and guidelines for system design, as well as “flown safely” regulations, as captured by flight procedures and operations.

Introduction

Over the last 20 years, the use of UAS in military contexts has increased significantly. However, even given the technology lag from military to non-military contexts, this increase has not been mirrored by a corresponding rise in civilian UAS usage. There is considerable demand for UAS within the civil aviation world, yet there remain crucial barriers to the legal and safe integration of UAS into civil airspace. The primary problem concerns UAS *certification* and, in particular, the impasse it presents. UAS *designers* require clear and precise guidelines from regulators before they proceed to construct a UAS; but *regulators* require an existing system that they can examine in order to determine its certifiability! Beyond this certification problem, there is the issue of *operational regulations*. In the case of manned air systems, these regulations have gradually evolved over the 100+ years of powered manned flight. However, the regulations for UAS remain to be defined. Finally, before UAS can be allowed in civilian airspace, questions over UAS safety with respect to other air users and the general public have to be satisfactorily answered. Although we can never expect (or mandate) that any UAS is absolutely safe, there is a level of acceptable safety that will be expected by society. In addition, “safety” goes beyond the design of the vehicle and incorporates how it is flown. The concept of “Safe to Fly – Flown Safely” captures these dual aspects and it is this that is addressed.

The term UAS reflects that the system comprises an aircraft and a remote control station and that the remote pilot may not be in contact with the aircraft which may be under the (temporary) control of an on-board autonomous system. This article specifically considers UAS that require certification, namely those weighing over 150kg, and involving civil operations in unrestricted airspace, where safety is paramount.

Generally “safety” means “an absence of danger”. Absolute safety, or absolute absence of danger, is generally an unachievable or at least a very expensive goal. Therefore the concept of acceptable levels of safety has been adopted in many risk bearing industries, including aviation. The term describes an event with a probability of occurrence and consequences that are acceptable to society, i.e. the society is willing to take or be subjected to the risk that the event might bring.

At first sight, the UAS certification problem seems straightforward: just treat the UAS as a manned aircraft and certificate it according to current, manned aircraft, regulations. Unfortunately, UAS are fundamentally different and the current regulations are often inappropriate. These regulations, developed over a long period, all operate under the assumption that there is a *pilot* on board, in control, and responsible for safe operation. For example, existing operational rules require air users to “See and Avoid” other aircraft. Of all the differences between manned and unmanned flight, satisfactorily replacing the pilot with technology that can perform in an equivalent manner remains the key.

As with many autonomous systems, UAS can be flown at varying levels of autonomy (though, in all these, the pilot(s), driver(s), or operator(s) are all remote from the vehicle).

- *Manually Piloted* – the aircraft is remotely controlled from the ground control station with all key decisions being carried out by the operator; such operation involves a high level of skill and, in addition to the skilled operator, secure, fast, and reliable communication is required.

- *Supervised Operation* – the aircraft is flown autonomously by the on-board autopilot, directed by the human operator, but without the necessity of real time feedback to the operator.

- *Autonomous Operation* – the aircraft is flown according to high-level objectives outlined by the operator, but with the on-board autonomous control system making most of the significant decisions. (Note: the International Civil Aviation Authority does not envisage fully autonomous operation in the foreseeable future [1])

These varying levels of autonomy have elements that are common to each other. Yet they are clearly very different to traditional piloted aircraft, with many of the safety and reliability issues needing to be handled on-board, rather than by the pilot.

The United Kingdom is addressing the issue of UAS civil operation through the ASTRAEA programme. The ASTRAEA programme aims to enable the routine use of UAS in all classes of airspace without the need for restrictive or specialised conditions of operation.

Certification: Aerospace Standards for Software & Hardware

To provide a context for our discussion, we will briefly describe some of the wide range of standards and requirements for air systems. What is considered to be (acceptably) safe, with respect to flight operations is prescribed by the relevant regulatory authority. In the UK, this body is the Civil Aviation Authority (CAA) which works closely with the European Aviation Safety Authority (EASA) who now publish Certification Specifications the Joint Aviation Requirements (CSJARs). The general airworthiness regulations for civil aircraft are covered in Sections 23 (concerning Light/Commuter vehicles) and 25 (Transport vehicles) of these CSJARs. Since medium to large UAS are under consideration, it is assumed that medium UAS are covered by an equivalent document, yet to be produced, to CSJAR 23 and large ones, by an equivalent to CSJAR 25.

