CRANFIELD UNIVERSITY

CHARLES H PATCHETT

ON THE DERIVATION AND ANALYSIS OF DECISION ARCHITECTURES FOR UNINHABITED AIR SYSTEMS

DEFENCE ACADEMY - COLLEGE OF MANAGEMENT AND TECHNOLOGY

APPLIED MATHEMATICS AND SCIENTIFIC COMPUTING GROUP

DEPARTMENT OF INFORMATICS AND SYSTEMS ENGINEERING

PhD THESIS

Academic Year: 2005 - 2011

Supervisor: V.V.S.S.Sastry

July 2011

ABSTRACT

Operation of Unmanned Air Vehicles (UAVs) has increased significantly over the past few years. However, routine operation in non-segregated airspace remains a challenge, primarily due to nature of the environment and restrictions and challenges that accompany this. Currently, tight human control is envisaged as a means to achieve the oft quoted requirements of transparency[1], equivalence[2] and safety. However, the problems of high cost of human operation, potential communication losses and operator remoteness remain as obstacles. One means of overcoming these obstacles is to devolve authority, from the ground controller to an on-board system able to understand its situation and make appropriate decisions when authorised. Such an on-board system is known as an **Autonomous System**.

The nature of the autonomous system, how it should be designed, when and how authority should be transferred and in what context can they be allowed to control the vehicle are the general motivation for this study. To do this, the system must overcome the negative aspects of differentiators that exist between UASs and manned aircraft and introduce methods to achieve required increases in the levels of versatility, cost, safety and performance.

The general thesis of this work is that the role and responsibility of an airborne autonomous system are sufficiently different from those of other conventionally controlled manned and unmanned systems to require a different architectural approach. Such a different architecture will also have additional requirements placed upon it in order to demonstrate acceptable levels of Transparency, Equivalence and Safety.

The architecture for the system is developed from an analysis of the basic requirements and adapted from a consideration of other, suitable candidates for

[1] Transparency means vehicle operation such that an external observer, such as Air Traffic Control or other aircraft, would not be able to determine whether the vehicle was manned or otherwise.

[2] Equivalence refers to the ability of the vehicle operation to adhere equivalently as a manned aircraft to the rules and regulations relevant to the airspace it is operating in.

effective control of the vehicle under devolved authority. The best practices for airborne systems in general are identified and amalgamated with established principles and approaches of robotics and intelligent agents. From this, a decision architecture, capable of interacting with external human agencies such as the UAS Commander and Air Traffic Controllers, is proposed in detail. This architecture has been implemented and a number of further lessons can be drawn from this.

In order to understand in detail the system safety requirements, an analysis of manned and unmanned aircraft accidents is made. Particular interest is given to the type of control moding of current unmanned aircraft in order to make a comparison, and prediction, with accidents likely to be caused by autonomously controlled vehicles. The effect of pilot remoteness on the accident rate is studied and a new classification of this remoteness is identified as a major contributor to accidents A preliminary Bayesian model for unmanned aircraft accidents is developed and results and predictions are made as an output of this model.

From the accident analysis and modelling, strategies to improve UAS safety are identified. Detailed implementations within these strategies are analysed and a proposal for more advanced Human-Machine Interaction made. In particular, detailed analysis is given on exemplar scenarios that a UAS may encounter. These are: Sense and Avoid[3], Mission Management Failure, Take Off/Landing, and Lost Link procedures and Communications Failure. These analyses identify the nature of autonomous, as opposed to automatic, operation and clearly show the benefits to safety of autonomous air vehicle operation, with an identifiable decision architecture, and its relationship with the human controller.

From the strategies and detailed analysis of the exemplar scenarios, proposals are made for the improvement of unmanned vehicle safety The incorporation of these proposals into the suggested decision architecture are accompanied by

[3] A Sense and Avoid system is the equivalent of the manned aircraft requirement for a pilot to lookout for other aircraft to avoid mid-air collisions (the so-called “See and Avoid” requirement).

analysis of the levels of benefit that may be expected. These suggest that a level approaching that of conventional manned aircraft is achievable using currently available technologies but with substantial architectural design methodologies than currently fielded.

Keywords: UAS, Safety, Human Machine Interaction, Sense and Avoid.

PROLOGUE

"Writing a Thesis is an adventure. To begin with, it is a toy and an amusement. Then it becomes a mistress, then a master and eventually a tyrant. Finally, just as you are reconciled to your servitude, you kill the monster and submit".

Paraphrased from Winston Churchill

Winston was right - writing a thesis as a contribution to original knowledge is an awesome task. One has to take a subject, forensically research it, investigate flaws and develop new areas, propose further avenues of work, and finally, defend it against all argument as a statement of certainty of knowledge.

However, in doing so, one becomes increasingly aware, and uncomfortable, that there can be no such certain statement. A Thesis is definitely not an statement of certainty – it is only an argument; perhaps a well thought out and researched one, but one that is always susceptible to the law of uncertainty – Heisenberg perhaps got it right after all; and the never proven, but incredibly successful theory of Quantum Mechanics, repeatedly provides evidence that there is nothing in nature that is certain, and more so, is unlikely (excuse the pun) to be proven to be so.

Neither is the so called "Test of Time" a good test – there were 350 years between Newton and Einstein – what were all the physicists doing in between; perhaps they were satisfied they were certain in their knowledge. If they were truly masters of their craft, they would have known, in their heart of hearts, that this was untrue.

If anything, all a thesis can ever say is that, at this moment in time, I believe this to be true and that is uncertainly right, and therefore likely to be, perhaps in the next moment of time, certainly wrong!

ACKNOWLEDGEMENTS

Researching a subject in order to qualify for the award of a PhD is probably the most subtle and introspective self-examination that can be ever experienced – and I think I have now, at my age, experienced a lot. Even from the 18 year old boy, who went to University at a prestigious college in London, with grand academic ideas, at great expense funded by his parents, and demonstrated such a complete lack of maturity, that he failed in virtually every academic and social aspect. So, therefore, the first acknowledgment must be to my parents – they would have known and trusted me better than I did at the time.

Without BAE Systems support, my academic journey would have been impossible and I am truly grateful to those managers who saw the value in the research I was proposing and who ardently supported it, particularly Mike Everett, Steve M^{c}Kinsey and Maureen M^{c}Cue.

To my supervisor, Venkat Sastry, who has been an inspiring navigator, always guiding me down the route to follow, never using a brute force of direction but constantly providing that backbone of knowledge and question, and without whose support this work would never have come about.

To the ASTRAEA[4] programme, which funded the vast majority of this work and to which I hope the output will be ultimately of value to its objectives.

Finally, and most importantly, to my family, particularly my wife, who have always given me the support and strength to complete the research that they instinctively, and unquestioningly, knew I needed to do. I love them deeply and dearly.

[4] ASTRAEA is charged with providing the technology and maturity of test applications which will eventually enable the routine use of UASs in UK un-segregated airspace.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ADA	A Programming Language
ACAS	Airborne Collision Avoidance System
ACE	Advanced Collaborative Environment
AESA	Active Electronically Scanned Radar
AFRL	Air Force Research Laboratories (US)
AIMS	Autonomous Integrated Mission System
AOPA	Aircraft Owners and Pilots Association (US)
AOS	Agent Oriented Software
APL	Applicable Plan List
APU	Auxiliary Power unit
AS	Autonomous System
ASAAC	Allied Standards Avionic Architecture Council
ASARP	ACAS Safety Analysis post-RVSM implementation Project (Eurocontrol)
ASIP	Airborne Signals Intelligence Payload
ASTRAEA	Autonomous Systems Technology Related Airborne Evaluation & Assessment
ATC	Air Traffic Control
ATM	Air Traffic Management
ATS	Air Traffic Services
BACN	Battlefield Airborne Communications Node
BBUP	Black Box with Unknown Processes
BG	Behaviour Generation
BDI	Belief Desire and Intention
BLOS	Beyond Line Of Sight

C^2	Command and Control
CAA	Civil Aviation Authority (CAA)
CAP	Capability
CAS	Close Air Support
CBR	Case Based Reasoning
CDT	Classical Decision Theory
CFiT	Controlled Flight into Terrain
CLARAty	Coupled Layer Architecture for Robotic Autonomy
COA	Course of Action
COCR	Communications Operating Concept and Requirements
COS	Common Operating System
CPA	Closest Point of Approach
CPDLC	Cockpit Pilot Data Link Control
CSAR	Combat Search and Rescue
DARPA	Defence Applied Research and Procurement Agency (US)
DCGS	Distributed Common Ground System
DM	Decision Maker
DoD	Department of Defense (US)
EASA	European Safety Agency
ECAM	Electronic Centralised Aircraft Monitor
EISS	Enhanced Integrated Sensor Suite
EM	Emergency Manager
EO	Electro-Optic
FAA	Federal Aviation Authority (US)
FAR	Federal Aviation Regulations
FCS	Flight Control System

FFA	Functional Failure Analysis
FL	Flight Level
FMS	Flight Management System
FRS	Future Radio System
GA	General Aviation
GCS	Ground Control System
GMTI	Ground Moving Target Indicator
GPS	Global Positioning System
GPWS	Ground Proximity Warning System
HFACS	Human Factors Analysis and Classification System
HMI	Human Machine Interface
HRR	High-Range-Resolution (imagery)
Hz	Hertz
ICAO	International Civil Aviation Authority
IBS	Integrated Broadcast System
IDA	Information – Decision – Action
IDC	Information – Decision – Control
IEEE	Institute of Electrical and Electronics Engineers (US)
IET	Institute of Engineering and Technology
IMA	Integrated Modular Avionics
IR	Infra Red
ISR	Intelligence, Surveillance and Reconnaissance
JAA	Joint Aviation Authorities (EU)
JAR	Joint Aviation Requirements
JPL	Jet Propulsion Laboratory (US)
J-UCAS	Joint Unmanned Air Combat System

KD	Knowledge Database
KIAS	Knots Indicated Air Speed
km	Kilometre
KTAS	Knots True Air Speed
LEO	Low Earth Orbit
LGA	La Guardia Airport (New York)
LOC	Loss of Control
LRE	Launch and Recovery Element
LRU	Line Replaceable Unit
MAS	Multi Agent System
MCE	Mission Control Element
ME	Master Executive
MFOP	Maintenance-Free Operating Period
MIP	Military Intelligence Program (US)
MoD	Ministry of Defence (UK)
MP-RTIP	Multi-Platform Radar Technology Insertion Program
MTBF	Mean Time Between Failures
NASA	National Aeronautics and Space Administration (US)
NATO	North Atlantic Treaty Organisation
NBS	Navigation and Bombing System
NIST	National Institute of Standards and Technology (US)
NTSB	National Transport Safety Board (US)
OODA	Observation-Orientation–Decision-Action
PACT	Pilot Authorisation and Control of Tasks
PC	Personal Computer
PDA `	Personal Digital Assistant

PF	Pilot Flying
PHA	Preliminary Hazard Analysis
PL	PACT Level
PM	Pilot Monitoring
PMF	Pilot Monitoring Failure
PWS	Pilot Warning System
QC	Quality Controller
QRF	Quick Reaction Handbook
QUASA	Quantative Analysis of Situation Awareness
RA	Resolution Advisory (TCAS)
RAF	Royal Air Force
RAP	Reactive Action Package
RCS	Real-time Control System
RHS	Right Hand Side
RPDM	Recognition Primed Decision Making
RPV	Remotely Piloted Vehicle
RSO	Remote Split Operation
RTB	Return To Base
SA	Situation Awareness
SADL	Situation Airborne Data Link
SAR	Search and Rescue or Synthetic Aperture Radar
SatComms	Satellite Communications
SDE	Shared Data Environment
SE	Synthetic Environment
SEAD	Suppression of Enemy Air Defences
SEBA	Synthetic Environment Based Acquisition

SEMS	Synthetic Environment Models and Simulation
SIGINT	Signals Intelligence
SIL	Software Integrity Level
SP	Sensory Processing
SPA	Sense-Plan-Act
SRG	Safety Regulatory Group (of the CAA)
TCA	Track Crossing Angle
TCAS	Traffic Collision Avoidance System
TEB	Teterboro Airport (New York)
TES	Tactical Exploitation system
TLA	Three Layer Architecture
TV	Television
UAS	Uninhabited Air System
UAAS	Uninhabited Autonomous Air System
UAV	Uninhabited Air Vehicle
UCAV	Uninhabited Combat Air Vehicle
USA	United States of America
USAF	United States Air Force
VHF	Very High Frequency (Radio)
VJ	Value Judgement
VME	VersaModule Eurocard
WM	World Modelling
XUV	Experimental Unmanned (Ground) Vehicle

1 Introduction

1.1 General Motivation

Airborne military and civil vehicles, and the means to control them, are changing rapidly. Although Uninhabited Air Vehicles (UAVs) have operated in the military sector for many years, there has been a recent and significant acceleration of programmes to develop, manufacture and operate them not only in combat arenas but also in the civil sector. Whilst the manufacture of small UAVs is cheap and development is relatively quick, the routine operation of these vehicles in civil airspace, currently containing only manned aircraft, is neither. This is largely due to the nature of the operating environment and the restrictions and challenges it brings. Restrictions, due to fact that the environment is highly regulated and therefore controlled, and challenges, due to the fact that the environment is uncertain and hazardous. The means to overcome these restrictions and challenges can be satisfied, to a degree, by having tight human control over the vehicle. However, if we require these vehicles to be operated with minimal human involvement, due to the high cost of human labour, then new ways of overcoming the restrictions and meeting the challenges must be sought.

Thus we see a current picture of thousands (literally) of different UAVs being built but only a few being able to operate in either diverse roles, conditions or airspace categories, and none whatsoever with respect to all three aspects.

So, the challenge today is to reduce operating costs by gradually replacing the human involvement in the control process whilst retaining satisfactory operating performance and regulatory adherence. This can be achieved by transferring authority for some of the control functions normally made by the pilot or ground controller, to an on-board system able to make decisions and implement them. Such a system is conventionally known as an **autonomous system**. When such a system is incorporated into a UAV, the consequent vehicle and Ground

Control System is described collectively as an Uninhabited Autonomous Air System (UAAS)[5].

Which functions should be transferred, how should they be implemented and in what context can they be allowed to control the vehicle are the general motivation for this study, as these aspects are still largely unproven and, where they have been identified, are often at the early stages of development and design.

1.1.1 The Operating Environment

As discussed above, the operating characteristics, and therefore the requirements, for an airborne system are largely determined by the nature of the environment in which it operates. The nature of this environment, and the characteristics it imposes, are briefly touched upon here and described more fully in the next Chapter.

The operational environment for a UAS:

- Has collision hazards, either with other airborne, and likely to be manned objects, or with the ground. These must be avoided with an extremely high degree of success as the consequences of collision are potentially life threatening. Typical probabilities of success are likely to be $< 10^{-9}$ per flying hour[6].
- Contains uncertain, incomplete and inaccurate data – the environmental dimension for a typical aircraft is vast. Even a light aircraft can routinely operate within an area of 50,000 sq. nm. Large aircraft are global in their sphere of operation. The aircraft density in the air environment is vastly uneven. Airborne entities are present in dense clusters near airports yet

[5] There are several descriptions and definitions of UAV, UAS and UAAS available from a variety of sources, some authoritive, some not. Nearly all are at variance with each other. Throughout the rest of this Thesis, the term Uninhabited Air System (UAS), taken to mean an Uninhabited Air Vehicle together with its associated Ground Control System will be used, irrespective of whether or not it is autonomous, automatic or remotely piloted

[6] For instance, a mid-air collision could be classified as “Catastrophic” – the highest level of hazard. Protection against such events for medium/large civilian aircraft is specified such that the probability of the event occurring is less than $< 10^{-9}$ per flying hour.

widely separated over oceanic areas. Objects, and their associated data, within the operating area may or may not be sensed. Even if they are, it may not be to a level of accuracy that is required.

- Is highly regulated – aircraft must conform to the rules of the air and operate to tight procedural control under certain circumstances. Operating under this control, i.e. Air Traffic Control (ATC), is mandatory when flying into and out of an airfield or airport. It is also mandatory for certain categories of airspace. This requires that the platform must communicate, understand and conform to ATC commands in a highly robust and reliable way.

  In order for UASs to operate in regulated airspace, there is a requirement for adherence to two main principles: Transparency and Equivalence [1]. Transparency means that UASs should not require special procedures or equipment fits and must operate in the same way as other aircraft. In short, it should be transparent to other air users or controllers that the aircraft is unmanned. Equivalence requires that the UAS operates to the same regulation set as other air users and performs to a standard equivalent to that of a manned aircraft. Thus some of the functions of a manned aircraft performed by the pilot, such as look out, must be replicated in some form, and to the same level of performance, as a human.

- Dynamic - and therefore capable of leading to a variety of unforeseen events such as :
  - Emergencies of a diverse nature such as fires, engine failures, operation outside a planned envelope etc.
  - In flight changes of missions, plans, targets, roles and responsibilities.
  - Pop up threats to avoid, such as threatening hostile objects (military) or storm and turbulent air centres.
  - Pop up objects to co-operate or co-ordinate with.
  - Changes in environmental conditions such as precipitation,

turbulence, icing, wind and temperature. These changes can sometimes occur quite rapidly and often have consequences on the operation of the vehicle.

- Has gravity as an opponent! The vast majority of air vehicles are heavier-than-air and rely on systems (engines for powered aircraft; vertical wind for gliders) for generating lift. These systems generally have fixed resources and before they run out, the aircraft must land. After they have run out, the aircraft will certainly crash. The former requires an airfield, which in turn must be precisely navigated to, and the latter usually implies some form of damage, which in the worst case may be fatal, either to the occupants or to those on the ground.

In short, the environment is inherently unsafe. This in turn requires that the systems which operate within it have certain characteristics, some of which are commented on here:

- The vehicles need to operate in real time and continuously. To do this, most of their systems need to do the same. This requires that:
  - They are robust and reliable – they cannot stop or pause. If they are computers, they must be guaranteed not to crash and to complete their computing within a defined timescale. This usually also entails some form of high integrity software.
  - Must handle Mission Critical functions. This usually entails some form of redundancy.
  - Must handle Safety Critical functions, failure of which must not either occur or must be handled in a fail-safe way. This usually entails some forms of redundant safety modes and even higher levels of software integrity.

Later, it will be seen that avionic systems, and the standards to which they have been built, have evolved to encompass these requirements and face up to the challenges of the operating environment.

1.1.2 Control of Unmanned Air System

UASs can be flown in a variety of control modes all of which have in common the fact that the pilots, commander or operator(s) are remote from the vehicle, either within, or beyond, the line of sight.

The most common mode of control, particularly for light UASs is remote control operation whereby the operator flies the aircraft manually at all times like a model aircraft. Such operation requires a high level of skill and many accidents of vehicles using this type of operation are skill based errors, the majority being prevalent in the take-off and landing phases of flight. Of course, if these vehicles are cheap, easy to repair, or even disposable, then such accidents, provided they do not endanger life, are acceptable.

However, where such accidents are not acceptable, then in order to reduce the probability of their occurrence, some UASs have automated flight control (i.e. an autopilot) and only use manual control for take-off and landing. The USA's *Predator* UAS is a good example.

Again, in an attempt to reduce skill based errors causing accidents, automated control can be used in all phases of flight. This automation can also be extended to include areas other than flight control, such as path and mission planning. The USA's *Global Hawk* UAS is an example.

Finally, it is possible to confer a degree of autonomy to the UAS and allow it to make its own decisions according to the situation it is in and under specific conditions of authorisation. Such systems are considered Uninhabited Autonomous Air Systems (UASs). They potentially offer many advantages compared to conventionally controlled UASs but also bring fresh challenges. The nature of those challenges, the means to meet them, and the consequences are the main thrust of this thesis. Currently there are no known examples of autonomous UASs, as opposed to automated (no matter how highly), in the field.

1.1.3 Control of Autonomous Systems

Systems for autonomous control have been progressing for the last twenty five years and that progress has been strongly associated with that of increasing computing power. Many theoretical arguments have been advanced and consequent system models have been built to do a variety of tasks. Recently these have branched into many areas: web crawlers, manufacturing and vehicle control are some examples. The most interesting, and probably diverse, are those for the control of robots; generally small and ground based. These have evolved over the last few years and their general design apparently seems to have stabilised with the general consensus of the robotic community. This general design, or architecture, is frequently referred to as the **Three Layer Architecture (TLA)**, so called because it is made up of three primary functions: **planning, sequencing,** and **control**. This general architecture is investigated in some depth at Para. 3.2.5.

1.1.4 Autonomous Control of Air Systems

So, it seems that there is consensus within the robotic community of the suitability of the TLA for control, and there is consensus within the avionic community for the general design principles of avionic systems (which can be used for control). However, there appears to be no confluence at present between avionic systems and robotic architectures for the control of autonomous air systems. This could be for several reasons:

- Money – avionic systems are very expensive (compared to robotic systems). Research using complete systems would normally be outside the financial scope of a university student or team. In addition, the funds for a UAS are usually used up on the airframe, engine and vehicle control system with little for the mission system. This is due to the need to get the vehicle flying as early as possible.
- Availability - high, or even medium, fidelity models of avionic systems are usually unavailable from system designers who are operating in a very competitive market. In addition, suitably detailed, high fidelity, models of the operating environment, known as a Synthetic Environment

(SE), are equally unavailable.

- Need – in order to just get airborne, fly around and land in fully restricted airspace, perhaps doing something interesting in between, does not require a heavy weight avionic system conforming to international standards. Corners can, and should for many cases, easily be cut and in many ways, particularly for research purposes. However, in order to develop a system that will conform to these standards is a serious and expensive undertaking.
- Performance - robotics research work is often limited in scope and achievement to focus upon a specialised topic of interest. Whilst valid, it has to be recognised that this approach would not test the limitations in a complex UAS control architecture where many compromises have to be made in order to do difficult and/or complex tasks.
- Automatic systems – many systems, particularly avionic ones, whilst claiming to be autonomous, are frequently found to be, on inspection, merely automated[7]. Why, because autonomous control of an air vehicle is a difficult, and as discussed, a potentially dangerous thing either to do, or be allowed to do.

1.1.5 Human Control Aspects

In identifying that the control mechanisms of a UAS are distributed between the remote ground operator, who has overall command responsibility, and the on-board autonomous system, it is likely that the architecture of future UASs will be driven by the need to reduce the involvement of humans, particularly in terms of direct control. The drivers for this can be summarised as follows:

- **Bandwidth** As the number of UASs grow, so the requirement for higher levels of bandwidth increases. In recent years the number of UASs in the "field" has grown exponentially. Unfortunately the available spectrum is characterised by the laws of physics and other users and is, in practical terms, capped. One way of reducing the bandwidth requirement

[7] The difference between autonomous and automated operating modes is discussed in detail at Appendix A.

is to increase on-board autonomy.

- **Operating Cost** There is a need to drive down operating costs. Currently, the major part of this are the number of humans required to operate the vehicle[8]. If some of these tasks can be undertaken by the on-board systems, then operating costs can be reduced.
- **Operator Remoteness** The fact the operator is remote from the vehicle can lead to loss of control and/or situational awareness and consequently, in extreme cases, loss of the UAS. In some cases, it may be possible that the UAS is better informed of its state than its operator. In such cases, if they can be identified, provision by way of design features can avoid such losses. This is a major point within this thesis and is discussed at length later.

Clearly, the inherent nature of distributed and remote human control of the UAS brings its own problems and, when compared to the control aspects of a conventional aircraft, the opportunity for failure within a UAS leading to an accident is clearly likely to be different.

Whilst it is true that humans cause accidents, they are also well placed to prevent accidents. Their special skills in reasoning and extensive experience, coupled with their ability to successfully react to unforeseen and complex situations, enable them prevent the propagation of errors could ultimately result in an accident. So, accident prevention in a UAS is likely to be different to a conventional aircraft.

The industry standard for the analysis of human error in aircraft accidents is the Human Factor Analysis and Classification System (HFACS) of Wiegmann and Shappell [*2*]. This in turn is based on a more abstract model of human error proposed by James Reason [3] and which is frequently referred to as the “Swiss Cheese” model. This is so named due to its likening of circumstances, failure and error being propagated through “holes” in barriers that would be normally expected to prevent accidents. This analysis taxonomy, based on the

[8] Global Hawk for example requires about 20 people.

classification of errors, can be applied to UAS accident modelling to discover accident causal factors and identify architectural safeguarding mechanisms that can be applied to reduce the accident rate of UASs to an acceptable level.

1.1.6 UAS Safety

The requirement for safety has already been mentioned. There are, however, several drivers associated with this. The primary ones are certification and, as usual, cost.

For UASs, certainly in the UK (of above 25Kgs weight), before they can be flown on a routine basis, they must be demonstrated to be safe to operate. In other words, there is a requirement to prove that there is a minimum of risk to human life. Unfortunately, UAS have a poor safety record with an accident rate at least an order of magnitude greater than that of General Aviation[9]. The route to certification is discussed in detail later but suffice it to say there is, as yet, not one UAS, fully certificated for flying in routine airspace, anywhere in the world.

This poor safety record is largely attributable to early design decisions to keep costs down. However, with the increased capability of UASs, this view is shifting as evidenced by the following remarks: General Jumper, the US Air Force Chief of Staff in 2005 [4]: *"We've... got to have some respect for the fact that (just) because these are UASs, they are neither expendable or disposable. They cost a lot of money"*. Similarly, General Hal Hornberg, Head of USAF Air Combat Command in 2008 [5]: ".. we can't treat these things like disposable diapers and just throw them out. These things cost money, and it comes out of your Treasury, just like it comes out of ours".

On a more tangible basis, the largest and most sophisticated UAS in the market is the Northrop Grumman Global Hawk which has a published fly away cost of about $80m (and increasing).

From the above remarks (and their tone), together with knowledge of costs comparable with manned aircraft, we can take it that there is a pressing need to

[9] This is formally defined in Paragraph 4.2.4.1 – for now we mean "light civil aircraft".

reduce the accident rate of medium to large UASs on a cost basis alone. (See also Table 4: Examples of Manned/Unmanned Aircraft Reliability

1.1.7 UASs' Control Architectures

Current UASs have simple control architectures based heavily on manned aircraft and there has been little in the way of design progression to accommodate the lack of the on-board human or the fact that he is remote. The only condescension is the addition of a simple "lost link" facility. Study of current control mechanisms, which will be called architectures, invariably leads to the impression that there is a pretence that he (the pilot) is, "in effect on board" and that there is little that therefore needs to be done to accommodate the fact that this is patently untrue. Every UAS Ground Control Station, to date, looks like a cockpit[10] – the flight crew (If we can call them that) even wear flying clothes (including "wings"), thus increasing the illusion of "in effect on board".

If we now extend the functionality of the on-board system to act autonomously under certain circumstances, this pretence becomes increasingly untenable. That the architecture must be (re)-designed to reflect the true reality must be acknowledged and this is the fundamental motivator for the study.

1.2 Statement of Thesis

The general thesis of this work is that the role, responsibilities and environment of an airborne autonomous system, specifically an Uninhabited Autonomous Air System (UAS), are sufficiently different from those of other conventionally controlled manned and unmanned systems to require a different architectural approach. Such a different architecture will also have additional requirements placed upon it in order to demonstrate:

- Acceptable safety levels, preferably at least comparable to conventionally manned aircraft.

[10] All of them have, for instance, flight instruments (altimeter, artificial horizon etc.) despite the fact that, when in Beyond Line of Sight operation using satellite communications, these instruments are lagging, sometimes considerably, due communications latency.

- Acceptable levels of performance despite reduced human presence on board.
- Acceptable levels of integrity and robustness to unforeseen events, again despite reduced human intervention.

1.3 Specific Aims of the Study

The aim of the study was to address the following questions arising from the general thesis:

- What do the issues of role, responsibility and environment force the architecture to achieve that is not found in other architectures?
- Can existing autonomous system architectures address these issues? If so, how, to what degree and why? If not, why not?
- In order to satisfy regulatory authorities and to operate safely and effectively, the UAS "decision agent" will be required to perform competently in routine airspace, almost certainly to the level approaching that of a human pilot. Knowing that the operator is remote, what effect does this have on the underlying accident rate compared to manned aircraft? Are the accident mechanisms the same or different?
- Given that operator involvement should be reduced to the lowest level possible, how does the need for operator interaction affect the design of the architecture, particularly with respect to the need to safeguard against novel accident mechanisms?

1.4 Approach

The approach taken to address these questions was to:

- Understand the nature of autonomous air systems and their underlying architectures for decision making in an airborne environment by:
    - Undertaking a review of existing architectures, irrespective of the environment for which they were intended, and determine their key characteristics.

- Determining the fundamental requirements specific to airborne architectures.
- Assessing the suitability of the existing architectures against these requirements.

- Understand the relationship between the distributed control mechanisms of the human operator and the on-board autonomous air system
- Propose a robust architecture suitable for increased devolved control whilst maintaining high levels of performance, integrity and safety.
- Investigate models and levels of human failure
- Extend the above models of accident causation with particular respect to additional and alternate modes of UASs and their operation when compared to conventional manned and unmanned aircraft.
- Use the accident model and associated data to improve the proposed decision architecture design.
- Assess the proposed architecture against conventional approaches and determine the likely impact of these improvements on UASs future safety records.

1.5 Assumptions

It is assumed that:

- The UAS is sufficiently large, nominally greater than 150 kg and generally of the order of 2000kg, to require adherence to the UK Civil Aviation Authority (CAA) recommendations covering the routine operation of UASs in UK airspace[11] as well as those regulations specified in the CAA Air Navigation Order. In addition, the class of the UAS is assumed to be Class 5 (i.e. designed to operate in all classes of UK airspace).

[11] These recommendations, which are likely to transition to become actual regulations, are defined in the CAA publication, CAP 722.

- The UAS should have a variable level of human operator interaction and therefore be capable of operating with a high degree of autonomy consistent with achieving appropriate levels of performance, safety and regulatory adherence.

- The sub-systems of a UAS, including the decision making system, will have the general characteristics of current avionic systems.

- The decision making system is contained within the avionic system.

1.6 Structure of the Thesis

This Thesis is structured as follows:

In Chapter 2, a statement of the problem to be studied is elaborated which encompasses a review of UAS and manned aircraft operations. This also identifies the key factors that motivate the trend from manual, through automatic and finally to autonomous operation of the air vehicle.

In Chapter 3, the requirements for decision architecture for a UAS are developed and a detailed proposal advanced.

In Chapter 4 the issue of safety is addressed and a preliminary probabilistic accident model advanced. This highlights accident modes specific to UASs and indicates a route to improved safety.

In Chapter 5, strategies for improved safety are identified and target levels are specified.

In Chapter 6, exemplar scenarios are proposed and the operation of the decision architecture in those scenarios is analysed in detail. In particular, a comparison with automatic modes of operation is discussed.

In Chapter 7, a discussion of the overall results is given, including an estimate of safety levels of the architecture likely to be achieved.

The final Chapter 8 draws appropriate conclusions and suggests avenues for further research.

1.7 A Summary of the Major Original Contributions of the Work

1.7.1 The Proposed Decision Architecture

Whilst development of the Three Layer Architecture (TLA) for robots and the need for effective world modelling was completed before this work began, the identification of a complex hierarchical Information Layer to complement that architecture was not and I believe that this work is the first to propose, implement, demonstrate and analyse, a complex, scalar and innovative decision architecture for an autonomous UAS. This is particularly so in the area of safety. Also, very few (if any) of the wide variety of robots using the TLA have been developed to incorporate complementary involvement of humans. The fusion of the TLA with avionic practises, the application of the Pilot Authorisation and Control of Tasks (PACT) levels to enable a complete range of human involvement from zero to full control, the demonstration that the architecture is fully capable of generating behaviours ranging from deliberative through to reactive as an inherent feature with no special provision or switching of modes, I believe to be not only original but also novel and exciting.

1.7.2 A Bayesian Approach to Accident Modelling

There has been at least one analysis of manned aircraft accidents using some aspects of Bayesian models [6] and there have been analyses of unmanned aircraft accidents using the Human Factors Analysis and Classification System (HFACS), but not, I believe, one to date that has combined both of these avenues and looked at unmanned aircraft accidents from the perspective of their control moding whether using a Bayesian approach or not and certainly not of autonomously controlled vehicles.

1.7.3 Strategies for the Improved Safety of Autonomous Air Vehicles

In outlining strategies for improving the safety of autonomous air vehicles, three aspects are presented all of which are novel. The checking of operator beliefs

by a process of abduction is original though based on work by McGuinness and Dawson when they developed Quantative Analysis of Situation Awareness (QUASA) (they stopped at generating an SA metric). The concept of plausibility checking has been around but again I know of no suggestion or implementation that it is comprehensively used for the checking of critical beliefs (to date no aircraft have beliefs, nor are any (sufficiently) self-aware as a route to safety. Since there has been little published research into the Human Factors of autonomous systems, the requirements, one could say design rules, presented here, together with a concept display demonstrating them, is entirely original. If nothing else, it identifies the need for a considerable investment in such research.

1.7.4 Findings from the Analysis of the Exemplars

Analysis of the Exemplars, particularly those regarding the Sense and Avoid and Flight Management scenarios, reveals new knowledge regarding the far more useful, and I believe safe, role an autonomous system with a decision architecture such as that proposed, can contribute especially in comparison to automated architectures. The analysis clearly shows that such an architecture keeps a human supervisor in the loop, and therefore informed, far longer and more appropriately that an automatic system. This together with the inherent deliberative/reactive range of behaviours demonstrated is, I contend, not only original but also a major breakthrough for the future acceptance of autonomous air vehicles.

1.7.5 The Identification of Accident Modes Relating to Pilot Remoteness

Although some Human Factors engineers have noted that UAS operators are affected by remoteness, I know of little work that has investigated that aspect in detail or identified that such remoteness leads to new classes of accidents. These classes can , of course, be added to the HFACS taxonomy.

1.7.6 The Nature of Autonomy and a Comparison with Automation

Several researchers, Weiner, Norman, Bainbridge etc. have identified the dangers of humans' relationship with automation and the danger that can bring. However, I know of little that has been researched to date on the relationship between autonomous systems and humans; certainly none to the detail herein. Nor, *ipso facto*, has there been such a comparison, as presented here, of the differences between humans controlling automatic as opposed to autonomous systems .

1.7.7 Validation of the SEBA Approach to Systems Design

Although not an aim of the thesis, it is a fact that the Autonomous Integrated Mission System (AIMS) decision architecture was developed according to the Synthetic Environment Based Acquisition and in doing so, validated much of the theoretically derived advantages of such a process, particularly that regarding integration risk, operator involvement at an early stage and incremental acquisition. This, I believe, is a first for such a large development programme over 5 years.

1.8 Publications and Presentations

1.8.1 Papers Submitted

The following papers have been published based on some of the findings in this Thesis:

- C H Patchett, V.V.Sastry: "*Decision Architectures for an Uninhabited Autonomous Air System*", 6th Eurosim Congress on Simulation and Modelling, Ljubljana, Slovenia, September 2007.

- C H Patchett, V.V.Sastry: "*A Preliminary Model of Accident Causality for Uninhabited Autonomous Air Systems and Its Implications for their Decision Architectures*", IEEE Tenth International Conference on Computer Modelling and Simulation, Cambridge, UK, April 2008

- C.H. Patchett, D. Ansell: ”*The Development of an Advanced Autonomous Integrated Mission System for Uninhabited Air Systems to Meet UK Airspace Requirements*”, CEAS 2009, Manchester, October 2009.

- C.H. Patchett, D. Ansell: ”*An Advanced Autonomous Mission System for an Uninhabited Air Vehicle to Meet UK Airspace Requirements*”, ISMS, Liverpool, 2010.

- C H Patchett, V.V.Sastry: “*The Pilot as an Intelligent Sensor: What UASs and Pilots are Missing*”, currently available in draft and awaiting publication, May 2011

1.8.2 Formal Presentations

A number of informal presentations of parts of this work has been presented, certainly too many to outline here. However, in addition to the papers presented at the conferences specified above, the following major formal presentations have been made:

- *“Control, Autonomy and Safety of UAASs”,* Autonomy Workshop, Shrivenham, Cranfield University, March 2009.

- *“Sense and Avoid Control Architectures for Uninhabited Air Systems”,* Student Symposium, Shrivenham, Cranfield University, June 2010.

- *“The Advanced Autonomous Integrated Mission System”,* to an audience of Members of the IET, BAE Systems Warton, November 2010.

- *“Certification of Civil Autonomous UASs: Research Questions and Issues”,* to members of the Civil Aviation Authority, November 2010.

2 Statement of the Problem

"When a distinguished but elderly scientist states that something is possible, he is almost certainly right. When he states that something is impossible, he is very probably wrong."

Clarke's First Law, Arthur C. Clarke

2.1 The Advent of UASs

In this Chapter, the evolution of unmanned flight is discussed. In particular, the nature and reasons for the compression of that evolution are presented. For some, the history of unmanned flight is merely a continued part of the history of manned flight and it is constructive to have a short but relevant review of this.

2.1.1 Manned flight

It is useful to momentarily reflect on the evolutionary scale and impact of the aeroplane on society in general. The aeroplane has made the world a far smaller place, has had an enormous effect on economics and warfare and has expanded man's horizons, not only physically but also mentally. It has given man that stepping stone to the "final frontier". And in such a short time – literally the span of a single lifetime. Between Louis Bleriot, the first man to fly across the channel (1909) and John Young, the first Space Shuttle pilot (1981), there are only 72 years. Between, the first manned flight of the Wright brothers and Chuck Yeager flying at Mach[12] 2.5 there are only 50 years.

The history of flight is really the history of **manned** flight. Without man's ingenuity, endeavour, and endurance in the face of failure, there would be no aeroplane. Above all, however, it is the skill, aura and sheer *bravura* of the pilot that has been always ascendant in the history of flight. He is seen as the necessary ingredient to make the plane perform, sometimes at the limits of its design and at the edges of the environment.

[12] Mach 1 is the local speed of sound, approximately 650mph at sea level.

That bravery is a necessary ingredient for a pilot is beyond question, especially when considering the early days of flight. The technology and limited understanding of materials and aerodynamics, coupled with the desire to expand the flight envelope, mainly for war, made the cockpit a dangerous place. In those early days, the aircraft was more likely to kill the pilot through mechanical failure than he was to kill himself through his own error. In the evolution of manned flight, that situation has been reversed.

2.1.1.1 The Changing Role of the Pilot

"There are old pilots and there are bold pilots – there are very few old and bold pilots". Traditional, Anon

In the early days of flight, pilots were true explorers, not to say adventurers – and they often did not last long! This exploration was multi-faceted. Not only were they exploring the environment and its physics, they were also exploring the theories of power, aerodynamics, structures and materials. However, it could be argued that the greatest exploration of all was that of flight control - the skills of the pilot, the response of the vehicle and the relationship between the two. The deeper the explorative venture, the higher the demands on pilot skill levels became.

This exploration, rapidly accelerated by the First World War, led to new demands on pilot skills. He was not only expected to be able to control his aircraft under all circumstances, he often had to be a tactician, a marksman or a crew leader, a navigator, an engineer and a radio operator. In particular, the concept of making good judgements of situations in order to complete the flight successfully, emerged - what we now call "Situational Awareness". A long standing Royal Air Force pilot adage, somewhat trite and school-boyish, but nevertheless, absolutely true, is:

"A superior pilot is one who uses his superior judgement to avoid situations that require the use of his superior skills". Anon.

In terms of flight control, automatic pilots or autopilots, have been around since the early days, aided by Charles Sperry's invention of the gyroscope in 1909, and have since progressed via "George[13]", through to the highly sophisticated Flight Control Systems (FCS) we have today, where an airliner is under automatic control shortly after take-off rotation until completion of braking on the destination runway, perhaps many thousand miles apart.

Similarly, it did not take long for machine guns in fighter aircraft to be fixed mounted and aimed by pointing the aircraft. Gun sight technology advanced from the reflector gun sight (to avoid parallax error) to radar ranged gunfights (reduction of fall of shot errors) and lead computing gunfights (reduction of aiming errors in turns). A similar story for bombing has also unfolded. In fact, the early post war bombers of the RAF (the V-Force) had a Navigation and Bombing System (NBS) where the radar operator could control the aircraft and the bombing computer[14] released the bombs automatically when the distance to the target equalled the forward throw of the bombs. This release, of course, had to be pre-authorised by the crew.

Similar stories can be told for the roles of the flight engineer, wireless operator and navigator, although these duties are sometimes retained for specialist areas. Commercial pilots maintain currency in their traditional skills for particular situations where the machines cannot help, but for routine flight, they are rarely exercised in practice.

However, the one thing that has not yet changed dramatically is the requirement for the pilot to exercise good situational judgement – and that requires the ability to project the current situation into the future and to plan accordingly. Machines can help but humans are still the best at situational awareness and planning,

[13] The nickname of the autopilot given by the crews of RAF heavy bombers in World War 2

[14] One of the finest and last true examples of analogue computing. For example, the solution to the perennial navigational problem of solving the "triangle of velocities" (knowing heading/true airspeed and wind velocity, then calculate track and groundspeed) was achieved by literally having a metal triangle with variable length sides, adjusting two sides manually and measuring the third. It was extremely accurate; much more than required for the delivery of a nuclear weapon.

particularly creatively, when a situation has not been foreseen. However, there are signs that even this last bastion of pilot skill is on the cusp of change.

2.1.1.2 Regulations and Procedures

When Bleriot crossed the Channel in 1909, he did not need approval; he was the only one up there. However, as flight became more common, means of directing traffic became necessary. In addition, Rules of the Air were introduced whereby air users could operate together in safety. These Rules were based on rules for the only other similar form of transport – the sea. They still are, despite plenty of evidence that the environments are fundamentally different!

As mentioned, the early aircraft were quite capable of killing the pilot and crew without their help, and so forms of regulations began to evolve which set standards for construction and licensing in order to improve safety. These have since evolved to a morass of regulations which have to be conformed to. These regulations, with an evolution of over a hundred odd years, have a single common factor. They all presume there is a pilot on board.

Hence we have a requirement for air users to “See and Avoid” other aircraft. Another is to remain “clear of cloud” or “within sight of the ground”. Others include statements such as “Not to descend below decision height in cloud unless “in sight of the runway”, “If there is smoke in the cockpit”; “if there are signs of fire” etc. In fact, there are countless similar ones.

In order to operate UASs, these requirements must be complied with - they are not going to go away (at least quickly). They have evolved over a hundred years, and thousands of air users comply with them every day. UASs will be expected to be no different.

2.1.1.3 Operating Environment

“What goes up, must come down – how it comes down is the important bit”.