There are two particularly relevant paragraphs in the CSJARs: 1301 and 1309.

Para 1301, Function and Installation states that installed equipment must "*be of a kind and design appropriate to its intended function….and function properly when installed*". In other words it must be "fit for purpose". This paragraph is designed to prevent systems from other vehicles being used, without proper modification and subsequent re-certification, in aircraft.

Para. 1309 Equipment, Systems and Installations, states simply, "*this requires justification that all probable failures, or combinations of failures, will not result in unacceptable consequences"*. This requires the identification of failure probabilities, including multiple failures, by detailed analysis of essential systems and evaluation of the consequences of those failures. In particular, it requires that the frequency of occurrence (probability) of system failures must be inversely proportional to the severity of the effects. These consequences are typically categorised as *minor*, *major*, *hazardous*, or *catastrophic* [2] and are defined in ACJs [3] to CS23/25.1309. Crucially, the more severe the consequence, the lower the probability of failure is required to be. So, a minor failure is declared acceptable at a probability of occurrence of $<10^{-3}$ per flight hour, a major failure level is $<10^{-5}$ and a hazardous failure $<10^{-7}$. The worst case, a catastrophic failure, must occur no more than once every 1,000,000,000 flight hours i.e. $<10^{-9}$ per flight hour. Associated with these requirements are the levels of software development assurance mandated, with potentially catastrophic failures requiring the highest.

While there is no finalised 1309 for UAS, it is widely assumed that the above principles would also apply. However, 1309 is primarily concerned with keeping passenger aircraft airborne or landing them safely. When we come to UAS, they will likely not (yet) have passengers and clearly have no crew. In addition, while we will require that the UAS land without endangering people on the ground, it may be that the vehicle is designed to crash safely! So, 1309 is problematic even before we come to the question of how to assess UAS compliance with 1309.

Why is the analysis required for compliance so problematic? Since the situations, and so the consequences of failures, are very broad, an exhaustive and exact safety analysis is impossible. This context-dependence also extends to the anticipated use of the UAS and the role of the operator/pilot. Thus, the analysis, even if we assume only worst-cases, will surely be incomplete. Given this, how do we evaluate the probability of failure? Even if we believe we can estimate this probability, then how can we be sure that this remains stable over time? Maybe faults will only appear after the system has worked successfully for some time?

To overcome such problems, and give a practical route to certification, a variety of documents have been produced to provide guidance when proving compliance of systems design with respect to 1309. There are very many of these, including

- *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*, ARP 4761 [4], describing safety assessment (associated with 1309 compliance) approaches for certification of civil aircraft,

- *Guidelines for Development of Civil Aircraft and Systems*, ARP 4754A, addressing the overall aircraft operating environment and functions and including validation of requirements and verification of the design implementation for certification and product assurance, and

- *Software Considerations in Airborne Systems and Equipment Certification*, RTCA DO-178B/C [5], providing guidance for ensuring that airborne software systems have airworthiness.

Subsequently, DO-178C has extended DO178B with supplements concerning Software Tool Qualification Considerations, Model Based Development and Verification, Object-Oriented Technology and Related Techniques, and (crucially) Formal Methods. Together, these documents provide a route to the certification of traditional air systems though, as we will see, UAS present new challenges.

Designing for Autonomy

In developing an air system with a view to certification, a typical approach is to assess, at quite an early stage of the design, the faults that might occur. This *Preliminary System Safety Assessment* involves an analysis of the system's architecture as well as its functional requirements. Typical failure analysis examines these functions and tests whether (a) the system ever performs the function when it is *not* required to do so, (b) the system does *not* perform the function when it *is* required to do so, or (c) the system performs the function incorrectly.

As we move towards UAS operations, this approach raises two concerns. First, as the human is not formally part of this process, our replacement of the human's capabilities by an autonomous software system is not well represented. Specifically, such an analysis says little about the decisions the software must make that were previously made by the pilot (as these aspects are not certified in current manned air systems).

Secondly, the analysis carried out on the system software must surely be strengthened once we rely so completely on its correctness. Traditionally, software is assumed to be deterministic and to fail in `predictable' ways. Consequently, the verification and validation carried out has involved a range of informal processes and testing, with formal aspects only appearing (and then in a very simplified form) in languages such as SPARK 2014 (based on Ada) [6].

Let us consider these two aspects in a little more detail.

Software Assurance Levels: Traditionally, the strongest level of system assurance (categorised through *Design/Development Assurance Levels*) is “Level A”, which is typically reserved for sub-systems that are fundamentally *safety critical*. However, since designing, implementing and verifying compliance with 1309 at this level is very expensive, AP 4754 proposes means to reduce the assurance burden by using the system architecture to mitigate or control the degree to which a function contributes to a specific failure condition. A common approach involves multiple implementations of a function (providing redundancy), very distinct functional implementations (to avoid c*ommon mode* failures), and physically separating these functional implementations (to avoid one failure compromising all functions).

However, as we move to operational UAS, not only might the designer be unaware of all the possible situations (and so expected behaviours), but we will require much more comprehensive verification, not only to convince certification authorities, but also to enhance public confidence. Current levels of testing are clearly limited; testing samples the possible space of behaviours and makes no guarantee about the behaviour of a system outside of these. In certain safety critical systems, some forms of formal verification have been utilised for the software involved. However, techniques such as SPARK are quite limited in their verification and analysis capabilities. For complex software, especially software that is now expected to make critical decisions in unanticipated situations, we will likely require full formal verification (see below).

Clearly, the UAS is remote from any pilot/operator. Consequently, the human will likely find it difficult not only to recognise or understand any current problems, but also provide a correct solution and do so quickly enough to solve the problem at the UAS. ICAO states that the pilot in command of a remotely piloted aircraft has precisely the same responsibilities as an on-board pilot-in-command i.e. the safe operation of the aircraft [7]. This inevitably leads to autonomy [8] whereby the UAS must make certain decisions, concerning safety and situational awareness, rather than waiting for guidance from the pilot/operator. In addition, the remoteness of the UAS means that, once communication failure occurs, then the UAS software must make the crucial decisions after consideration of its situation.

Verification and Validation

In an attempt to reduce (or even eliminate) software faults, the notion of software `correctness' or reliability is important [9]. There are two main components to this. Firstly, does the software deliver what it was required to do and in accordance with its original specification. Secondly, has the software been correctly produced. These are sometimes identified as: *validation* – "have we built the right thing"; and *verification* – "have we built it right".

While the legal framework for UAS is still under development, it seems clear that stronger verification processes will be necessary. We must be *sure* that the software systems responsible for autonomous decision-making will indeed behave as expected. Although extensive testing can help, it is likely that we will have to turn to more comprehensive techniques for "proving" properties of the new internal software such as *formal verification* used in critical systems. This involves the deep, formal analysis of software and, in particular, providing logical justification that software will always match its formal requirements. These formal verification techniques take a set of formal requirements, presented in a formal logic, and then undertake a comprehensive mathematical analysis of the software in order to `prove' whether or not its behaviour corresponds to its formal requirements. Traditionally, formal verification techniques are used for safety-critical systems, and so invoking them in the case of UAS autonomous decision-making seems not only appropriate, but essential.

The most popular form of formal verification is called "model checking" [10,11]. Here, all possible executions of the software (or system) being checked are explored and each one is assessed against the formal requirement. If any execution fails to match up to its requirement, an error is flagged. While model checking is widely used in the analysis of critical systems, the development of model checking for autonomous software is relatively recent [12,13], and application to the verification of practical autonomous systems is still at a very early stage [14].

Validation involves ensuring that the system/software has the expected behaviour once embedded in its target environment, and is often concerned with satisfying customer/stakeholder wishes. For example, does the system match required legal standards set by regulators, does it have all the functionality expected by a customer, and how well does it work in practical environments?

So, verification and validation of software involves a variety of techniques, from *formal verification* through *testing*, to in-situ *evaluation*. As it is impossible to accurately model the real-world, we clearly cannot exhaustively explore the correctness of software/systems in all possible environments and so must utilise abstractions when carrying out formal verification, together with subsequent testing in order to validate these abstractions.