The air environment is unforgiving, primarily because we have had millions of years to adapt to the ground environment and only 100 years or so for the air.

For humans, it is *unnatural*. The truth is our brains can cope, hopefully more often than not, but our bodies frequently let us down. There is a vast world of aero-medical physiology which deals with the effects of the air environment on our bodies. We have no birdlike sense of direction – we need a reliable compass, map and judgement of the wind. We have no sense of altitude in cloud, nor speed through the air. Even the dynamics of turning flight can give us a wrong sense of balance, even to the point of not knowing which way is up. In clear sky, we can use our eyes, but in cloud we rely, literally for our lives, on instruments.

As we go up, it gets colder, the air becomes thin and the partial pressure of oxygen in our lungs can get so dangerously low that we die. And there is weather to deal with. Not just cloud but thunderstorms, lightning, ice accretion, high winds, zero visibility, turbulence[15] – the list is very long.

Finally though, once flying, the only major concern is the landing. You can't pull over as in a car or head back out to sea in a ship. You can't halt things temporarily or press pause. Once things stop working, you've either landed or you're going down.

2.1.1.4 The Nature of Air Accidents – Human Error and Reliability

"To err is human; to forgive is often too late"

Because of the nature of the operating environment, human characteristics and the fact that machines have finite limits of reliability, there are always going to be accidents. The causes of these accidents are dealt with in detail later but suffice it to say that the breakdown of accident causality has changed considerably over the years. As mentioned, in the early years, when the sciences and machines were immature, aircraft were positively dangerous. Since then reliability has improved enormously, mainly due to improvements in design, materials, fuels, engines etc. and a recognition that the safest part of the flight envelope is in the middle – not at the extremes!

[15] And rather topically for the first time in UK aviation, volcanic ash!

That increase in maturity and reliability is not however matched by improvements in human abilities for evolutionary reasons. Nowadays, human error is responsible for about 75-80% of all air accidents. That is overall human error, not just pilot error. Human errors can occur due to poor organisational processes and culture, supervisory errors, including maintenance procedures and poor physiological states [3]. However, the pilot is the final barrier. He is the one who through poor judgement, lack of skill, bad decision making, risk-taking or just plain honest failing, causes the accident. Or, conversely, by using those extraordinary human skills to advantage, the pilot averts the accident and saves the passengers.

2.1.2 The Rise of UASs [*7*]

Most lay people will imagine that unmanned aircraft, let's call them UASs for now, are a modern invention – certainly in the last 40 years. In fact, the Englishmen, John Stringfellow and William Henson built a steam powered propeller driven model aircraft with a 10 foot wingspan called the Aerial Steam Carriage. This odd looking aircraft successfully flew for a distance of approximately 60 yards in 1848, 55 odd years before the Wright Brothers. During the American Civil War, attempts were made with balloons fitted with incendiaries and timers to try to set fire to Republican lines - the first flying bombs! However, they were largely unsuccessful. The American experimenter Samuel Langley successfully flew a steam powered model he called "Aerodrome Number 5" down the Potomac River for 3/4 of a mile 6 months before the Wright Brothers first manned flight. However most would agree that the most recognisable forerunners of today's UASs were the American Navy Curtiss/Sperry "flying bomb" which first flew in1918, together with the Charles Kettering Aerial Torpedo, also known as the Kettering Bug (see right). The "Bug" was a petrol fuelled propeller driven biplane which flew on a pre-set course for approximately 50 miles and was developed late in 1918. Many

thousands were due for manufacture (but only 36 were actually made) though it was rather unsuccessful, ironically, until the war ended when it then demonstrated a true capability

The key technologies enabling the use of UASs were the Sperry designed barometer/altimeter and gyrocompass (for automatic flight control) and radio transmissions (for real time and remote, rather than pre-set, operation). The use of radio control was originally envisaged by Nikola Tesla in 1898. He conceived, and developed, radio controlled operation of a pair of six feet long boats, capable of carrying an explosive charge[16]. These were demonstrated to a crowd at Madison Square Garden, New York who were amazed at the operation of remote control and thought miniature trained monkeys or Black Magic were responsible. Tesla referred to these boats as "Teleautomatons[17]" and envisaged scores of them attacking naval ships controlled by specially trained operators. He offered the concept to the US Navy and British Admiralty without success. With the advent of aircraft some 6 years later, this concept was quickly translated, by others, to the air environment.

With the onset of (relative) peace in 1918, the development of UASs slowed considerably but a new role emerged, that of aerial target tracking, which enabled weapon operators to hone their skills on live targets, with live weapons – the simulator age was still a way off! Many hundreds of UASs were manufactured for this role in the mid-1930s, the best known of which was the Fairey "Queen Bee"[18], and within which they were colloquially known as

[16] A perfect example that topicality is as powerful as technology. Science should be grateful that his invention never gained the interest of the US Navy or British Admiralty! This failure resulted in his teaming with Elmer Westinghouse for the development of modern alternating current electricity transmission – clearly far more useful to mankind.

[17] "*Telautomats will be ultimately produced, capable of acting as if possessed of their own intelligence, and their advent will create a revolution*". -Tesla, Nicola, in his book *My Inventions*, published in 1921. He also referred to them as tele-automatons. How true this all rings nowadays and what prescience.

[18] This was a derivative of the DeHavilland Tiger Moth (many DeHavilland aircraft were named after insects) and some 420 were build. A senior RAF officer remarked that " .. nearly all of them rendered valuable service over many years which says more about the state of contemporary UK anti-air defences than the resilience of the aircraft. Such UASs proved the vulnerability of ships prior to World War 2 (though most Naval people ignored the fact) and heralded the ascent of the aircraft carrier over the battleship.

"drones"[19]. The first really successful flying bombs though were German and developed in World War 2. These were the Henschel 293 rocket powered bomb and the Fritz-X glider bomb. They both had mid-course and terminal guidance using radio control directed by an operator on board the launch aircraft and both were highly successful. However, the most well-known, and it can be argued, the most successful[20], was the Fieseler Fi103 better known as the V-1 Flying Bomb, though this was unguided in the sense that it was aimed at its target and then launched.

A third role, in addition to aerial targets and flying bombs, was introduced in the mid-1950s – that of reconnaissance. The motivation for this is similar as for the other roles; simply that wartime reconnaissance missions are very dangerous [7]. The key enabling technologies were automatic camera operation and in-flight film processing and, as long as the targets were not too far away, reasonably large and static, good results were obtained. However, the key limitation was effective navigational accuracy. A side show concerning this aspect came about as a race between cruise and ballistic missiles. Cruise missiles could carry the weight of early nuclear warheads but suffered from navigational inaccuracy as the duration of the longer flight accumulated unacceptable errors. Ballistic missiles however, with their relatively short flight time were accurate but could not carry the weight. The race was decided by a massive reduction in warhead weight and size thus favouring the ballistic missile but, crucially for UASs, the race speeded the development of the inertial platform which enables autonomous navigation[21]. This factor, together with modern updates and by closely coupling the inertial platform with a Global

[19] The origin of the "Drone" descriptor for UASs, doggedly and annoyingly used by modern journalists (and no others!), is sometimes ascribed to the widely used Queen Bee target UAS developed in 1933 (although why a UAS with a female bee's name should be given a male bee's title is beyond me). However, the first recorded reference was by a US Navy Lieutenant, Delmer Fahrney in 1936.

[20] A British assessment in 1944 compared the cost of the V-1 campaign to the cost of its impact on the Allies concluded that the V-1 offered a 4:1 return on investment.

[21] Generally credited to Dr Charles Stark Draper who evolved the theory, invented the technology , develop, manufacture and field the first inertial referenced platform in 1949.

Positioning System (GPS), has enabled the advent of the UAS to where it stands today.

2.1.3 Bombs, Missiles and UASs - Differences and Similarities

There is clearly a grey area in differentiating between flying bombs, missiles aircraft and UASs, particularly visually. Obvious discriminators are that bombs and missiles are fundamentally disposable, non-returnable and lethal whereas the vast majority (if not all) of UASs have some sort of recovery system. However, those aspects aside, would one describe the Kettering bug (see photo above) as a bomb – it clearly is an aircraft without a pilot in it. Similarly, the *Kamikaze* pilots of World War 2 Japan used the *Baka*, obviously at first sight a manned missile, but they also used converted aircraft loaded with high explosive. The specialist target UASs were clearly a cross between a missile and an aircraft, as was the V1 flying bomb, known colloquially as a Doodlebug[22]. Its successor, the V2, is obviously (to us) a missile and yet it had the same objective (and target – London) Perhaps because it had a ballistic trajectory and no wings, it qualifies as a missile. However, nearly all non-ballistic missiles have wings. It's not clear cut and it is confusing – even to the so called experts!

.More importantly however, because of the above mentioned key aspects of UASs, bombs and missiles, we can start to determine differentiators from manned aircraft in the areas of versatility, cost, reliability and safety – fundamental aspects that require addressing for modern UAS operation and, as far as autonomous operation is concerned, the underlying basis for much of this work.

[22] This could properly be referred to as a drone, as its pulse jet engine made a characteristic drone like sound.

2.2 Differentiators between UASs and Manned Aircraft

2.2.1 UAS Employments and Versatility

As related, UASs, certainly for their first 50 odd years were extensively and almost exclusively used as aerial targets or flying weapons. These roles were augmented with that of reconnaissance in the mid-1950s. Since unmanned aircraft inherently operate without life support equipment, their range/duration can be extended, theoretically almost indefinitely. These considerations have led to the oft quoted preferred, or even essential, UAS roles as:

- Dangerous – e.g. battlefield reconnaissance, either photographic or electronic. Politically, UASs are excellent for this role. If discovered, blame can be apportioned to the computers "malfunctioning"[23]. If shot down, there is no pilot to stand trial or be used as a hostage. To this must be added the obvious reduction in the loss of human life that typically arise from these missions being undertaken by UASs and all that accompanies it.
- Dirty – e.g. nuclear reconnaissance[24], volcanic ash sampling[25]. UASs can clearly undertake roles for which there is no (current) manned equivalent.
- Dull – long (i.e. > 24hour) endurance missions. These tend to be the most interesting from engineering and human factors viewpoints. Current experience in operating aircraft in long endurance missions is lacking and certain factors assume a lot higher importance than previously. One factor is reliability. Most aircraft consume their fatigue life during take-off and landings. If the proportion of such events relative to flight hours is drastically reduced, then either the airframe can be made lighter, or the

[23] A Snark missile fired eastwards from Florida in 1956 disappeared and was discovered by a Brazilian farmer in 1982 [7]

[24] "Sniffing" for radioactive fallout after nuclear weapon tests was a routine mission during the Cold War and resulted in many crews suffering from radiation sickness despite wearing lead-lined flying suits (and drowning after ejection because of the weight of them).

[25] When Mount Usu erupted in Japan in 2000, the country's UAS fleet, normally occupied on agricultural duties, were deployed onto this task.

life time of the vehicle is extended. Equipments, particularly engines, mission or safety critical avionics, may have to be redesigned to have a much higher Mean Time Between Failures (MTBF) than considered previously in order to realise the extra endurance. In the area of Human Factors, long endurance missions require operators to perform handovers, often many times during a mission and there is quite a body of evidence, which will be referred to later, that such handovers can be a fertile ground for causing accidents.

As previously mentioned, reconnaissance remains the predominant role of the UAS – both in military and civilian usage - and this role is constantly being extended. For military reconnaissance, there is a sensing gap in the atmosphere. Manned aircraft can routinely fly up to about 50,000ft, whilst more specialist, and it has to be acknowledged, more expensive, aircraft can fly up to about 85,000ft. Far above this, the satellites in Low Earth Orbit (LEO)[26] operate; the lowest generally about 200km (656,000ft) altitude. The role that UASs are now seen to fill is reconnaissance in the region above conventional aircraft and below that of the satellites. Whilst the specialist aircraft can operate for several hours, a UAS, such as Boeing's *Phantom Eye* Demonstrator is designed to endure for 4 days. The operational version should extend this to 7-10 days. At 65,000ft, such an aircraft can cover a radius of 965km with a single sweep of an rotating antenna (covering an area about the size of Afghanistan). This extension in capability is in response to the military requirement for continuous surveillance (so called "unblinking" coverage) of large areas.

However, the single factor that stands out when considering UAS employment is that the missions that are currently flown are generally simple and require little versatility[27]. For example, the vast majority of Predator missions can be described as: take off from A and fly to area B; search area B and transmit

[26] Defined by NASA as "*Low Earth orbit (LEO)* - The region of space below the altitude of 2000 km" cited in "NASA Safety Standard: Guidelines and Assessment Procedures for Limiting Orbital Debris, Office of Safety and Mission Assurance , Washington. August 1995.

[27] Versatility here is meant to imply operation in environments that are complex or operations that are complex in themselves. It does not suggest here the ability to do several (simple) things at once. (Some authors prefer the term "flexibility").

photographs; return to A and land. There are few alternative goals, such as: process imagery, avoid other objects and the ground, manage fuel efficiently, re-role in flight etc. Similarly, the vast majority of civil UASs are used for agricultural purposes (in Japan), primarily crop spraying and photography. Again, a simple and uncomplicated mission with few conflicting goals or tasks and requiring fairly low levels of performance and versatility to achieve the objective.

It may be considered that the mission of the airliner pilot is similar in terms of versatility, but even a cursory contemplation indicates that it most certainly is not. Although he has a sophisticated flight management system which will assist in fuel management, flight control, navigation, landing, and sometimes take off, the airline pilot has to consider a variety of factors such as weather, both en-route and in the terminal areas, other air users, integration with the Air Traffic Management system and most importantly, conform to a plethora of regulations under a wide variety of conditions. All these, and much more, must be done to a very high standard, to high levels of safety and sometimes under extreme pressure. The pilot is there because he can provide the level of versatility required to maintain overall man/machine performance in achieving the mission.

UASs, in general, do not fly in unrestricted airspace and therefore many of the demands that this places on the man/machine interface are absent, particularly in the aforementioned areas. Combat UASs[28] do not currently attack other (manned or unmanned) aircraft and it is interesting to ponder why. Clearly they may shoot down an airliner (instantly categorised as “Catastrophic”) or the wrong target, at the wrong time or the wrong place - and this is obviously unacceptable. However, the nature of aerial combat, which requires high levels of tactical skill and instantaneous Situational Awareness, means that the mission may be just too demanding, in terms of versatility, for current UASs. This is despite the fact that a UAS can potentially turn at G levels well beyond

[28] There is an uncorroborated report of an exchange of missiles between a Predator A and an Iraqi MiG 25. The Predator lost!

human capacity. In fact, combat UASs rarely, if at all, have been assigned missions to attack humans who can fight back with any chance of success.

By comparison, manned aircraft are much more versatile and undertake more roles than UASs at present, particularly in the application of lethal force and the reduction of collateral damage. The conclusion one is forced to draw is that building the versatility of manned aircraft into UASs, even with the operator in the loop (but always remote), is currently just too difficult.

2.2.2 UASs and Cost

Cost, or more properly put, cost effectiveness is frequently cited [8] as one of the two main reasons for replacing manned aircraft with unmanned ones. Cost, when applied to an aeroplane, is not a simple variable and has many ingredients and attributes. A fairly simple example is the relationship between non-recurring (e.g. manufacture) and recurring (e.g. operation) costs. Manned military aircraft are increasingly more expensive but at the same time increasingly more capable and operate in a variety of roles (but not of course simultaneously). Where flexibility, survivability and rapid response are key requirements, manned aircraft are more cost-effective. Where speciality, expendability, cost and endurance are uppermost then UASs are the preferred choice [9]. Reference [9] also cites that “it is cheaper and faster to produce an entire squadron of medium range UASs for the price and effort required to replace a single F-14 .. and .. though the UASs are not strictly comparable to the F-14 due to the large differences in capabilities of the platforms, the figures do tend to highlight the fact that UASs can be produced cheaper” (taken from [10]).

So, it appears that the consensus of opinion is that UASs are currently cheaper to procure, and possibly operate, than manned aircraft for equivalent size, weight and performance and for a singular role. However, that is not to say they are cheap. As previously stated, Global Hawk is currently priced at about $80m (and rising); Predator A is about $6m and Predator B about $25m.

Airframe empty weight is frequently used in the aircraft industry as a rough metric of aircraft costs. The table below gives an indication of the costs/weights of a variety of UASs:

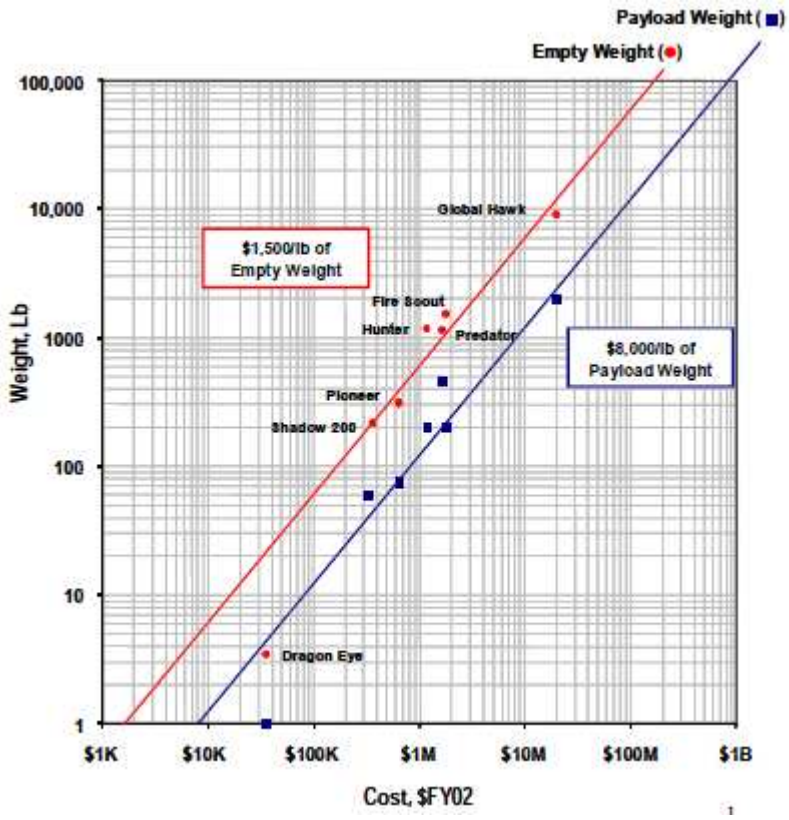


Figure 1: UAS Cost Vs. Weight ($/Lb.) [11]

Similarly, the analysis below gives an indication of the costs/weights of a variety of manned aircraft.

Figure 2: Cost Vs. Weight ($/Kg) for a Variety of Manned Aircraft[29]

A comparison table drawing the salient features of these graphs is provided below:

Aircraft Type	**Cost/Weight ($/Kg)[30]**
Fighter/Multi Role Aircraft	6700
Military Transport Aircraft	2100
Commercial Transport Aircraft	1700
UAS (Including Payload)	20768
UAS (Weight Empty)	3894

Table 1: Comparison of Aircraft Costs Vs. Weight ($/Kg)

These graphs and the above table tend to not support the generally held view that UASs are cheaper than manned platforms if we characterise them by cost/weight. However, the caveat must be applied that at least one reason for the high costs of advanced fighters and unmanned reconnaissance aircraft is the cost of the sensor suite, which can sometimes double the airframe cost. This is of course generally a non-recurring cost. It could also be that better

[29] An anonymous but apparently well researched and referenced article: "*An Aircraft Worth its Weight in Gold*", available at http://theblogbyJavier.wordpress.com/2010/03/13/an-aircraft-worth-its-weight-in-gold/

[30] These costs are estimated at Financial Year (FY) 2008. The FY 2002 costs previously shown are assumed to increase by 3%p.a. for inflation (US Bureau of Labor Statistics)

metrics, capturing some of the realities of capabilities should now be used when making comparisons, such as: pixels per hour or cost/kg/hour (endurance). If we take into account endurance[31] on the above figures:

Aircraft Type	Cost/Weight/Endurance ($/Kg/Hr.)
Fighter/Multi Role Aircraft	3350
Military Transport Aircraft	300
Commercial Transport Aircraft	212
UAS (Including Payload)	865
UAS (Weight Empty)	162

Table 2: Comparison of Aircraft Costs Vs. Weight vs. Endurance ($/Kg/Hr.)

These figures now reflect a general equity between manned and unmanned aircraft. Unfortunately, they also show that different metrics lead to different conclusions – the reader is invited to pick!

When it comes to service support and training, other differentiators can come into play. The range of metrics to consider is vast but one example is the cost of training a ground based commander or sensor operator which is lower than for a combat pilot as depicted in the tables below [12]:

SUPT (fighter/bomber track)	$392,861
B-52 IQT	+292,190
Total	$685,051

Source: Air Combat Command/XOFT. This table uses a B-52 pilot as a valid sample of several Predator pilots, past and present, who maintain the B-52 as their MWS. Also, these figures do not include the cost of B-52 mission qualification training, B-52 requalification training after the Predator tour, survival schools, altitude-chamber training, life-support training, and so forth.

IFT	$5,500
Instrument rating	6,500
Hi-fidelity simulator check	+1,000
Total	$13,000

Source: Air Education and Training Command/ XOFT. The cost is only $1,000 if the nonrated selectee already possesses an instrument rating. The table does not include the cost of Predator IQT because a B-52 pilot under the old system would still have to attend Predator IQT; therefore, the cost would be the same.

Table 3: Comparison of Training Costs for a UAS and B52 Pilot

For a Squadron of 15 pilots, that generates savings of $10m. Furthermore, crew currency can be maintained far more cheaply, particularly if the UAS is

[31] Endurances used are: Fighters – 2, Military Transport – 7, Commercial Transport – 8, UAS - 24

equipped with automatic take-off and landing systems, thus not requiring direct flying skills to be maintained. Hitherto, the USAF required its UAS pilots to be qualified officer pilots whereas other services have enlisted men as pilots. Again this can translate to cost savings. Equally, by not deploying crews in the field, cost savings can be easily identified such as transport, maintenance, support and infrastructure, pay and conditions etc. Finally, the real cost benefit, that of the saving of human loss of life, especially when considering the exemplar dull, dirty and dangerous missions, cannot be readily quantified but must certainly be taken into account.

The research to date has failed to pin down high quality literature of the nature of UAS costs. Much of what is available identifies some of the areas of savings (but frequently not all) but often is lacking in quantative data and referenced material and thus remains somewhat conjectural. In fairness, aircraft manufacturers and operators have been arguing over the best method of characterising aircraft costs for many years and the fact remains that the rising and rapid advent of UASs, together with their unusual and hitherto unique features, makes this a difficult subject.

The general consensus in the literature and the conclusion reached here, is that UASs have considerable potential for cost benefits compared to manned aircraft operation but much of that potential has yet to be reached.

2.2.3 UASs and Safety

The design and build quality of UASs has changed remarkably over the past eight years (the majority of the time taken for this study). Consequently so has the safety of UASs. In 2002. Global Hawk has an accident rate of 157 per 100,000 flight hours. Today it is less than 30 with only one published accident occurring in the last eight years. Predator and its variants show a similar record. These statistics, excusing the pun, are no accident. It is a fact, for all aircraft, that the accident rate drops as the system enters service and starts to accumulate flight hours[32]. This is due to a variety of reasons. These may be

[32] Evidence for this is given later and in more detail.

maintenance and operating experience, replacement of weak components or design flaws that come to light, implementation of accident investigation recommendations, improvements in materials. However, it remains true that general safety records of UASs, although improving and approaching that of General Aviation, show that they are still demonstrably more unsafe.

There is a clear trade-off between cost and safety and the fact that the vehicle is unmanned allows an extension of that aspect. This is easily seen when one considers the expendability of the vehicle – no manned aircraft is ever considered expendable. However, there are many small military UASs that are designed to be expendable and there may be occasions when even the large ones are sacrificed to achieve a desirable outcome[33]. However, the link between cost and safety, although subtly different for UASs compared to manned aircraft, cannot be ignored.

So what causes UASs to crash? The USA DoD conducted a UAS Reliability Study in 2003 [13] and presented the following graph based on data collected from 1986-2002 covering the Predator, Pioneer and Hunter UAS fleets:

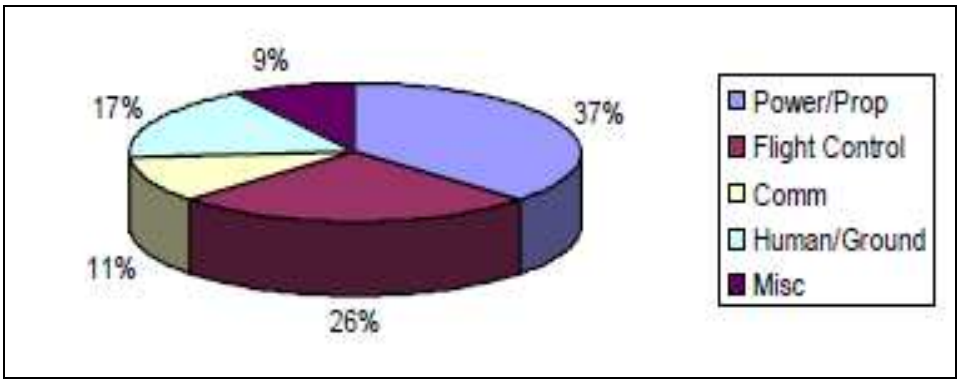


Figure 3: Average Sources of System Failures for U.S. Military UAS Fleet (Based On 100,000 Hours)

[33] A perfect example is an emergency manoeuvre to avoid a mid-air collision. One can countenance ordering a manoeuvre so violent that it results in losing vehicle control and consequently causes a crash into the ground in order to achieve safe separation and avoid the collision

It should be borne in mind that this is a general statement of what was true 7/8 years ago and things have improved since then for a huge variety of reasons, all of which are covered in the Reference.

In contrast, statistics for General Aviation in the USA shows the following breakdown of accident causes (taken from the 2006 Nall Report [38])

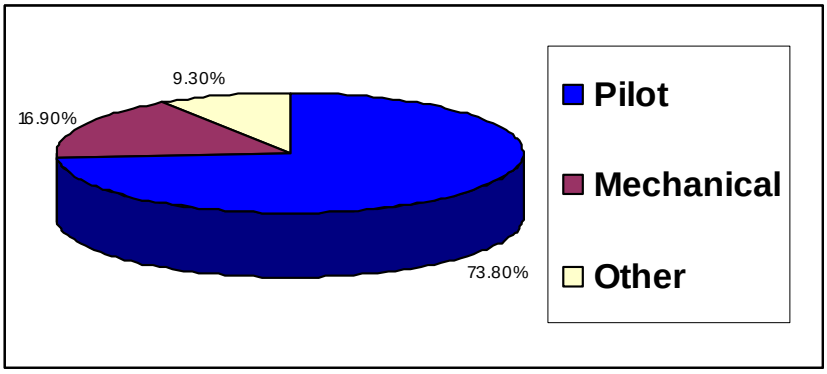


Figure 4: Major Causes of USA GA Accidents 2006

The breakdown of the "Mechanical" sector into components is shown below:

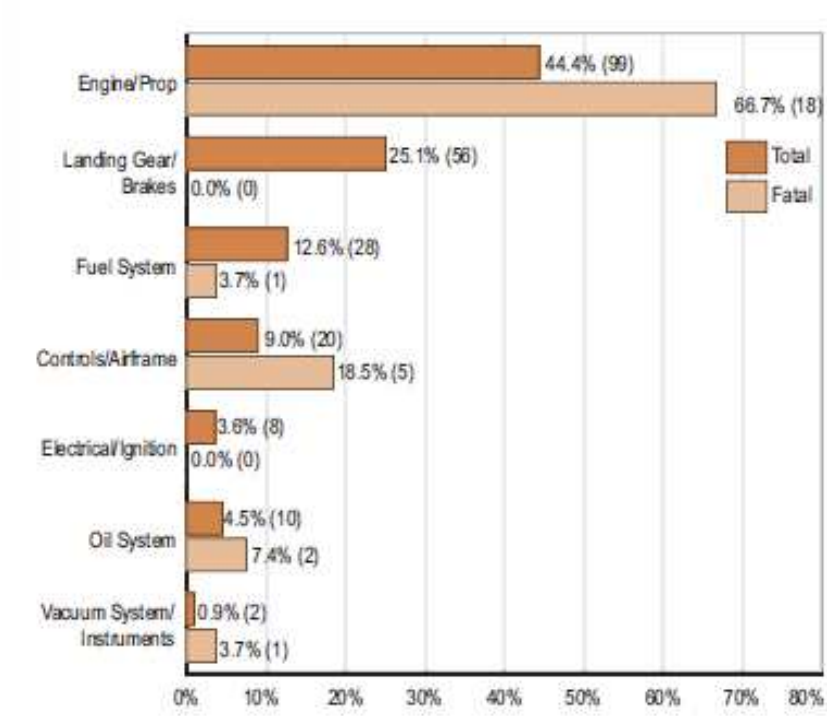


Figure 5: USE GA Accident Causes 2006 – Mechanical/Maintenance

However, caution should be observed before making an analysis of these differences due to the facts that are from slightly different years (although GA data tends to remain roughly constant over this decade), they both cover a variety of different aircraft and the systems on-board are often very different[34]. Having said that, it is quite clear that pilot error is the overwhelming cause of manned aircraft accidents whilst mechanical failure is predominant for UASs. In fact, a casual glance at these statistics (wrongly) implies that UASs are easier to fly. This is not quite so. It's just that mechanical failure is the overwhelming influence for accidents. For sub categories of maintenance /mechanical failure, the predominance of propulsion failures is remarkably similar. For flight control, the statistics cannot be compared since both types have completely different control system designs. This aspect will be covered later.

In the UAS Reliability Study of 2003, a comparison was given of manned/unmanned reliability and this is below:

Aircraft	Mishap Rate (per 100,000 hrs)	MTBF (hours)	Availability	Reliability
General Aviation	1.22	*Data proprietary or otherwise unavailable*		
AV-8B	10.7		*Data unavailable*	
U-2	6.5	105.0		96.1%
F-16	3.35	51.3		96.6%
F-18	3.2			
Boeing 747	.013*	532.3	98.6%	98.7%
Boeing 777	.013*	570.2	99.1%	99.2%
Predator/RQ-1	31	55.1	93%	89%

*NTSB data for all commercial air carriers operating under 14 CFR 121.

Table 4: Examples of Manned/Unmanned Aircraft Reliability

This clearly highlights the differences which are particularly apparent for airliners and Predator.

So the first question to answer is "Why are UASs so unreliable"? Firstly, manned aircraft obviously are designed from the outset with safety in mind (they

[34] For example, the majority of GA aircraft do not have autopilots or software embedded systems.

cannot be introduced into service without stringent analysis and verification – a process known as certification). Military UASs are not subject to the same process and they are not perceived to require the same levels of safety as manned aircraft. Additionally, it must be said that UASs (nearly all of them) have been put into production based on experimental prototypes. Cost, weight, performance and time into service, not reliability or safety, were, and still are, seen as the most important aspects. Due to this, cost savings and short cuts have often been made in design and materials. In addition, military UASs have frequently been deployed into an operational theatre early in their lives and are frequently placed at risk through hostile action – remember the 3D[35]s tag for UASs. Military commanders were quick to realise the capability of these vehicles and demanded more at whatever price. As more vehicles went into service, around 2003 onwards, it was realised that the cost/benefit trade off needed re-assessing (hence the motivation for the reliability study). At this point, improvements were steadily introduced but at additional cost. A perfect example is Global Hawk. This was rushed into service, had four accidents in 3 years each costing about $30m. Subsequently the design was improved and, consequently, it now costs about $80m but has had only one accident since. Despite accumulating an exponential increase in flight hours, Predator, a far cheaper aircraft has not had a complete re-design (it was more mature anyway) but is now more reliable and in turn costs more to replace. Having said that there are still many Predator accidents caused by single point failure of critical systems (i.e. no redundancy). It is therefore reasonably clear that the cost/reliability trade-off is complex in nature and likely to be different for different UAS classes. The USAF Reliability Study recommended improvements to the propulsion (although nearly all are single engine) and flight control systems together with improvements for operator training. This was implemented (to a degree) and, together with the fact that the systems have become more mature, the accident rate has subsequently reduced. This is discussed in detail later.

[35] As a reminder – Dull, Dirty and for this case, Dangerous.

Nearly all that has been said applies to military UASs since they are the most prevalent and have the most flight hours from which facts can be ascertained. However, in this period, the capability they have given commanders[36] is several orders of magnitude greater than previously. This growth in capability has not gone unnoticed in the civilian community[37] and there is now an increasing awareness of the potential uses of UASs for civilian purposes. However, the requirements for safety in this arena are much more demanding and will have to be met before UASs can routinely operate with other air users. The CAA is on record in saying that UASs will not be allowed to operate in UK civilian airspace until they are "at least as safe as General Aviation". Current experience with military UASs shows that there is still some way to go before this can be demonstrated.

In summary, over the last ten to fifteen years there has been a growing realisation that UASs are not throw away objects. They can confer a lot of capability but can cost a lot of money. It makes good financial sense to design, manufacture and operate them to standards approaching that mandated for manned aircraft. For civilian operation, it is likely that this will be obligatory.

2.2.4 UASs and Communications

Communications is a huge differentiator between manned aircraft and UAS operations. So large is the difference, that any sort of full discussion is outside the scope of this thesis thus necessitating only a brief outline of the subject here.

Situating the pilot remotely inherently sets the requirement for an external communication link between the controller and the vehicle – the co-called Command or C^2 (Command and Control) link. Such a link is, of course, entirely absent in manned aircraft where the pilot maintains the connection between aircraft's current status (the displays) and his demands (the controls). More

[36] It has been reported that there are three Predators airborne over Afghanistan permanently (24/7) on reconnaissance duties.

[37] The country operating the most civilian UASs is Japan where they are deployed on agricultural duties in large numbers.

importantly, in manned aircraft, this link is virtually instantaneous, and is designed to be (and is fairly easy to make) highly reliable. The fact that this is not the case in UASs is the cause of many an accident – several of which will be discussed in detail later. In addition, in manned aircraft, the pilot's awareness of aircraft state and status from his displays is augmented by that from his own senses. Not so in UASs. The nature of the link is conventionally by radio in either Line of Sight (LOS) or Beyond Line of Sight (BLOS) modes. To operate anywhere away from the local area requires the latter and for this, the medium of choice is invariably Satellite Communications (which we will abbreviate to SatComms forthwith). Unfortunately, SatComms has a few undesirable characteristics. Firstly is the bandwidth limitation. UASs can consume a large amount of bandwidth, especially when several are in the air at once [14]. This bandwidth comes at a cost and, if civilian satellite constellations are used, the UAS competes with other users such as TV channels and of course other UASs. This means that bandwidth may become unavailable at short notice. SatComms antennas can often be blanked from satellite view by the airframe when in a turn which leads to short term outages. Latency is an issue for SatComms and it is commonplace for transmissions to be received several seconds late. Finally, satellite constellations are vulnerable to physical phenomena (such as sunspots[38] and other radiations) and again performance can become degraded or unavailable. All this means that the traditional high bandwidth, availability and reliability of the pilot vehicle interface in conventional manned aircraft cannot be taken for granted in a UAS.

The C^2 link is not the only communications required for UAS operation. For military reconnaissance UASs, the mission payload[39] frequently has to download sensor data and imagery (usually the higher resolution the better), often in real-time, to its Mission Exploitation Ground Station. Again, this carries penalties in bandwidth.

[38] Topically, a huge degree of sunspot interference is predicted for 2012, perhaps necessitating a temporary shutdown of National Grids to avoid permanent damage.

[39] Military reconnaissance UASs typically carry a suite of Electro-optic, Radar and TV systems as a mission payload.

Finally, UASs required to operate in non-segregated or busy airspace have to comply with the Air Traffic Management (ATM) system which controls and co-ordinates air movements. For civilian traffic, this is routinely done using Very High Frequency (VHF) LOS voice communications. A UAS operating under such conditions must be able to comply in accordance with the principles of Equivalence and Transparency. Designing a UAS communications architecture to meet these requirements is not a simple exercise since they usually require that:

- VHF transmissions emanate from the UAS[40]. This requires that uplink SatComms transmissions are relayed over VHF and UAS VHF receptions are transmitted on the SatComms downlink, in real time, to the GCS (thus using more bandwidth).

- The GCS must have some alternative means of contacting the Air Traffic Controller (and *vice versa*), usually by landline, to cater for emergencies. This contradicts the principle of Transparency.

- The cognitive communications medium is speech. If a reversionary mode is invoked, there seems to no recourse other than to text-to-speech (fairly easy) and speech recognition[41] (fairly hard) technologies.

A solution to some of these problems is currently being operated in the Maastricht Sector of Eurocontrol for flights above Flight Level[42] (FL) 245. It is known as Cockpit Pilot Data Link Control (CPDLC) and provides an alternative

[40] Air Traffic Controllers, particularly those at aerodromes, usually utilise VHF direction finding equipment (VDF) to rapidly correlate the aircraft speaking to them with their Radar picture. Additionally, transmissions from the aircraft ensure that all aircraft within LOS of the transmitting aircraft can pick up the transmissions and analyse their content for enhanced Situational Awareness.

[41] The vocabulary used by ATC and pilots is tightly specified (in the UK by CAA publication CAP 413) which makes speech recognition far more accurate and therefore more reliable. However, particularly in GA, this standard is frequently not adhered to and *ad hoc* messages are commonplace.

[42] Aircraft above the transition altitude (generally 3000ft in the UK) fly at Flight Levels, which is the altitude in hundreds of feet when the Standard Atmosphere pressure datum

Figure 6: Cockpit Pilot Data Link Control Display (Airbus A330)

to VHF communications. It also offers far higher data integrity because of encryption, virtually guarantees reception at the targeted receiver and prevents garbled or misunderstood voice messages. It is, however, not widespread (Maastricht is the only sector of more than a hundred currently using it). Although, increased use of CPDLC is expected over the coming years, it is only a partial solution to the problem. For the thousands of light aircraft in General Aviation, VHF speech will be the primary communications medium for many years to come.

In summary, controllers of UASs (Air Traffic, Mission Payload, and Pilots) require a communications architecture that is required to be safe, reliable and capable of integrating within the present and future communications architectures and with existing air users. It should also operate at an equivalent level of performance to manned aircraft in terms of connectivity, availability and bandwidth. Such an architecture is likely to be complex and difficult to design in order to satisfy these requirements. Consequently, it is imperative that the UAS should be able to operate independently of its controller, at least for short intervals.

(1013.25mb) is set. Thus at FL245, the altimeter will read 24,500ft.

2.2.5 UASs and Control

Of all the differences between manned and unmanned flight, how the vehicle is controlled represents the biggest aspect. UASs can be flown in a variety of control modes all of which have in common the fact the pilots, commander or operator(s) are remote from the vehicle, either within sight or beyond line of sight.

There are many concepts of control moding. Some of these are:

2.2.5.1 Remotely Piloted

In this mode, the aircraft is flown manually from the ground control station at all times in a manner somewhat similar to a Radio Controlled model. This is the most common mode of control, particularly for light UASs. Such operation requires a high level of skill and many accidents of vehicles using this type of operation are skill based errors, the majority being prevalent in the take-off and landing phases of flight. In addition to the skilled operator, a communication link of high integrity and negligible latency is required. This can often only be achieved with Line of Sight (LOS) communication links thus restricting the range of operation to the limit of the Radar Horizon[43]. However, generally they are frequently operated within visual range since that is the usual means of control feedback to the operator. Such operation can be augmented with automatic take-off, landing and cruise modes. In summary, the pilot is in direct (and constant) control of the vehicle.

2.2.5.2 Supervised Automation

Here the aircraft is flown under full autopilot control in accordance with a specified route plan. This may, or may not, include provision for alternative routes. Such operation allows use of Beyond Line of Sight (BLOS) datalinks since real time feedback control is not now necessary. Versatility of operation can only be incorporated by uploading new routes or waypoints to fly. In

[43] The straight line path of a radio wave, typically about 100nm for a vehicle at 10,000ft altitude.

summary, the pilot is now a supervisor of the automation which is directly controlling the vehicle.

2.2.5.3 Autonomous Operation

The aircraft is flown according to an agenda of objectives which can be specified by the pilot, the on-board autonomous control system or jointly by both in co-operation. In summary, the pilot is now a manager of the autonomous system which is controlling the vehicle.

Of the above control philosophies, the trend is that we are rapidly moving toward collaborative human-machine decision making or fully autonomous decision-making rather than relying on human supervisors of autonomous systems, particularly if operators are not on-board [15].

There are a few features of such control modes that are common to each other but different to manned aircraft. A major one is pilot remoteness. It has already been noted that the immediacy and reliability of the conventional pilot - vehicle link in manned aircraft is absent in UASs. This inherent remoteness of the pilot or operator of a UAS introduces problems, some subtle, others not so. Some of these problems are now discussed.

2.2.6 Operator Remoteness

The pilot can be viewed as a multi sensor, information processing system. The most important of those sensors, the eyes, is replaced by a “Sense and Avoid” sub-system which replaces the “See and Avoid” responsibility we would normally assign to the pilot of a vehicle. Humans have far more senses that the traditional five normally assigned by the layman – sight hearing, touch, taste and smell. It is also true to say that some of these have a much higher importance when removed to the air domain than would be normally expected. Some but not all, of the less well known are: pain, joint movement (position and acceleration), balance and acceleration (vestibular), sense of time, temperature difference, and other minor ones not including those of internal body sensing.

In addition, humans are naturally pre-disposed to associate data from different sensors, occasionally incorrectly[44], thus reinforcing perceptions to promote a single cause from the combined effects – vibration and noise mean more concurrently than separately, as does heat and smoke, fire and light, aircraft movement and vestibular responses, thunder and lightning etc.

Some examples of how humans respond, and the procedures they follow in the event of emergency are given below. It should become plain that these are fully based on past experience with the pilot in the loop:

- An aircraft fire is often first noticed by aircrew by the smell of smoke. This may be later followed by visual signs of smoke. Aircrew associate the two and diagnose the possibility of a fire. This in itself triggers a procedural response using emergency checklists where “Actions in the event of a Fire” are proscribed. One of these responses is to seek for and confirm or otherwise, other indications of a fire. A fire light may come on and then go out[45]; fire may be seen; the aircraft is trailing smoke; an accompanying aircraft or the aerodrome personnel may report the smoke/fire. Now consider translating that situation to one where the pilot is remote. The pilot would certainly not see, smell or confirm (if they knew) the presence of fire. They have neither the on-board sensors to do so, nor could they otherwise deduce it. The only response a UAS designer can do is to put more fire sensors, and perhaps smoke detectors, in the vehicle. Thus the manned aircraft the “standard” procedure for identifying and confirming a fire as contained in the checklists is now inappropriate when applied to a UAS.