In our UAS, software must now make the decisions that a human pilot once made. But how can we be sure what decisions the software will make and, importantly, *why* it chose to make them? Current research on formally verifying autonomous systems involves isolating this decision-making entity and verifying the detailed working of this software. So, by using new formal verification techniques specifically developed for such autonomous decision-making [15], we can prove properties of the software making the high-level decisions within our autonomous systems. For example, in UAS, we have shown how such verification can be used to establish that the autonomous system's decision-making matches (at a basic level) the pilot's [16]. (We will consider how this formal verification fits within an approach to UAS certification within the next section.)

Virtual Certification of UAS

One approach to solving the certification conundrum mentioned earlier in the article is to try to certify, or, at least, identify where the certification issues are going to arise using *virtual environments*. This process is called virtual certification (VC). The key point here is that wherever possible, existing tools and techniques would be employed to ease the certification burden. In order to achieve this, a representative model of the airframe and its flight control system (FCS) is required.

This might be a simple point mass characterised by kinematics representative of the envisaged UAS' performance, linear models of the aircraft flight dynamics all of the way up to a fully non-linear flight dynamics model including actuator saturation and rate limits plus FCS processing time delays.

In addition, a representative model of the environment in which the UAS will operate is also required. This might include

- An abstract environment model of the real world for use with the decision-making software through to a fully detailed 3-dimensional database for real-time visualisation of test results;
- An implementation of the decision-making software (Including any formal specification that it is required to meet); and
- Simulation models of the sensors and processing algorithms that are being used to provide the decision-making software with the data/information on which it bases its decisions (again, this could range from simple 'truth' data feeds through to full-physics representations of the sensors including, for example, signal noise and degradation due to the modelled atmospheric conditions).

These items provide the framework to allow different forms of verification and validation to be carried out during different points in the design life cycle of the UAS. The first stage would be to formally verify the decision-making software against the logical specification. Here, formally verifying the actual decision-making software is advantageous. Current manned practice uses highly trained, highly motivated test pilots who may or may not have decision-making skills that are representative of the particular line pilot at the controls of an air vehicle on any given day. The decision-making software could then be coupled to the airframe environment models in a variety of ways to start to at least generate evidence to demonstrate that the UAS can be "Flown Safely".

Simple models of the airframe/FCS allow rapid processing of a wide variety of scenarios, particularly in the presence of High Performance Computing facilities. Here, the models could be placed into a wide variety of situations varying with respect to environmental factors, other air traffic, sensor performance regimes etc., and the performance of the decision-making software assessed against relevant metrics, e.g. adherence to the "Rules of the Air". Monte Carlo techniques could be employed here to provide the required probabilistic estimates most crucially for catastrophic failures. Importantly, this kind of testing could be used to inform where the likely areas of difficulty for the decision-making system would be in terms of adhering to the requirements demanded of it.

The results obtained above would be most likely based upon linear aircraft models and abstract models of the environment. High fidelity, non-linear, full physics models could then be brought into play to refine or validate the “Safe/Not Safe” operational boundaries for the UAS. Real-time simulation could be used to demonstrate key results to stakeholders e.g. regulators.

It is, of course, unlikely that a UAS will ever be certified using evidence generated virtually. However, none of the above would be wasted effort. It provides a useful start point to the test program for any UAS as it provides the test team with the key areas of either uncertainty or risk to be focussed on rather than the benign flight conditions that can be easily handled by the decision-making software.

The ASTRAEA programme is developing a virtual certification process with the aim of de-risking the commercial development of UAS [17]. The start point for this process is an analogue to the baseline regulatory code, CS-23. The key technology themes to be incorporated in the VC process are: Detect and Avoid; Autonomy; Command, Control and Communications and Ground Operations and the Human-System Interface. For each of these technical themes, the key deliverables are planned to be: a certification plan; functional specifications; a preliminary hazard analysis; a compliance checklist and the documenting of any special conditions that apply over and above manned aircraft processes. The special conditions will be developed for: emergency recovery capabilities; command and control links; levels of autonomy; ground control station and human factors; operational requirements and the system safety assessment.