- In descending in cloud, with the outside air temperature less than 5C, rain is noticed on the windscreen. The pilot would immediately identify the likelihood of icing of the airframe, engines and the pitot static system which

[44] One of the more famous (though tragic) examples of this is the East Midlands 737 accident of 1989, where the right engine was shut down after the fan blades of the left engine failed. Ironically, the crew were not sure which was the faulty engine when 150 odd passengers behind them, and who could see out of the window, knew exactly which one it was.

[45] A fire warning light going out either means the fire has gone out or (more likely) that the fire has just burnt through the fire sensor or wiring!

provides vital air data. In this event, he would respond by getting clear of cloud or precipitation thus avoiding the dangerous situation. Again we could put rain/ice sensors on the airframe but do we in practice? The answer is no because we (the designers) “never thought of that”. We assume the aircrew are there when they are not.

- Similarly, the pilot feels severe turbulence and manoeuvres to avoid the situation. He may notice lightning in the vicinity. He takes appropriate action. What does the ground based pilot either know or do and, as we shall later explore, it is possible that he may not even care.

So, in removing the pilot, the vehicle is being denied important sensor information that he normally provides (for free) together with his sophisticated sensor fusion and associated processing. As mentioned above, the UAS designers could add these extra sensors as a replacement for the pilot’s sensors. In doing so, however, another problem will emerge and that is the increased possibility of false alarms (the more sensors, the higher the number of false alarms). This is a standard problem for designers – how to achieve a high level of true alarms with a low probability of false alarms. If those false alarms trigger safety critical responses, that probability may be required to be as low as $1x10^{-6}$ – a requirement that may be extremely difficult, and costly, to achieve. Thus, the implementation of the remedy may be more damaging than ignoring the problem. Certainly costs would go up.

Similarly, in removing the pilot, he is being denied important vehicle information such as peripheral vision, tactile feedback etc. Consider a conversation with a person in front of you. Aurally, you may (or may not) be concentrating on what they are saying or the emphasis on how they are saying it, but research has shown that communication also takes into account facial expressions, the body posture (language), movement of the hands, shaking or a nodding of the head etc. This extra information can be vital in fully understanding their views. If we now translate that to a pilot trying to understand what may be a complicated situation, then the denial of that extra, subtle and background information may be the difference between a correct decision/response and an unsafe one. He is

also being denied the opportunity to proactively investigate the situation because he is blissfully unaware of it.

Another aspect of removing the pilot, is that he is being denied his natural instincts of self-protection even if perhaps sub-consciously. This can have the following effects:

- He is less immersed (or more detached) in his attention to the state and status of the vehicle which could lead to cognitive disengagement.

- He is less aware of the danger to the vehicle but quite aware of the lack of danger to himself. This could make him prone to take greater risks or contemplate action which he would otherwise never ordinarily do were he aboard.

Latency has already been discussed in terms of communication aspects. Direct control feedback is hampered by latency, and UAS control signals can be quite latent. Anyone who had a computer mouse that did not respond in real-time, or a computer function that did not respond to the mouse, will know the sensation of such loss of control. Imagine a pilot who moved the control column of an aircraft and then waited for a response up to 4 seconds later. Or even worse, waited for a few minutes and, in the air, a lot can happen in that time. It is also not just direct control. Demanding the autopilot to turn onto a new heading is simple, direct and trustworthy in a manned aircraft, but not necessarily so in an unmanned one. Humans in control usually insist on direct and immediate feedback of demand inputs. Unfortunately, such tight human control cannot always be guaranteed for UASs.

To summarise, in a UAS:

- The operator is remote

- The operator’s Situational Awareness (SA) may be different from the on board Autonomous System (AS)

- Control is shared between the AS and the operator

In a manned aircraft:

- The pilot is in the vehicle
- The pilot's SA is immediate and direct
- The pilot is in sole control of the vehicle

2.3 The Motivation Towards Higher Levels of Autonomy

It will come as no surprise that the primary motivation for introducing autonomy into UASs is to overcome the negative aspects of those differentiators between UASs and manned aircraft and introduce corresponding increases in the levels of versatility, cost, safety and performance. How this may be achieved, and at what cost, is discussed below.

2.3.1 Versatility

As previously discussed, UASs currently undertake roles that require little versatility despite a human being in overall control. Versatile operation requires the ability to:

- rapidly assess current states and predict likely future states
- identify alternate courses of action and assess their chances of success
- to choose appropriate action and to monitor progress

Such abilities have a root requirement to observe and comprehend the operating environment and its enclosed objects in real time – a process known as generating situational awareness. A remote controller is, by nature, less able to achieve this than a pilot of a manned aircraft for reasons discussed earlier. As for the UAS itself, none is currently fielded that can generate self-situational awareness.

It was argued earlier that the pilot of an airliner operates in a complex environment where identification of changes of internal and external states is vital to the achievement of the mission objective. For UASs to operate in the same environment will require it to augment the deficiencies caused by remote operation and transmit this augmented data to the pilot in real time. This is not a trivial problem. Communication links, particularly those involving satellites (SatComms), are by no means fail-safe[46]. In addition, processing of that data, which may be somewhat stale by the time it arrives, by the human controller may prove difficult. Many a manned airliner has had the recorded comment from the crew "What is going on" shortly before crashing – this could only be much worse for a remote operator. The ability for the crew to get rapidly vital data, which may be directly indicating some failure or provide appropriate context, is fundamental to safe and efficient operation, particularly for unforeseen events. However, if we provide the UAS with abilities to be versatile in operation as defined above and provide the means to effectively interact with the human controller, then we can start to expand the roles and operations of such vehicles into more complex missions and environments, even if communications have been lost. Such versatility is inherent to autonomous systems as will become clear when this area is explored more fully in Chapter 3.

2.3.2 Cost

Increasingly over the last 50 years, machines have replaced humans. First in manufacturing and subsequently into the service area (automated telling machines, washing machines etc.). In general, the net effect has been reduced cost (with often more performance). The main problem with automation as such, is that it is inflexible. Similarly to current UASs, provided the operation or the environment is simple, then automation can be introduced with subsequent cost benefit. If, as it is proposed, systems can be introduced that can handle complexity, it is highly likely that cost benefits will also be found. The primary

[46] The Civil Aviation Authority have stated that they regard the failure probability of such links to be $1x10^{-3}$ per flight hour.

mechanism for this is reduced (note, not replaced) human involvement. The cost benefits are found via several sources:

- Direct human employment costs such as income/pension, training, deployment, welfare, accommodation, infra-structure etc.
- Indirect human costs such as those caused by strikes, lateness, fitness, inefficiency, rostering, capacity, supervision, performance etc.

Current medium/large UASs are manpower intensive with a typical crew of 3, although there is usually only 1 pilot per vehicle. Experiments have shown that it is possible (under certain constraints) to have I controller operating up to 4 vehicles simultaneously and the effects on costs are estimated in the Figure below:

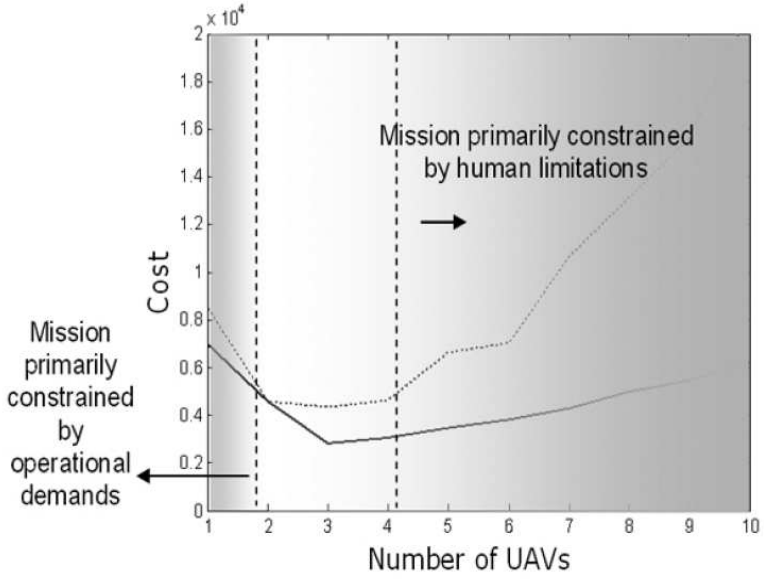


Figure 7: Operator Capacity as a Function of Mission Constraints [16]

The solid line is theoretical performance and dotted line the measured performance. The graph indicates that costs can be reduced by an estimated factor of 2 when operators control several vehicles. The sophistication of the vehicles used in these experiments was very low (comparable to current UASs) and therefore this estimate perhaps should be regarded as the upper-bound for reduced costs with the expectancy of further gains with increased autonomy.

2.3.3 Performance

One metric for human performance as a decision maker is speed. Unfortunately, humans operate at different cognitive levels according to the problem at hand. Where automatic (or reactive) responses are required (driving the car, manually flying the aircraft etc.), typical responses are about 2Hz. For deliberative responses (a higher cognitive level, sometimes known as the Rule Level), the decision – act cycle can be anything from 2 to 10 seconds[47]. For the highest, and hardest, cognitive level, the planning level (also known as the Knowledge Level), decisions may be made in tens of seconds, minutes or even hours; some planning functions are actually beyond human computation (e.g. The Travelling Salesman problem). Cycles of decision making are captured in Boyd's Observation, Orientation, Decision, Action (OODA) loop theory (see Appendix B). He proposed that the ability to progress the OODA "loop" in as short a timescale as possible is essential to success. This theory was extensively explored (and empirical evidence was gained in support) in Ref. [17]. This view is also supported in the following quotations:

> *"One of the most important elements to consider with this battlefield is the potential for UAS to rapidly compress the observe, orient, decide, and act (OODA) loop. Future UAS, able to perceive the situation and act independently with limited or little human input, will greatly shorten decision times. This Perceive - Act line is critical to countering growing adversary UAS threats"* - USAF UAS Roadmap [81]
>
> and also:
>
> *"Autonomy means different things to different people. To [USAF] Information Directorate chief scientist Rich Linderman, it is a matter of speed -- autonomous systems can sense, self-adapt, dynamically plan and make decisions at speeds that preclude having people in the loop. But while the cyber domain*

[47] Examples of human decision timelines will be provided later.

requires decision speeds far beyond human capability, timescales in the aviation domain still allow human interaction, says Dan Thompson, [USAF] Air Vehicles Directorate scientist."
Aviation Week [18].

Such abilities, to make effective decisions at high speed, are also inherent to autonomous systems and this performance benefit can be gained in stressful situations such as:

- When emission control requirements are in effect or when communications are down though failure or jamming
- When operator workload is high, if the operator is required to command/supervise multiple UASs, or if the operator has other tasks to perform such as flying another aircraft.
- Where safety of life during an emergency is an issue.

2.3.4 Safety

Good performance can also be regarded as a lack of failure or a high degree of safety. Unfortunately, again humans are prone to failure as discussed earlier (and will be in detail later). It is a fact that 80% of air accidents are caused by human error[48]. Some of these error modes can be entirely eradicated by careful design of an autonomous system. In general terms, an autonomous system possesses inherent attributes that can contribute to flight safety:

- They do not suffer from memory loss, lack of concentration, or boredom.
- They can react extremely fast to recover situations – very important in time critical situations such as the take-off and landing phases of flight where most manned accidents occur.
- They do not take chances nor do they wilfully violate regulations or dodge safety procedures

[48] It is recognised that humans are also highly effective at preventing accidents caused by triggered events. This is discussed in detail later.

- They can be forced to objectively re-evaluate situations[49] – no human type "confirmation bias" [50].
- They can be forced to attend to a variety of situations virtually simultaneously – no human type "framing bias" or fixation.
- Since humans are not carried, accidents caused by failed life support systems will not occur.
- Loss of control due to human sensory failure, such as disorientation, is absent.
- Automatic take-off and landing modes can reduce accidents due to manual loss of control
- They are under no pressure to return to base or family instead of landing at the most suitable airfield.

2.3.5 General Limitations of Autonomous Systems

Whilst arguments have been provided for increased autonomy for the control of future UASs, it would be unbalanced not to consider some general limitations of autonomy. Again these are expanded in Chapter 4.

Certification All airborne systems have to be certified for use. This in turn requires the appropriate standards[51] to be developed and tested. These standards are not currently in place for airborne autonomous systems and agreements for their development will require consensus across a large community of developers, user-groups and regulators. Even if such standards were in place, then it is currently essential to demonstrate that the system response can be guaranteed in response to specific system inputs. This is

[49] Although having some persistent beliefs can be beneficial, It is normally better practice to destroy all belief functions at the beginning of each computer cycle, typically every 50ms, and then re-assemble them in the next cycle. Such practice assures the evaluation of all evidence which in turn will determine subsequent behaviour and actions.

[50] The Confirmation Bias is the human attribute of clinging onto beliefs or mental models even when contradictory evidence, sometimes overwhelmingly contradictory, is available.

[51] The current accepted standard for the production of airborne software is DO-178B. However this standard assumes that the pilot is on board and it makes no mention of autonomy..

highly likely to remain true for autonomous air systems. Such a system is said to be **deterministic**. It has not been fully researched, but this fundamental requirement for determinism is likely to preclude the use of machine learning techniques where the response of the machine is different according to recent (i.e. since testing) experience. It may be that the requirement for stringent testing places a practical limit on the degree of complexity that can be introduced.

Inappropriate System Responses This inability to learn may provide limitations in the systems' response to issues not foreseen by its designers thus providing a cap to the versatility that potentially is wanted. Note that pilots are allowed to learn and are not constrained to be deterministic, but then we don't certify them because we palpably cannot. However, there is still a lot that can be done with straight-forward rule based systems under the 80/20 rule[52].

Whilst it was proposed that autonomous systems could preclude some accident modes, it is also probably true that some new accident modes may be introduced by the advent of autonomous air systems. A particular example could be what Reason [3] classes as rule based errors[53]. Two modes can be identified. Firstly, an incorrect antecedent belief, which may be generated in a variety of ways, will positively cause an inappropriate action to be performed. Secondly, a need for action may be identified (with hindsight), but no rule was available to generate the appropriate action, either because the designers had not anticipated it, or because no antecedent belief had been formed to fire it. Later Chapters discuss how accidents due to these modes may be reduced.

Human Machine Interaction (HMI) There have been several hundred (nay, thousands) of books, reports and studies into the way humans interact with machines, particularly within aircraft cockpits. These have produced a variety of guidelines into appropriate design methodologies and yet accidents in

[52] A commonly held unwritten rule of systems engineering that states" You can spend 20% of the cost to achieve 80% of the requirements but you will then spend 80% of the cost to achieve the remaining 20%. In other words, if possible, see if 80% of the task will satisfy the overall requirement.

[53] A generalised Rule being: IF (antecedent) BELIEF is TRUE then DO (consequent) ACTION

which the primary cause was faulty HMI, especially with respect to automation, continue to arise.

The best, and kindest, thing we can say about automation is that it is still an incomplete science. If we now start to shift away from automatic machine modes, which can shut the operator "out-of the loop", and progress towards human interaction with decision making machines, we are likely to run into several new and difficult areas when it comes to safety. These are likely to be not just of an engineering nature, but also from social[54], legal and ethical aspects.

For the future, it is likely that new models of HMI and appropriate design guidelines will be required to be developed for autonomous air systems. These problems will be exacerbated by the human's remoteness and consequent lack of immediate SA. Until such systems have been developed, operated and understood in order to provide requisite levels of human trust, they are just not going to be accepted, despite their potential.

2.4 Summary

This Chapter has looked in detail at the concept of UASs, their characteristics and their uses. In particular, the concept of remote operation, and more importantly, all that this implies, was discussed. This has enabled a review of those features of manned and unmanned flight to be made. This concluded that the following differentiators should be considered:

- Employments and Versatility
- Costs

[54] Whilst observing a recent (May 2011) Human Factors synthetic environment experiment with a Sense and Avoid autonomous system, a pilot received a plan proposal from the system to turn right through 30 degrees as a response to an imminent collision. This proposal was promptly rejected by the pilot who then immediately instructed the vehicle to turn right by 30 degrees. When queried, his response was "I am not having a machine tell me what to do!" It was pointed out that his actions had delayed appropriate action by several seconds which in a head on collision scenario could prove fatal. This is an interesting observation from a huge variety of viewpoints and is returned to later.

- Safety
- Communications
- Control topologies, including those aspects required to cater for operator remoteness.

It was argued that, in order to realise the potential of UASs by increasing their versatility, safety and performance whilst at the same time reducing costs, a move away from remote operation, a trend already in progress, has to be made. It was also argued that this trend towards automated operation, in itself, places limitations on the achievable levels. The argument was proposed that a move to autonomous operation would potentially remove this limit on achievable potential. Such an autonomous vehicle would be capable of operating in conjunction with a human controller at whatever level of authority was invested by him **or,** if he was unavailable, at whatever level was appropriate to the situation. This latter requirement, *ipso facto*, places a requirement for the system to be self-aware and aware of its situation. This in turn requires that the system is capable of:

- identifying courses of action (plans) in order to meet the objectives set for it and, from these alternatives, choosing the most suitable.
- proposing the preferred plan for authorisation, if required, and actioning the plan on receipt or by default
- monitoring the progress of this plan and identifying when it has either succeeded or failed

All of the above must be achieved with due regard to the fundamental requirements, previously identified, of Transparency, Equivalence and Safety.

It should now become readily apparent that today's UAS architectures, based heavily on current practice in manned aircraft avionics and cockpits, are not

suitable to achieve all of the above requirements and to the degree necessary. The study now proceeds to propose and analyse such a system architecture.

3 Development of an Advanced Decision Architecture for a UAS

3.1 Autonomous Systems

3.1.1 Definitions

Several definitions of autonomy and autonomous systems are discussed in Appendix A and the following definition is offered from this:

> ***"An autonomous system is one that operates within an environment and is capable of independent decision and action in pursuit of its objectives".***

The concepts of decision and action are intertwined. It is difficult to understand the nature of a decision, or the point in making it, if the decision is not followed by action and in particular, action to change or influence the environment. Therefore we can always think of decisions and actions as a tuple (*D*, *A*). Support for this notion, which is not universally accepted, particularly in the intelligent agent community, is given below.

In Multi-Attribute Decision Theory, a decision is defined as:

> ***"A decision is the commitment to irrevocably allocate resources. A decision is a commitment to act. Action is the irrevocable allocation of resources[55]".***

Similarly, in the Lexicon of Decision Theory published by The Decision Analysis Society[56]:

> **"A decision is an allocation of resources. It can be likened to writing a cheque and delivering it to the payee. It is irrevocable, except that a new decision may reverse it".**

[55] Available at http://en.wiktionary.org/wiki/decision

[56] Decision Analysis Society*: "Lexicon of Decision Making",* at http://faculty.fuqua.duke.edu/daweb/lexicon.htm#decision

As far as the rest of this document is concerned, the above definitions, which are believed to be equivalent to each other, will be used.

3.1.2 Automation and Autonomy

In defining autonomy as above, it should be possible to make a clear distinction between automation and autonomy. At a high level this proves to be the case, however when considering both concepts in detail, especially in relation to a deterministic machine, the distinction becomes somewhat blurred.

A discussion of the definition of automation is also given at Appendix A together with a view on the differences between autonomy and automation. This discussion concludes that an automatic machine is something that does not think, does not decide, nor acts according to an agenda, whereas an autonomous system should be capable of at least one of these, and preferably, all three. It also re-forces the view that autonomous machines should be authorised to make decisions. This may mean real time, operator-in-the-loop control authority, or pre-authorised control, perhaps even hard-coded at the design and implementation stage.

The problem starts when we consider the need for a deterministic system – i.e. one that will produce a consistent single output for a given input. Later, it is argued that in order to produce a certifiable system, then at least the safety related elements must be deterministic. If this case is accepted and the output must be pre-determined given the input, then the resultant system is hardly the nature of what has been defined as autonomous. That is, an independent controller, a deliberative thinker or one that is capable of responding to external situations according to its objectives. In fact, it can be argued that for the latter case, the (so called autonomous and deterministic) machine can, and will, only respond to situations pre-considered by the designer.

This aspect needs more thought and impacts on the question as to whether a machine that learns its behaviour can ever be considered to be certifiable for airborne use.

3.1.3 Intelligence and Autonomy

Is there a relationship between autonomy and intelligence? The answer is yes but they are not the same as each other as is frequently implied, if not stated. A competent autonomous system operating in a complex, threatening and uncertain environment would certainly start to approach what the majority of us would call intelligent behaviour. However, an intelligent being may not be very autonomous at all – a very obedient dog for instance, or my wife's husband! Bacteria, viruses and genes have all the characteristics that we would naturally suggest autonomous systems have – however, they are clearly not intelligent in the accepted sense. For less demanding environments, the need for intelligence or to act intelligently to achieve a stated level of performance reduces, but the system can still be considered highly autonomous if it, rather than someone else, decides what it is to do (and perhaps more importantly, why). Given that there is a difference between intelligence and autonomy, the question remains do we want our autonomous system to be intelligent or at least behave intelligently. For the types of environment that is under consideration, it seems fair to say, at this stage, that some sort of intelligent response is definitely required. It also seems fair to say that the attributes of that intelligence is likely to be different from that of humans, but whilst a human is in the loop, and generally in charge, those attributes of machine intelligence must be able to be aligned with those of humans.

3.1.4 Decision and Action

Although it has been defined that a decision should always be followed by action, it is not necessarily true that action is always preceded by a (conscious) decision. If you stick a pin in some-ones arm, they will try to withdraw it instinctively as a consequence of reaction and not of considered decision. Pain is a sufficient trigger to force an immediate change irrespective of the cause. There is no world model which tries to represent the pin in the arm or assessment of alternative action possibilities. The action is directly "hard-wired" from the perception of pain to the withdrawing of the arm. The action is a stimulus – action pairing, or a tuple (S, A), and can be termed **reactive**.

Conversely, we can say that action preceded by a decision, tuple (D, A), is **deliberative** behaviour. This notion of reactive and deliberative behaviours, and their attributes, is discussed further in Para.3.2.3.

3.1.5 Information

If we accept that a decision is followed by action, and some do not, it is reasonable to ask what comes before the decision. Fortunately there are three, generally accepted, theories which explore this area:

- Classical Decision Theory (CDT),
- Recognition Primed Decision Making (RPDM) Theory
- Boyd's Observation-Orientation–Decision-Action (OODA) cycle.

These are described in more detail at Appendix B.

CDT requires the formulation of alternative Courses Of Action (COAs) which are assessed according to the Decision Maker's (DM) values. The COA that gives the greatest expected utility is chosen (the decision) and then embarked upon (the action). Unfortunately, CDT assumes that the derivation of acceptable alternatives is the starting point in the process but it certainly recognises the key elements of **values** and **objectives**. Objectives come in two major forms, **fundamental** objectives and **means** objectives. The differentiator between the two is in the asking of the question "Why is that an objective". If the answer cannot be cast as another objective, perhaps at a higher level, it is a fundamental objective. If it can, it is a means objective. So we can now infer, from CDT, that the choice of (D, A) is one that ultimately gives the greatest chance of (ultimately) achieving a fundamental objective. This may be represented as tuple (O, D, A).

RPDM is very different from this in that it assumes, and requires, full or sufficient Situational Awareness (SA) of the problem at hand. SA is discussed at Appendix B which proposes the process of achieving SA as a continuing sequence of perception, comprehension and projection (or prediction) i.e. tuple

(Pr, C, Po). So the full recognition primed decision action cycle is tuple ((Pr, C, Po), D, A).

The OODA cycle is very similar. The process is Observation-Orientation–Decision-Action. Observation and Orientation, or “what is going on in the world and how is it relevant to me”, can be seen to be equivalent to the SA process described above. Indeed, the whole OODA cycle has been shown to be equivalent to a more general Information – Decision – Action (IDA) cycle [19] thus implying a tuple (I, D, A).

3.1.6 Nature of A-Systems

In the definition of an autonomous system at Appendix A, the words “operate, action, decide, control” were regarded as generally equivalent. It has already been discussed that, under certain circumstances, decide and action are inseparable. Within the context of the operation of a vehicle, the concepts of control, action and operate have identical meanings. For the context of an autonomous vehicle system, it is proposed that the term control is used throughout. This is useful in that several architectures have been designed for the autonomous control of vehicles such as cars, robots, submarines and even aircraft.

Thus we finally end up with a process, which we can call Information – Decision – Control (IDC), operating over the contexts of objectives, consequences and constraints.

A summary view of the above is given below:

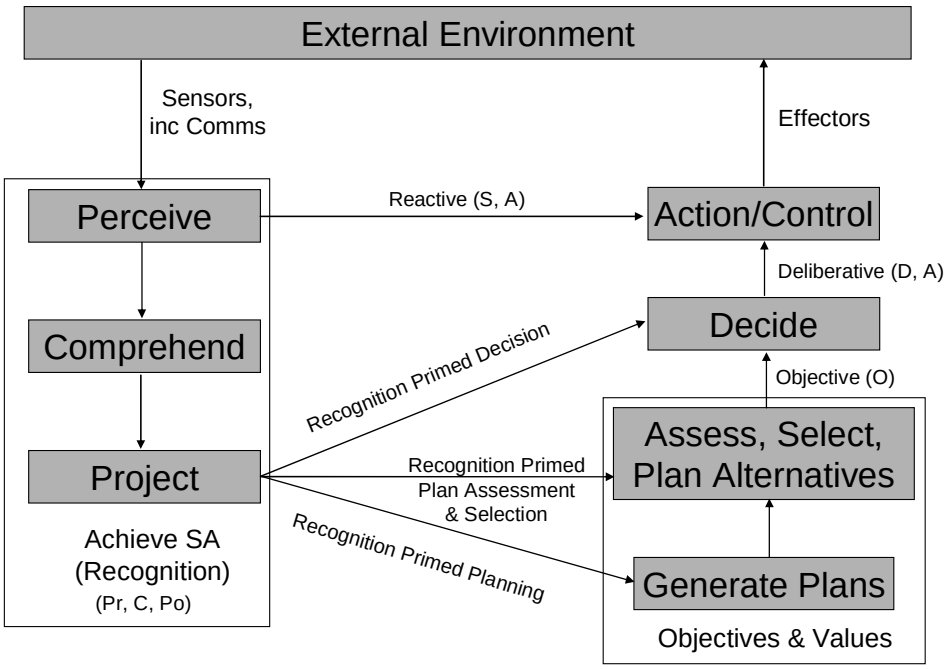


Figure 8: Information – Decision - Control Model

It is believed the above mechanisms for control are a full and complete set with the exception of those directly ordered by an external operator. Therefore, ideally, an architecture for control of a UAS should encompass all the above mechanisms including the latter i.e.:

- Direct Sensor to Effector Reactive Control
- Direct Operator to Effector Control – in effect, a control override
- Deliberative Effector Control using:
    - Recognition Primed Decision Making – this mechanism constitutes a direct decision consequent to the recognition of a particular situation. This mechanism could be effected by a rule based system e.g. IF Under Attack, THEN Turn by 180 degrees, or by a more sophisticated and complex method such

as Case Based Reasoning (CBR) by retrieving a suggested Course of Action from the best fit of input conditions to a case database.

- Operator Decision Making – this can be effected by presenting the Operator with inferenced situational data in order that value judgments can be reached, and accepting his decision as the pre-cursor to a control override.

- Recognition Primed Plan Assessment and Selection – The result of the Situational Assessment is used to draw up a list of applicable plans to that situation and select the most appropriate, either by on the basis of maximum expected utility or by some other mechanism.

- Maximum Expected Utility (Value) Plan Assessment and Selection – the generation of alternative plans to achieve a perceived objective and scoring these plans from value metrics and probability of outcome

- Operator Plan Assessment and Selection – as above but allowing the operator to select the plan

- Recognition Primed Plan Generation – the generation of a plan to achieve an objective within a perceived situation.

- Objective Based Plan Generation – using plans based on applicable objectives; initiating a take-off sequence for instance.

- Operator Commanded Plan Generation – the acceptance of a plan override commanded by the operator.

3.1.7 The Nature of Distributed Control

3.1.7.1 Man Machine Interface

In a manned aircraft, controls for the aircraft are vested in the cockpit primarily, and are under the direct control of the pilot. These controls are the starting point in the generation of a chain of functions that ultimately lead to that control having the desired effect. The last of these functions in the chain is known as an end event function[57]. To generate the required evidence to approve the system as safe, these end-event functions are analysed for their criticality[58]. If this end event function is considered to affect safety adversely, it can be described as a **safety-critical, end-event function**. The complete chain from control function to safety-critical end-event function, routing by whatever path in software or hardware is then termed a **safety critical chain[59].** By a similar process, it is possible to identify **mission critical chains[60].** Since the identification and analysis of these safety critical chains is fundamental to the subsequent certification of the system, consideration of how these would be generated in a UAS is essential.

Controls that would lead to a safety-critical end event function being generated are routinely protected for inadvertent operation. Such protection may include guarding, shielding or covering of switches, or switches that are inoperative without other switches being made. These measures cover hardware originated controls. For areas where the control signal originates in, or passes through, software, the signal is protected by writing the software to a Software Integrity Level (SIL) of 4 or more and often duplicating the hardware in it resides or by providing reversionary paths, which are invoked upon failure detection. Whatever the case, it can be shown that, for manned aircraft, all end-event

[57] An end event function is one that is delivered externally from the system under consideration.

[58] A typical taxonomy of criticality is: non-critical, safety involved and safety critical. The latter implies that a failure of such a function would lead to the loss of the aircraft and/or life.

[59] A chain which, if its integrity is compromised, would lead to an endangerment or loss of life.

[60] A chain which, if its integrity is compromised, would lead to a failure of the mission.

functions are generated by the pilot, even if perhaps remotely. As in the discussion of decision-action model above, it is interesting to speculate on what precedes the control function generation.

The answer lies in the displays and aural warnings[61] that are presented to the pilot by the avionic system and, to a much lesser extent, those perceived by the external audio, tactile and visual scene. A schematic of this, representing the pilot as a Black Box with Unknown Processes (BBUP), is shown below:

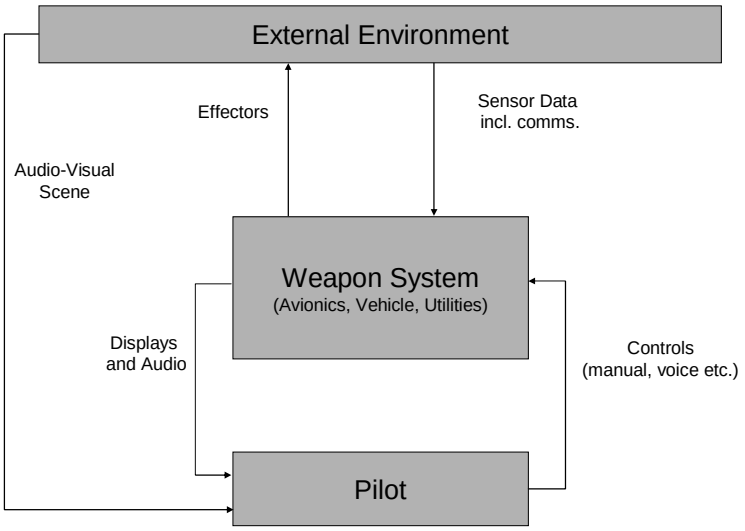


Figure 9: Pilot Vehicle Interface

Although apparently innocuous, the above schematic will give a valuable insight into two major problem areas for the control of a UAS - positive control feedback and determinism, both of which lead to the predictability of UAS behaviour.

Representing the pilot as a BBUP is interesting in itself. However, from a point of view of certification, airworthiness bodies and aircraft manufacturers do not certificate the pilot. It is assumed that he is competent enough not only never to inappropriately invoke a safety critical mode by choice, but also to provide a

[61] I concede that the stall warning “stick-shaker” is a tactile message, but it is probably the only one.

cross check on system integrity. Indeed, many of the FFA reports pass the overall system safety analysis by citing the pilot as being required to monitor certain events. In addition, we do not require, and we cannot specify or test, that the pilot must be deterministic. In fact the reason why we make no attempt to certify the pilot, in formal terms at least, is because of the knowledge that we know he cannot be deterministic in all circumstances. All we can do is specify that the pilot must have adequate training, knowledge and skills to operate the aircraft in question – but that is not the job of the systems design engineer or part of the design process.

So here we have a system, the BBUP, that is not certified, but is required to be operative for the system to pass the certification process. The BBUP therefore represents somewhat of a discontinuity between the receipt of data and the invocation of a control signal, at least in systems engineering terms.

If we now replace the BBUP with an autonomous decision and control system, such as below, we have an entirely different result, although it apparently looks the same:

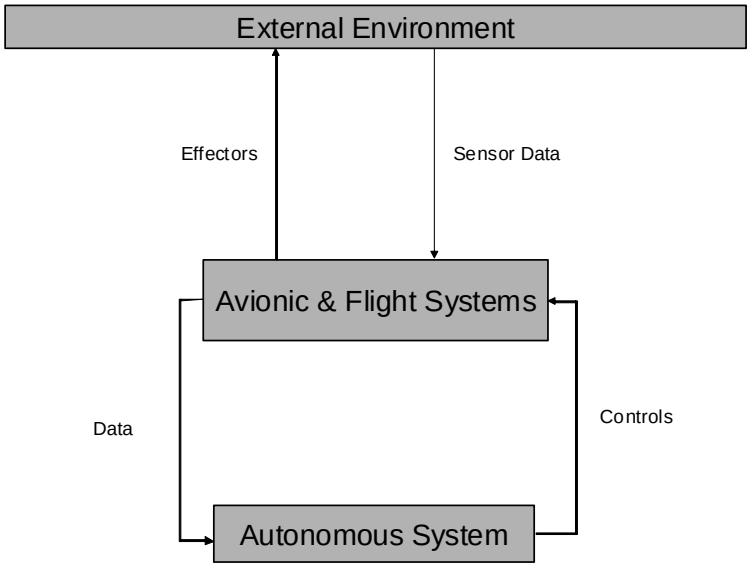


Figure 10: Autonomous System Control – Vehicle Interface

The provision of an autonomous system cannot be regarded as equivalent to a pilot and the insertion of it as a replacement changes things as follows:

- It can be, and indeed must be, certifiable This requires it to be:
    - Deterministic. For every possible state of data input, there must a reliable and repeatable output. The system will be, by design requirement, predictable.
    - Competent or “Fit for Purpose”. Therefore it must have a range of behaviours and control functions that will allow it to carry out its assigned tasks. Furthermore these functions must have been proven to be able to do so.
    - Capable of handling safety critical functions.
- Given the nature of machines and humans and their differences – humans are inherently good at Situational Awareness, whilst machines are not - the requirement for an intelligent and knowledgeable cross-monitor to achieve a safe system output, as normally performed by the pilot, now becomes a real problem to satisfy. It is possible for another, independent, system to perform this cross monitoring role but it likely to suffer from the same problems that would lead to an unsafe output in the first place. Namely, a lack of knowledge, understanding and, dare I say, intelligence.
- There is now no element of discontinuity. There is a seamless transition, and indeed translation, from data to control. This aspect allows for the possibility of unwanted positive control feedback. As we have discussed, the system is deterministic and a given data input set will produce a guaranteed output. Consider an external unit that just happens to know the process, and data, by which that output is produced, and of course such an output may be safety critical. All that the external unit needs to do is to provide that data, perhaps by deliberate disinformation or deception, in the correct form and context, and an inappropriate control

signal is generated and the end event function is passed to the environment. This signal or effector may be detected by the malpractor, the originating data reinforced and subsequently returned to the system. A process known to control system engineers as positive feedback.

Some of these problems may be overcome by returning the operator to the loop to fulfil some of the control functions. The problem then swaps to a consideration of how control passes between the two control systems, human and autonomous, and the integrity and availability of their communication. This is discussed further.

3.1.7.2 Distributed UAS Control

If we consider a distribution of control between the autonomous system and the external human controller, and assume that the human has executive, or overall, control, then a system of appropriate and dynamic authorisation of control can be envisaged. One such system is described in the schematic below:

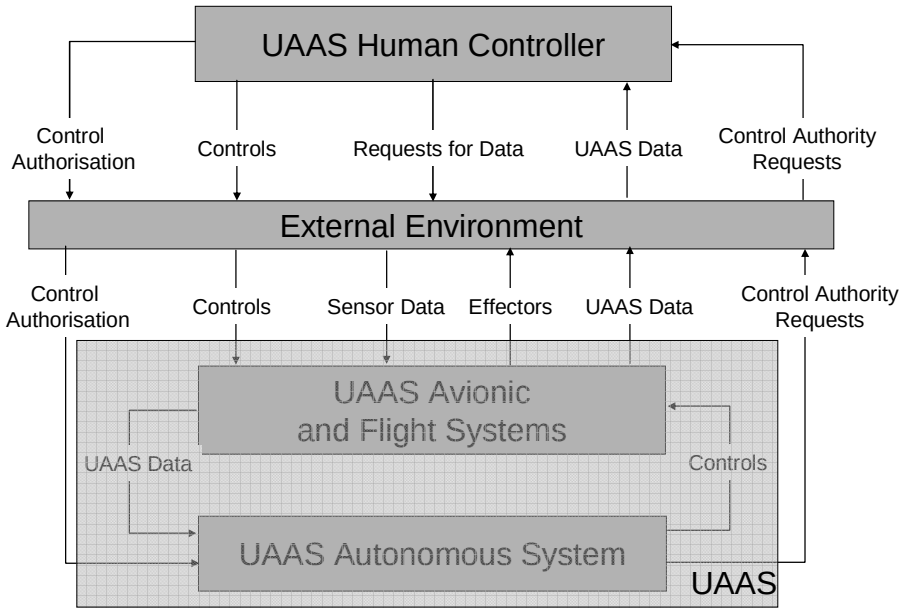


Figure 11: Human - Autonomous System Interface

This concept of distribution of control between the autonomous system and the remote human controller is fundamental to the architectural design. The concept calls for a taxonomy of control hierarchies whereby control of a specific functions is allocated a control protocol and an associated authorisation to proceed.

3.1.8 Level of Automation and Autonomy

There has been much work done on developing such a taxonomy. The ground breaking work in this area is generally acknowledged to be by Sheridan & Verplank in 1978 [20]. Their work, together with that of Parasuraman, Sheridan, and Wickens in 2000 [21] is described in the table below, reproduced from [22].

Automation Level	Automation Description
1	The computer offers no assistance: human must take all decision and actions.
2	The computer offers a complete set of decision/action alternatives, or
3	narrows the selection down to a few, or
4	suggests one alternative, and
5	executes that suggestion if the human approves, or
6	allows the human a restricted time to veto before automatic execution, or
7	executes automatically, then necessarily informs humans, and
8	informs the human only if asked, or
9	informs the human only if it, the computer, decides to.
10	The computer decides everything and acts autonomously, ignoring the human.

Table 5: Levels of Automation of Decision and Action

3.1.8.1 PACT Levels

The levels shown above have been further modified by Taylor [23] and are called the Pilot Authorisation and Control of Tasks (PACT). These original PACT levels originated from consideration of work conducted separately by the (US) Air Force Research Laboratory (AFRL) and Sheridan and Verplank. The PACT levels so discussed have been further modified by BAE Systems Human Factor engineers and their results are given at [24]. These modified levels,

together with the original levels they were derived from and other influences are described below:

AFRL Original	PACT Locus of Authority	Computer Autonomy	PACT Level	Sheridan & Verplank Levels of HMI
Fully Autonomous	Computer Monitored by pilot	Full	**5b**	Computer does everything autonomously
			5a	Computer chooses action, performs it & informs human
Remotely Supervised	Computer backed up by pilot	Action unless revoked	**4b**	Computer chooses action & performs it unless human disapproves
		Action if authorised	**4a**	Computer chooses action & performs it if human approves
Remotely Operated	Pilot backed up by computer	Advice, and if authorised, action	**3**	Computer suggests options and proposes one of them
	Pilot assisted by computer	Advice	**2**	Computer suggests options to human
	Pilot assisted by computer only when requested	Advice only if requested	**1**	Human asks computer to suggest options and human selects
Remotely Piloted	Pilot	None	**0**	Whole task done by human except for actual operation

Table 6: Modified PACT Levels

It can be seen that, for the type of distributed control problem we envisage, the human will delegate authority to the autonomous decision making system in the UAS using PACT Levels (PLs) 4-5. This implies that, for each function, at either the control or plan level, the UAS needs to have an understanding of what level of authorisation it is operating at. A way of doing this is to maintain an authorisation status list for each control function. When a control function is selected to be invoked, by whatever process, reactive or deliberative, the authorisation status list is checked and the appropriate response generated. This functionality is discussed further in Para.3.3.3.

This concept can be taken further and the PLs could be used for delegating authority to the management functions of sub-systems lower in the command hierarchy to the decision making system.

3.2 Decision Architectures for Air Systems

3.2.1 Architecture Definition

As with “autonomy” and “agents”, there is no unanimous agreement on the definition of “architecture” as it relates to a software/hardware system. The Institute of Electrical and Electronics Engineers (IEEE) cites [25]:

> **“The structure of components, their relationships, and the principles and guidelines governing their design and evolution over time”**

An excellent reference covering virtually all known definitions, both classical and modern, is given by Carnegie Mellon University, Software Engineering Institute[62]. These authors appear to prefer a modern definition emanating from [26] which states:

> **“The software architecture of a program or computing system is the structure or structures of the system, which comprise**

[62] Carnegie Mellon University, Software Engineering Institute: *“Published Software Architecture Definitions”*, available at http://www.sei.cmu.edu/architecture/published_definitions.html

software elements, the externally visible properties of those elements, and the relationships among them".

At [26], the implications of the above definition are discussed in detail. These are summarised below:

- An architecture is an abstraction of the system that suppresses internal details
- Systems can, and do, comprise more than one structure
- Every system has an architecture because every system can be shown to be composed of elements. However, this does not mean that the architecture is necessarily known.
- The behaviour of each element is part of the architecture
- The definition is indifferent as to whether an architecture is a good or bad one. This raises the question of architecture evaluation. In other words, what constitutes a "good" architecture?

3.2.2 Architectural Viewpoints

From the generalised definition of a software architecture given above, it is clear that differing viewpoints of the same architecture can be stated, not only from different levels of abstraction (i.e. detail), but also from different properties of the composing elements and their relationships.