Operational Issues

Given that a UAS.1301/1309 will eventually exist, and assuming that much of it will be based on the current regulations, there seems to be little doubt that a future UAS can be produced which is “Safe to Fly”. However, there still remains the “Flown Safely” requirement for full and acceptable safety levels in operation. There are a number of documents that cover these aspects and the more important are briefly mentioned:

- The UK CAA CAP 393 - Air Navigation: The Order and the Regulations. This details the Rules of the Air for the UK [18] and cover general operations and specifically the Instrument Flight Rules (IFR) and Visual Flight Rules (VFR) all of which UAS operations must comply;

- The UK CAA CAP 722 - Unmanned Aircraft System Operations in UK Airspace – Guidance [19].

Concluding Remarks

The integration of UAS into non-segregated airspace remains problematic despite many research and development efforts being undertaken. While deeper understanding of what needs to be achieved in order to certify the aircraft and its systems, including the Ground Control Stations, is progressing well, there is still much to be done to prove the operational aspects can provide the requisite levels of safety. It seems clear that the additional burden of responsibility placed on the autonomous software will necessitate a much deeper level of analysis, such as that provided by formal verification methods. How these methods can then be incorporated into the compliance process then becomes a problem. However, it appears that the combination of formal verification, virtual prototypes, and substantive testing can provide a route towards the certification of unmanned air systems.

References

1. Circular 393, “Unmanned Aircraft Systems”, ICAO, 2011 available at http://www.icao.int/Meetings/UAS/Documents/Circular%20328_en.pdf.
2. The FAA includes the category “No Safety Effect” in their broadly equivalent FARs
3. JAR23/25.1309 Equipment Systems and Installations (Interpretative Material and Acceptable Means of Compliance). Further information about compliance is given in AC 25.1309-1A - System Design and Analysis published by the FAA and available at http://www.faa.gov/regulations_policies/advisory_circulars/index.cfm/go/document.information/documentID/22680
4. Aerospace Recommended Practice (ARP) 4761 is available at SAE International at http://standards.sae.org/arp4761/
5. Available from the RTCA on-line store http://www.rtca.org/
6. An overview of SPARK 2014 and its associated toolset can be found at http:/docs.adacore.com/spark2014-docs/html/ug/
7. Convention on International Civil Aviation - Doc 7300 otherwise known as the Chicago Convention available at http://www.icao.int/publications/pages/doc7300.aspx and ICAO Chicago Convention, Article 8, “Pilotless Aircraft”
8. There are several definitions of autonomy and most are open to (heated) academic debate and are generally unhelpful to the layman. A simple and practical one is: “An autonomous system is one that independently makes decisions from choice”.
9. The terms software “reliability” or software “correctness” are taken in this paper to mean an absence of faults.
10. “Model Checking”. Edmund Clarke, Orna Grumberg, and Doron Peled. *MIT Press*, 2000.
11. "Principles of Model Checking", Christel Baier and Joost-Pieter Katoen. *MIT Press*, 2008.
12. "Model Checking Rational Agents". Rafael H. Bordini, Michael Fisher, Willem Visser, and Michael Wooldridge. *IEEE Intelligent Systems 19(5):46-52*, 2004
13. "Automatic Verification of Multi-agent Systems by Model Checking via Ordered Binary Decision Diagrams". Franco Raimondi and Alessio Lomuscio. *Journal of Applied Logic 5(2):235-251*, 2007.
14. "Verifying Autonomous Systems". Michael Fisher, Louise Dennis, and Matt Webster. *Communications of the ACM 56(9):84-93*, 2013. http://doi.acm.org/10.1145/2494558
15. "Generating Certification Evidence for Autonomous Unmanned Aircraft Using Model Checking and Simulation". Matt Webster, Neil Cameron, Mike Jump, and Michael Fisher. *Journal of Aerospace Information Systems 11(5):258-279*, 2014.
16. "If you want to trust a robot, look at how it makes decisions". Michael Fisher. *The Conversation*, March 2014. http://theconversation.com/if-you-want-to-trust-a-robot-look-at-how-it-makes-decisions-24134
17. Available at http://www.astraea.aero/downloads/RAeS%20Conference%202011/ASTRAEA_VC_overview_RAES_2011_V1.ppt.pdf
18. The overarching document for the Rules of the Air is *ICAO Chicago Convention, Annex 2* and the FAA equivalent is in the Code of Federal Regulations (CFR), Title 14, Part 91 “Aeronautics and Space”.
19. Available at https://www.caa.co.uk/cap722