3.2.2.1 Reference Architecture

A Reference Architecture describes the highest level of abstraction, free from implementation details, but encompassing definitions of relationships, not only between top level elements within the system of interest, but also those interacting elements external to it.

3.2.2.2 Functional architecture

In contrast to a Reference Architecture, a Functional Architecture is solely concerned with what the system of interest actually does. It describes the functionality of the system and specifies the interfaces between the interacting

elements, at increasing levels of detail for each aspect. A top level or system level viewpoint expresses general functionality and indicates the lower levels of system partitioning. These sub-systems' functionality are further detailed as required. At the highest level of detail, each algorithm fulfilling the functionality, together with the necessary input/output, is explicitly described and allocated to the computing hardware that will undertake them. This latter process is sometimes called the Functional Allocation.

3.2.3 Robotic Architectures

Since there has been a vast amount of research in the field of robotic architectures, which by nature are architectures designed for control, and usually for control of a vehicle, it is worth a review of the general nature of these.

Earlier it was conjectured that that a process termed IDC would be useful for portraying the required control loop for an autonomous system. This is uncomfortably similar to the now out-dated (at least since 1985) notion of **Sense-Plan-Act** (SPA) architectures. These had three primitive functional elements: a sensing system, a planning system and an execution system. It is valuable to reflect on the history of robotic architectures and the demise of SPA in order that old lessons may be reviewed and/or re-learned.

3.2.4 BACKGROUND

The nature of SPA architectures is that they are characterised by a one way flow of control from sensors to effectors via a plan which was similar to a computer program – a series of elements executed using traditional programming control flows such a conditionals, loops and orderings. The execution of such a plan is conventional and therefore simple. The interesting, and therefore difficult, part of producing an effective SPA was seen to be the quality and sophistication of its world modelling and planning. This proved to be a non-trivial problem.

The first departure from SPA was Rodney Brookes' **Subsumption** architecture. This renounced the need for a world model (the information processing

element) and the planning system and replaced them with a series of direct reactive responses which connect the sensed input and provide a ”hard-wired” output. These could be hierarchical with one response link subsumed by a higher level one. Thus the nature of Subsumption is hierarchical reactive control which is, by nature, very fast. This fact enabled robots using Subsumption to outperform those using SPA enormously. Unfortunately, and this is a key fact which will be returned to, the degree of sophistication is governed by the number of links and the quality of the Subsumption design. In short, the architecture is not directly scalable. Gat [27] states that one possible cause of Subsumption’s apparent “capability ceiling” is that the architecture lacks mechanisms for managing complexity. He quotes Hartley [28] who reflects that Subsumption is not sufficiently modular and that higher layers interfere with lower layers and cannot be designed independently of each other. Gat himself designed a modification to the Subsumption architecture whereby instead of suppressing lower layers, higher layers provided input or advice to the lower layers. There were manifested in the Tooth and Rocky III robots. However, the robots were single task only and could not achieve different tasks without rewriting the control program. This is probably true of all such reactive architectures.

So, reactive architectures replaced the SPA architecture based robots by essentially replacing all (subsumption type), or mostly all, of the world modelling and planning aspects. In doing so, robots could be built which were satisfyingly fast and performed, in the main, pretty well - at least for unsophisticated and singular tasks. However, as computers and programming languages improved, it became more attractive to introduce a modicum of planning and world modelling, especially where speed of operation was not demanding. This bred a new architectural approach commonly called the **Hybrid Architecture** where a reactive layer was combined with a deliberative layer. An example of this was the **Guardian** architecture of Barbara Hayes-Roth which was one of the first to attempt to control complex tasks.

Eventually, a number of researchers[63], working independently, but with a common background of research material, arrived at solutions surprisingly similar. An abstraction of these solutions is now commonly known as the **Three Layer Architecture (TLA)**. Such architectures have the following components [27] which may operate asynchronously:

- A fast reactive feedback control mechanism, sometimes called the **skill layer** or **controller**
- A slower deliberative planning layer, sometimes called the **planner** or **deliberator**
- A mechanism that connects the first two, sometimes called the **sequencer, executive or manager.**

Since the three layer architecture has almost become an industry standard (for robots at least) it is important to see where they are different from the SPA architecture and perhaps incorporate their positive aspects into any proposed new architecture for a UAS.

3.2.5 The Three Layer Architecture

3.2.5.1 The Controller

The control flow in an SPA architecture is uni-directional. By contrast, in a TLA there is reactive feedback. This is required for several reasons:

- The controller should know the current state of its actuator and operate it to achieve the desired state over the period of time allocated to it. This may be a continuous function (for something akin to a control surface) or a time-bound state transition function for a finite state machine.

- The controller should know when it has failed (to achieve the required state of its actuator)

[63] Gat, Bonasso and Connell. The difference was in the type of sequencer used. Gat used Firby's Reactive Action Package (RAPs), Bonasso used Kailbeiling's REX/GAPPS system initially and RAPs later, and Connell's was based on Subsumption.

- The controller should be able to report that failed state to a higher level for contingency, alternative or remedial action.

The above implies that the controller need know nothing of anything else save the state of its actuator and the control requirement (or order) from the higher level.

Restricting access to information within a system in general is desirable for many reasons:

- It makes interfaces between modules simpler and therefore less likely to be specified or implemented in error.
- It conforms to the general principles of Occam's Razor [29]
- It allows for an easier analysis and assessment of mission and safety critical functions for certification.
- It simplifies any hardware or software connections.
- It may reduce the data required at some connections which in turn allows for higher speed data transmissions.

Controller Requirements

If we implement a generic mechanism for a controller such as that shown below, we shall achieve the above requirements and bring the design into line with the general required characteristics of a TLA:

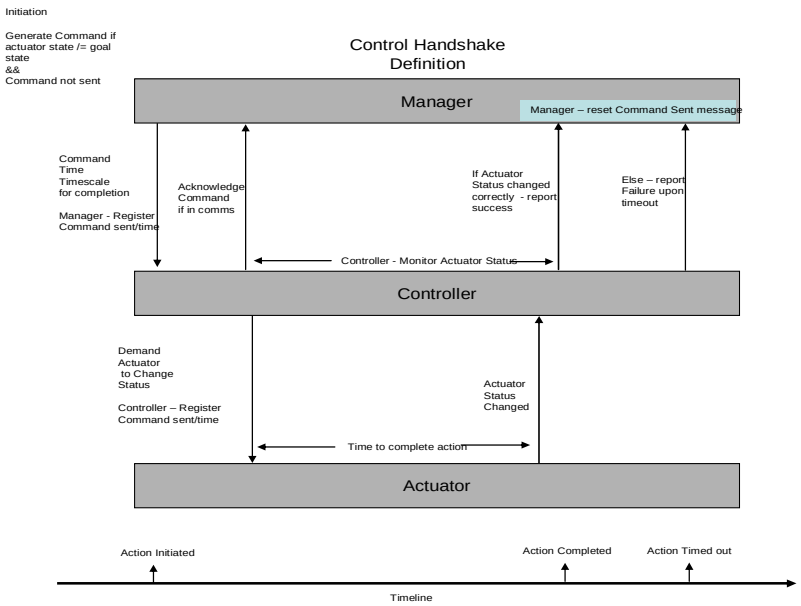


Figure 12: Generic Sequencer/Controller Handshake Protocol

3.2.5.2 The Sequencer

The role of the sequencer is to connect the deliberation layer to the control layer. Specifically, the sequencer is to:

- Assess and if necessary select plans generated by the deliberator
- Request plans, or modifications to plans, from the deliberator.
- To implement selected plans and sequence them to the controller.
- To react to external and internal, events and situations
- Monitor progress of plans
- Decide when a plan is no longer relevant or is inadequate to achieve the objectives

TLA requires that the sequencer be a reactive plan execution mechanism [27]. This is taken to mean that a plan sequence of primitive actions does not proceed to the next element of the plan until the controller has reported success

with the current element. The above generic sequencer/controller handshake protocol embraces this requirement. It is worth considering that non-linear plan sequences are now being developed. These can still fit with the original requirement for plan sequencing

Sequencer Requirements

If we accept the handshake protocol in Figure 12, then the proposed sequencer will be in accord with the general principles of TLAs.

3.2.5.3 The Deliberator

The deliberator or planner has the task of generating plans for the sequencer to assess, choose and implement, and to respond to the sequencer for specific queries or requests. For a UAS, both tasks are likely to be required. An example may be a route generation to avoid certain no-go areas. The route plan will be generated using constraints such as fuel restrictions, avoidance margins, etc., perhaps using default ones initially. Once the route plan is passed to the sequencer and assessed for suitability, it may be either accepted or returned for modification/optimisation by changes to the constraints.

Deliberator Requirements

There appear to be no extra requirements or characteristics required for the deliberator suggested by TLAs.

3.2.5.4 Sensing and the World Model

TLAs do not appear to consider the generation and maintenance of the world, or world model, as a separate layer but clearly they act upon its present and predicted future states. Gat [27] states that stateless sensor based algorithms inhabit the control layer, algorithms based on past states are in the sequencer layer, whilst those that make predictions about the future inhabit the deliberator.

The type of internal state for the controller has already been remarked upon. It should only consist of present knowledge of the effector that it is controlling.

This knowledge should include, not only the state of the effector, but also its status and this may be generated either from:

- the effector itself as part of the feedback process, or
- perceived by the controller from a consideration of the performance of the effector, or
- be independently generated from a separate monitor of the effector.

In general, for most robotic architectures, the TLA serves well. However, in practical terms, the world modelling required for robot applications is usually simple. Given the background of subsumption, where a direct sensor to actuator link replaced the world model, this is not surprising. For many TLA robots, with tasks not very much more demanding than those using subsumption, a simple representation is all that is required. This is highly unlikely to be true for a UAS given the nature of its environment and the degree of competency required. If this proves to be the case, the architecture for a UAS must make specific provision for an information processing layer.

3.2.5.5 Suitability

Many robotic architectures have been designed and implemented. The most successful have been highly specialised applications either in academia or in manufacturing. Such systems tend to have low levels of functionality and world modelling. However, robotic architectures are well developed, have a strong theoretical underpinning and can clearly point the way to an architecture suitable for an autonomous airborne application.

3.2.6 Architecture Proposal

An architecture is proposed based on the following bases:

- By accepting those modifications above as suggested by consideration of TLAs, and
- Including an information collection and processing layer, and
- Make clear provision for mission and safety critical control functions, and

- By making provision for the human control of the system by authorisation in accordance with a defined protocol (such as the PACT levels)

3.2.7 Avionic System Architectures

3.2.7.1 The Evolution of Avionic Systems

The term “Avionics” originated from an amalgam of “Aviation Electronics” sometime around the mid 1940’s. In those days, avionics were standalone items complete with their own power supplies and controls, and sometimes displays and antennas. The equipment was heavy but sparse and for more functionality, large aircraft were required. The weight problem eased somewhat with the advent of transistors and, eventually, digital computers but still most avionics provided a bespoke functionality within Line Replaceable Units (LRUs). Such systems are known as **federated systems** and their overall design is referred to as a **federated architecture**. Given that some redundancy was required, such avionic systems were heavy, inefficient and expensive.

By the late 1970’s, It was clear that by providing communication links between LRUs would in itself enable utilisation of common resources such as computing, power supplies and displays thus saving weight, increasing efficiency and reliability. Such systems were known as **integrated systems**. However, the systems were still bespoke and the communications between them were un-standardised at the application layer. This meant that upgrades were expensive and the effect of changing one LRU would almost certainly impact on other LRUs. These systems are often referred to as **closed systems**.

The major change came with the advent of standardised interfaces and protocols, a notable one being the US MIL-STD-1553B standard. Adoption of this standard throughout an avionic system enabled sharing, fusing and broadcasting of data at high rates (in those days up to 1Mbit/sec, nowadays at nearly 2GBit/sec). Avionic systems which utilise open standards exclusively are often known as **open systems** with **open architectures**.

A good example of an integrated and (predominantly but not completely) open avionic system using such interfaces is the Eurofighter Typhoon where the communications are distributed using "twisted pair"[64] wiring and which form the 1553B databuses. In Typhoon, the system architecture is functionally partitioned into three major sub-systems; avionics, utilities and flight control. The avionics sub-system is further functionally partitioned into seven sub-sub-systems each with its own computing centre. Such a system is referred to as a **distributed architecture.**

3.2.7.2 Current State

However, the Typhoon avionic architecture design dates from the early 1990s when computing had severe limitations compared to today. Many issues [30] needed to be addressed such as:

- Affordability of systems throughout the life cycle, i.e. reduction of both support costs and acquisition costs
- System complexity, resulting from increasing functionality, performance and integration
- Obsolescence management, exacerbated by the increase of system lifetimes and the decrease of component lifetimes
- System upgrades:
    - difficulty in providing enhanced capability
    - evolution of the initial system into multiple variants and for multiple customers
    - extent of system re-qualification following upgrades
- Improved system availability, typically by means of an extended Maintenance-Free Operating Period (MFOP)
- System flexibility, i.e. the ability to easily change the functionality of the system to adapt to different operational scenarios, roles or mission

[64] Twisted pair cabling is a common form of wiring in which two conductors are wound around each other for the purposes of cancelling out electromagnetic interference which can cause crosstalk. [Wikipedia]. Also it should be noted that Typhoon also uses fibre-optic links.

modes.

These issues are addressed to a greater or lesser extent by the development of **Integrated Modular Avionics (IMAs);** a trend which commenced in the early 1990s and continues to date. Such architectures frequently have **centralised** or **core** computing hardware to further drive down weight and space restrictions and reduce the cooling and power needs. The development of the VersaModule Eurocard (VME) rack is a good example of this. Together with the hardware development, a major driver to reduce life cycle costs and obsolescence is the development of a modular software architecture in which the application software is made independent of the hardware by layering of software functionality. The BAE Systems proprietary 3-layer stack and Smith Industries Common Operating Environment are good examples of this.

In summary, avionic systems and their architectures have evolved to become powerful integrated systems delivering increasing functionality but at constantly reducing life cycle cost. There is every reason to believe that this trend will continue.

3.2.7.3 General Requirements for Avionic Systems

The very nature of their environment forces avionic systems to have certain general characteristics compared to other electronic systems. Pre-eminent of these is that avionic must be provably safe.

3.2.7.4 Safety

3.2.7.4.1 Design and Assessment

Avionic systems, in fact all airborne systems excluding, perhaps, missiles, must have a high tolerance to failure and when they do fail, they generally must fail safe. The proof of a safe system resides in its certification by airworthiness bodies, which are usually independent of, but work in concert with, the system design authority. To achieve certification, systems must be designed and assessed for safety and the system tested to ensure safe system responses. In general, there are three phases to developing safe or high integrity systems*31*:

- Development and Integration
- Verification and Validation
- Integral Activities

During Development and Integration a Preliminary Hazard Analysis (PHA) is produced which will analyse the system functional design. For a functional assessment, the initial step is to assess each end-event function[65] for its criticality[66]. These are then assessed by Functional Failure Analysis (FFA) which considers three failure modes:

- Failure to provide a function when it is required
- The provision of a function when it is not required
- The provision of a function incorrectly.

The effect of the failure is qualitatively assessed and the severity of the failure can be categorised as, say, Catastrophic, Critical, Marginal or Negligible. The assessment is then extended to include failures in conjunction with other failures or situations. Fault trees are then composed to determine how the failure can be generated. Finally, safety critical chains are established. Once these are understood, the design can be modified, procedures put in place, or redundancy built in to reduce the probability of a system failure occurring to an acceptable level.

Appropriate flight worthy code is produced to requisite software standards using approved procedures, languages and validated compliers. Use of the latter may in itself force certain procedures to be followed.

During Validation and Verification, the system response is tested to ensure that the requirements, including those that are safety related, are correctly implemented.

[65] An end event function is one that is delivered externally from the system under consideration.

[66] A typical taxonomy of criticality is: non-critical, safety involved and safety critical. The latter implies that a failure of such a function would lead to the loss of the aircraft and/or life.

Integral Activities covers the use of appropriate processes such as configuration management, version control, documentation and traceability[30]

3.2.7.4.2 Standards

Certain safety standards will be imposed on the system designers. An example is the set of standards developed by ASAAC for Integrated Modular Systems. Another set is the standards and procedures required for the development of software compliant to the European Standard EN50128. This Standard defines Software Integrity using a six-level scale (zero to five). Airborne software is usually produced to Software Integrity Level (SIL) 3, whilst that deemed to be safety critical will be produced to SIL4.

3.2.7.4.3 Robustness

An avionic system must be robust. Unlike a PC it cannot crash, stop and reassess, require operator input etc. These requirements are at the heart of hardware /software integration and high integrity operating systems, running in real time, are vital.

3.2.7.4.4 Mission Orientated

Airborne equipments are in general expensive to operate and the functionality which gives it its ability to perform its mission is expensive to produce. Thus, like the concept of Safety Criticality, the concept of Mission Criticality has developed. The approach to achieving mission success has closely followed the safety approach and the system is analysed to identify those components whose failure will cause a mission failure. The system can then be modified to introduce redundancy and high integrity to those areas.

3.2.7.5 UAS Specific Requirements

Development of avionic systems for UAS has, not surprisingly, generally followed the evolution described above with a few additions and modifications described below.

Volume and Weight The requirements for low volume, cost and power together with high reliability and robustness are probably more in demand for UAS than for manned aircraft.

Transparency Transparency is generally understood to be the ability to operate such that, to an external viewer (such as ATC), the vehicle's appears to be identical (transparent) with other air users. In order to operate in non-segregated airspace, a UAS must:

- be fitted with the equipment specification applicable to the class of airspace it intends to operate in
- comply with ATC instructions using the same infrastructure as manned aircraft
- operate with no adverse effect on other air users
- not require special or extra services

Equivalence Equivalence is generally regarded as the adherence to operational regulations in exactly the same way as other air users. In order to operate in non-segregated airspace, a UAS must:

- be capable of complying with all existing operational procedures, rules and regulations
- be certified to operate which includes:
    - The vehicle, including the systems on board the vehicle
    - The operator, pilot or commander (licensing)
    - The off board systems used by the operators
    - The communications infrastructure linking them and other users
- have functionality that the human pilot inherently provides e.g. lookout

Safety The vehicle's operation must be no less safe than manned aviation.

In order to be demonstrably safe as a manned aircraft, a UAS must:

- have a reliability level as good as manned aircraft

- be able to respond effectively to unforeseen events:
    - emergencies, equipment or systems failures
    - safe diversionary routing/landing/crashing
    - avoid ground/air collisions
    - communications loss
- not present hazards to other air users or personnel on the ground
- have high integrity and security of its systems

Current accident rates (accidents per 100,000 hours) are*:

- Large Airliners ~ 0.01
- Regional Commuter Airliners ~ 0.1
- General Aviation ~ 6.5
- UASs e.g.:
    - Predator ~ 18
    - Global Hawk ~ 80
    - RPVs ~ 150

In addition, if the system has autonomous functionality incorporated to any degree, then the system is likely to have higher than normal safety critical functions. This view is supported in the recent Defence Industrial Strategy document issued by the UK MoD [32].

3.2.7.6 Summary of Requirements

In general, the general requirements for an avionic system are:

- To be capable of certification for airworthiness by design. This turn requires:
    - It to be capable of being assessed and tested for safety.
    - It to have repeatable, deterministic and safe responses to external and internal input.
    - The use of robust and rugged hardware running high integrity

software in real time.

- Adequate redundancy for safety and mission critical components and processes.
- Handling of safety-critical functions which must fail safe and be incapable of operating when not required.

- To have low volume, cost and power needs.
- To conform to appropriate standards
- To be highly integrated, modular and efficient
- To be secure

3.3 Proposed Decision and Support Architecture

3.3.1 Cardinal System Requirements and Characteristics

The example systems reviewed in the previous section are a mixture of components for making decisions through to complete integrated vehicle, avionic and decision sub-systems. In deriving a cardinal requirement specification, it is difficult at this stage to separate out those components responsible for making decisions, those for providing the information with which to do so and those that will generate and implement appropriate sub-plans. The requirements below do not specify the implementation or the partitioning of the system into appropriate sub-systems. This aspect will be addressed when considering the system design.

Based on the previous discussions, the following cardinal system characteristics and requirements are proposed:

1. **Decision Making** - The system should be capable of making decisions, preferably using all of the methods described in Para. 3.1.6. This in turn requires:
   - Direct Sensor to Effector Reactive Control – a sensor invoked reactive control function
   - Direct Operator to Effector Control – operator control override function

- Deliberative Effector Control:
    - Recognition Primed Decision Making – a situational state recognition function
    - Operator Decision Making -
    - Recognition Primed Plan Assessment and Selection – a plan/situation association table
    - Maximum Expected Utility (Value) Plan Assessment and Selection – a plan value generation algorithm
    - Operator Plan Assessment and Selection – operator plan override function
    - Recognition Primed Plan Generation.
    - Objective Based Plan Generation -
    - Operator Commanded Plan Generation – operator commands the system to generate a suitable plan.

2. **Certification** – The system must be capable of handling safety related functions and must be robust for the air environment. If it is accepted that the system must follow the general requirements of an avionic system, these in turn require the following as discussed in Para. 3.2.7.6:
    - To be capable of being assessed and tested for safety.
    - To have repeatable, deterministic and safe responses to external and internal input.
    - To have robust and rugged hardware running high integrity software in real time.
    - To have adequate redundancy for safety and mission critical components and processes.
    - To handle safety-critical functions which must fail safe and be incapable of operating when not required.
3. **Competency** – The system must be able to conduct missions, handle failures and other unforeseen events and respond appropriately to produce

successful outcomes where possible[67].

In order to conduct missions effectively, it must:

- Effectively communicate with, respond to, and comply with, Air Traffic Control and System Operator authorities.
- For operation within UK Airspace, comply with the CAA Air Navigation Order (which includes the Rules of the Air), CAP 722[68] and other air regulations and mandatory procedures. A key requirement of CAP 722 is the requirement for a "Sense and Avoid" capability which is the equivalent to the requirement for a human pilot to "See and Avoid" other air vehicles.
- Implement standing procedures for each phase of flight.
- Constantly monitor, assess and update, where necessary, its mission (including route) and fuel plans to ensure safe arrival at its destination. This must be achieved whilst avoiding no go areas and threats such as thunderstorms, etc.

In order to handle failures successfully, it must:

- Detect and diagnose system vehicle failures, and
- Generate, assess, select and undertake the most appropriate remedial action and monitor progress.
- Determine the best course of action following system failures and initiate action, showing a high standard of airmanship (if possible).

In order to respond to unforeseen events and situations and achieve successful outcomes, it must:

- Identify events and situations

[67] It would be unfair to ask it to cope with some major failures such as loss of directional control, operation outside of the air envelope or airframe failure. However, it would be fair to ask it to cope with an engine failure and carry out remedial action, even if that ultimately entailed a controlled crash landing.

[68] CAA Publication detailing the regulations and requirements to be observed for the operation of UASs in UK Airspace

- Predict and the estimate the consequences
- Asses and predict the probability of success of alternative courses of action (COA) in response and select the most appropriate. In doing so, it:
  - Must act in accordance with its objectives
  - Must not rapidly or inappropriately “flip-flop” between alternatives but only commit to a particular COA when it is clearly superior or if circumstances dictate the need for urgent short term action.

The more complex situations the system can deal with, the more it will be judged to be competent and the less need it will have for operator override of control. Ideally, it should be able to recognise situations for which it can offer no solution and to report that to the human operator. In the absence of communications with the human operator, it should have reversionary modes of control which should in general fail-safely.

4. **Authorised Control** – The system must be capable of determining whether it has the authority for committing a decision into action. This authority can be likened to a meta-control under the jurisdiction of the human operator. The above implies the following authorisation functionality:
   - For each control function, there must be an assigned PACT Level (PL)
   - Before invoking a control function, the requirements of the associated PL must be checked and if below PL5, satisfied by requesting authorisation (PL4a) or waiting for authorisation timeout (PL4b). If the associated PL is PL5a, the operator must be informed after the control function has completed .
   - Assigned PACT levels can only be changed by the operator, either directly (on command) or indirectly (according to an operator defined rule-set).

3.3.2 Proposed Architecture

3.3.2.1 System Partitions

In Para. 3.2.6, a 4-layer architecture was envisaged: a conventional TLA, together with an additional information processing layer. Since the system is also likely to be more complex than that of robotic architectures and is likely to therefore consist of separate computing centres, a functional breakdown into related areas, or system functional partitioning, is likely to be beneficial. Since it is reasonable to consider the decision making sub-system as part of the overall avionic system, and taking into account the requirements in the previous section along, the following partitions are proposed:

- A Mission Master Executive – responsible for all cross partition plans, decisions and actions. To act, in conjunction with the human control element, as the surrogate pilot controller.
- An Information System – responsible for:
    - The management, processing and dispersement of external data to provide the necessary information for Situational Awareness.
    - The collection of internal data from contributing sub-systems.
    - The retrieval and storage of data from/to databases
    - Flight data recording
- A Vehicle System – responsible for all non-avionic sub-systems such as hydraulics, electrical power and airframe systems (undercarriage, flaps, brakes etc.).
- A Flight Management System – responsible for the directional control of the aircraft and operation of the engines.
- A Navigation System – responsible for all navigational aspects including the flight (including route) plan and fuel plan.
- A Communications System – responsible for all communications plans and actions
- An Air Safety System – responsible for the recognition and avoidance of

threats such as weather cells and other air traffic

- System Health System – responsible for monitoring and reporting system health by diagnosing failures, assessing the impact of those failures on the mission and proposing plans for remedial action.
- Sensor Suite – responsible for operation of all sensors.

3.3.2.2 General Structure

The general outline of the architecture follows that of an SPA architecture with those modifications suggested by a consideration of the TLA described in Para. 3.2.3. I.e. it should have:

- A planning layer
- A sequencing, or management layer
- A re-active feedback control layer

In addition, provision of a separate information layer is implemented, partly due to the complexity of the world modelling that is necessary, and partly as an aid to simplifying the hazard analysis that will be required as part of the certification process.

3.3.2.3 Design Aims

In Para. 3.2.5.1, it was also considered that restriction of data throughout the architecture was an aid to analysis of the system design. This is also likely to be true for plans and controls. The following design aims, comparable to general principles of Command and Control, are proposed:

- Information should be encapsulated where possible and restricted to those elements which have a direct need for it.
- Plans that have no impact on other functional elements should also be encapsulated within their own sphere, or partition. On the other hand, plans which do not, should be referred upwards to a higher authority. This implies a hierarchy of authority within the overall system.
- Controls should be managed at the lowest level possible and controllers

should only have access to the state of the actuator they are responsible for controlling.

- Separation between the information layer and management layer (which is the sole means of generating behaviour) will enable easier identification of safety critical chains.

From these principles, a generic view of the relationship between the four layers and their higher authority, which is called the Master Executive, can be envisioned as follows:

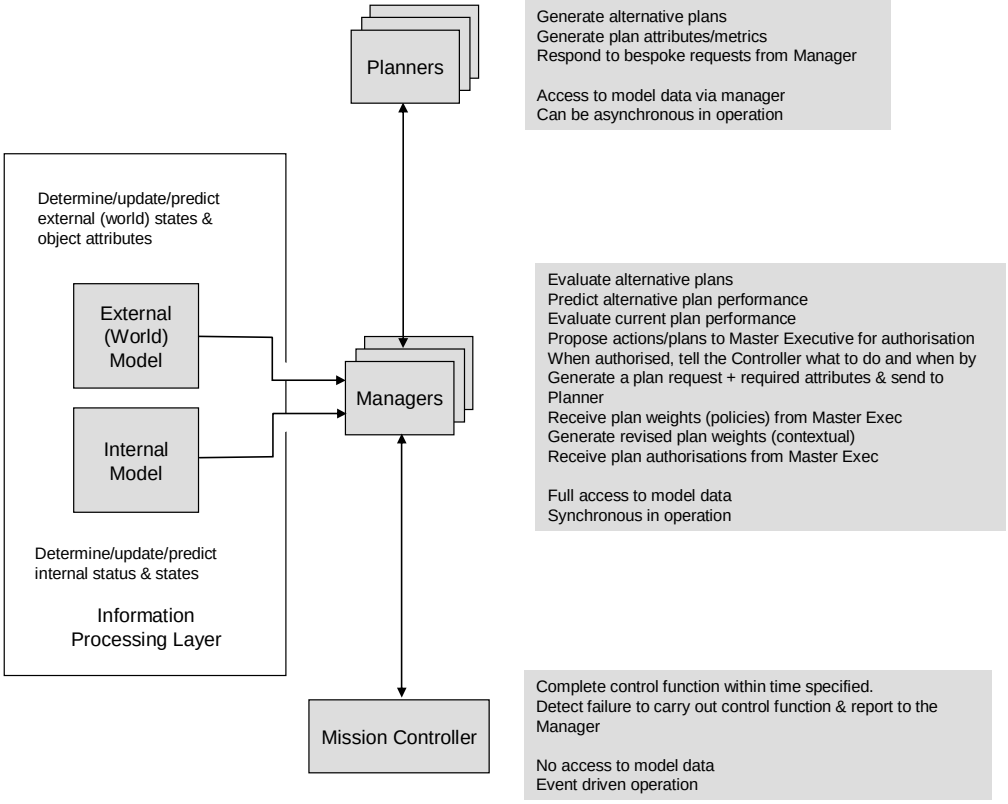


Figure 13: Generic 4-Layer Architecture

This schematic encompasses the above principles. Data is restricted to the planner and controller by the associated manager which provides the minimum required for each of the processes. However, the manager has access to a wide range of data via the information databus and to other managers, principally the Master Executive.

3.3.2.4 Reference View

The above constitutes the basic elements of the proposed architecture and their top level relationship. If we now structure the generic 4-layer model with the sub-system functional partitioning referred to earlier, we can now form an overall view of the architecture and see how the decision making system, described to here as the Mission Master Executive, sits alongside the other sub-systems and the aircraft vehicle system.

This can therefore be considered the reference viewpoint of the proposed design, a schematic of which is shown below:

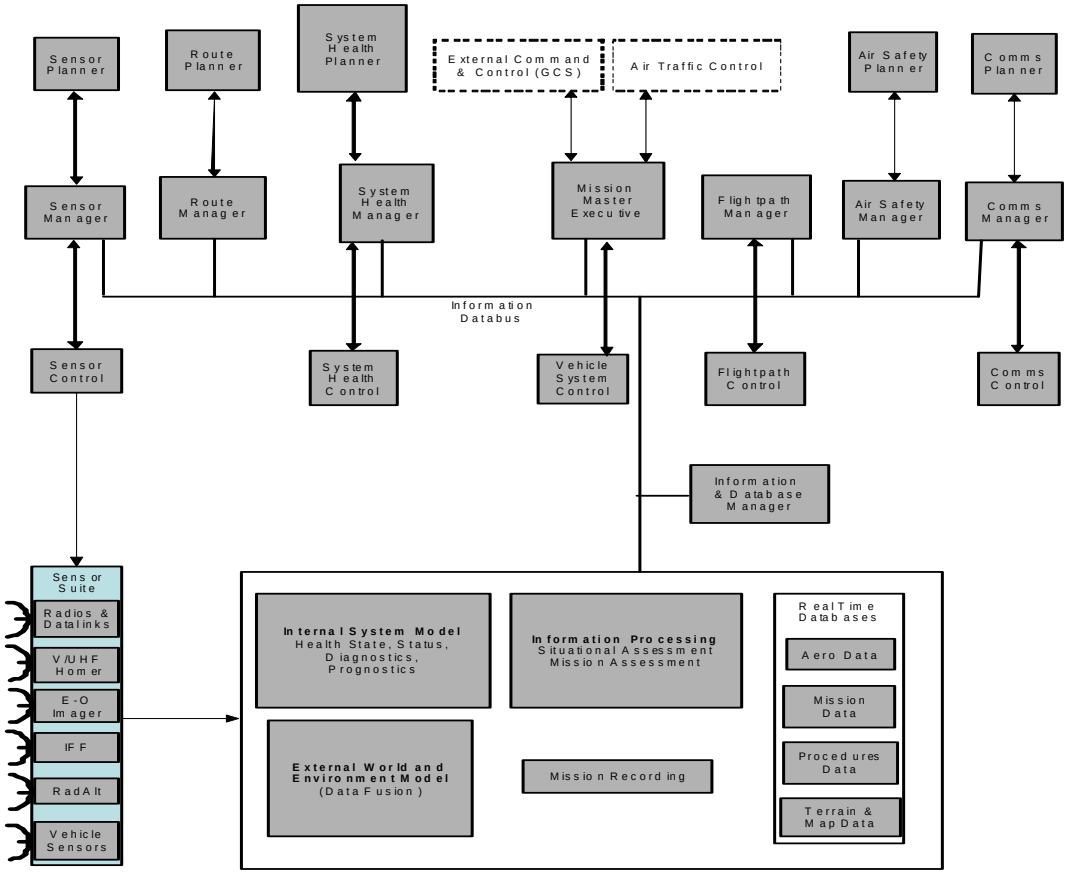


Figure 14: Reference View of Proposed AIMS Architecture

3.3.3 Functional View

From the reference view, we can start to add more detailed functionality and show the interfaces and message passing between sub-systems and functional modules. This design has been implemented[69] and a functional view of the latest implementation (June 2011), designated the Autonomous Integrated Mission System (AIMS) Capability (CAP) 2, is shown below:

[69] By a team of BAE Systems researchers, including the author.

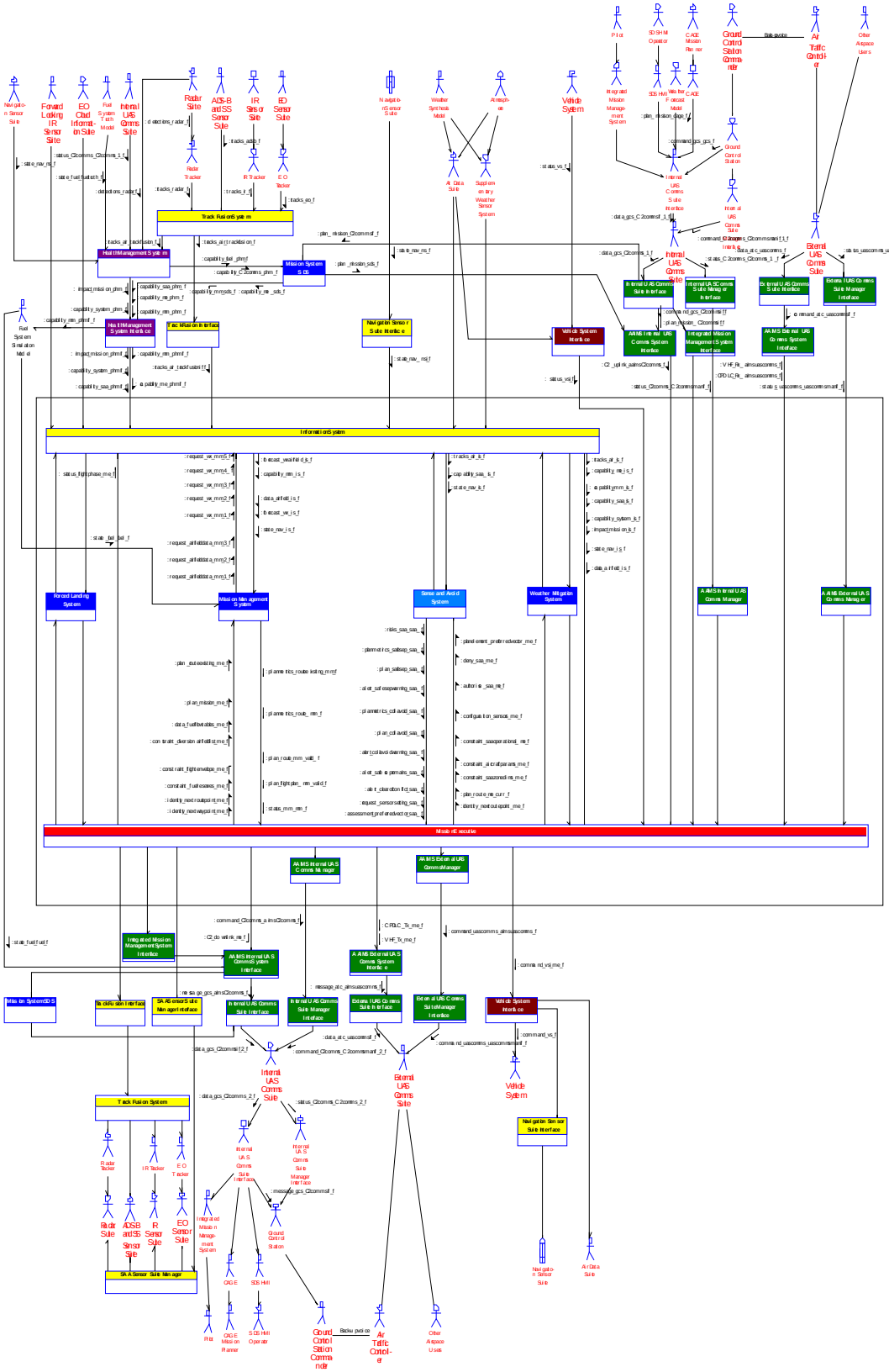


Figure 15: AIMS CAP2 Functional Architecture

The above functional implementation is described further:

Mission Master Executive

This sub-system is the prime area for decision and control, and is thus the centre for the autonomous behaviour. It comprises six elements:

- The Authorisation Manager – this element:
  - maintains the Authorisation Status List which is used by the system managers to check whether they have sufficient authorisation to command a particular control signal under their remit
  - Request authorisation changes from the GCS, on behalf of the system managers when they identify a command need for which they have insufficient authority.
  - Receives authorisation changes from the GCS in response to the requests and updates the Authorisation Status list.
  - Receives authorisation overrides from the GCS and updates the Authorisation Status list.
- The Mission Planner – this element
  - Takes input from the Mission Assessor, which is a Bayesian Belief Net (BBN) which outputs an assessment of Mission Success, based on time on task and sensor availability
  - Takes input from the Survival Assessor, which is a Bayesian Belief Net (BBN) which outputs an assessment of Mission Survival, based on the level of threat and the mission time remaining.
  - From the above inputs and fuel remaining, it calculates the optimum balance between fuel allocated to: countering present and future threats, maximising time on task, allocating safe margins for the landing fuel.
  - When the above balance falls to sub-optimum levels, it generates

alternative plans for the landing airfield and diversion alternative, together with their associated expected values, to the Master Executive for assessment and selection (or rejection).

- Flight Path Manager – this element provides the reactive (emergency) aspects of vehicle control:
    - Takes inputs of status from the:
        - Master Executive override (Priority 1)
        - Collision Avoidance Steering sub-system (Priority 2)
        - Threat Assessment and Counter sub-system (Priority 3)
        - System Health sub-system (for emergency diversion steering) (Priority 4)
    - Directs the Air System Control to respond appropriately if any or more of the above status signals are positive.
- Flight Phase Manager – this element:
    - Determines the current Phase of Flight
    - Determines the current Phase of Mission
    - Selects the appropriate plan for each of the above and directs the Air System Control to implement it.
- Air Systems Control – this element:
    - Generates the control signals for the operation of the air vehicle in response to the Flight Phase Managers commands
    - Receives feedback responses from the actuators
    - Reports success or failure back to the Flight Phase Manager
- Master Executive (ME) – this element:
    - Responds to ATC (not yet implemented) and GCS commands appropriately
    - Evaluates and selects (or otherwise):
        - route plans generated by the adaptive router – these will provide optimum routes around threats and no-go areas and plan the on task phase of flight.

- mission plans generated by the mission planner

System Health Manager

This sub-system provides prognostic and diagnostic data to the ME and proposes plans for emergency actions in response to failures.

Adaptive Router/Planner

This sub-system constantly calculates an optimum mission route and associated value metrics (fuel, time, threat level). If there is a difference between the current (authorised route) and the latest optimum route, it will propose the acceptance of the new route.

Air Safety System

This sub-system comprises two elements:

- Sense and Avoid – this element generates appropriate[70] emergency avoidance steering if it detects that a nearby external entity will pass within a pre-determined radius of the UAS.
- Threat Avoidance – this element will provide emergency steering to avoid threats such as storm cells.

Information Management System

This sub-system is responsible for generating all requisite data for the managers to act upon. It handles external (world) entity information generated by the sensors and communication suite and internal (state and status) information. It also retrieves and stores information to and from on-board databases.

[70] In accordance with the Rules of the Air as defined in the CAA Air Navigation Order, and if the object is within 10k and sufficient to provide a pre-determined miss distance (currently set to 500m laterally and 500ft vertically).

Vehicle and Flight System Interface

This sub-system is the main interface to the vehicle[71] and flight control[72] systems

3.3.4 Implementation of the AIMS Functional Architecture

The AIMS functional architecture described above was implemented by a team of avionic specialists during the ASTRAEA programme. This implementation was a based on the MoD's Synthetic Environment Based Acquisition(SEBA) process, which is described in Appendix D, and in doing so underwent several spiral updates. The nature and results of that testing is not relevant to this thesis as they was not designed to determine the qualities of the architecture. However, in operating the architecture, some valuable lessons were learned from the final development standard that fundamentally affects the architecture as defined above. These are discussed below:

3.3.4.1 HMI and the Authorisation Protocols

When the basic generic 4-layer architecture was first envisaged, the Human Machine Interaction was using the PACT Levels identified in Table 6. Initially, it was envisaged that only the pilot could change the PACT levels for specific functions. When operating the system in response to generated inputs, such as a Sense and Avoid encounter, it became quickly obvious that this was an unsound practice, primarily because it limited appropriate behaviour, especially when the pilot was not in contact with the vehicle. As an example, consider the Sense and Avoid scenario. If the pilot refuses to authorise the avoiding manoeuvre plan, assuming that this plan is correctly specified, then a collision will occur (which is unacceptable). If the pilot is unable to authorise the manoeuvre due to loss of communication a similar situation arises. However, if the system requests authority to manoeuvre (i.e. a PACT 4A level) upon cognisance of the collision risk and then, at say 30 seconds to impact, self

[71] Such as the undercarriage, flaps, brakes etc.

[72] This is primarily the autopilot, which is the only means of flight control.

escalates the level to PACT 4B (with a 10 second timeout) and then finally self-authorises the manoeuvre at PACT 5A in time for the collision to be averted, then the appropriate behaviour is now generated. This would be an example of PACT escalation due to time pressure. A similar case can be made for routing to avoid threats (such as bad weather). If the route change is minor (and of course that has to be defined), then the system could self-authorise at the PACT 5A level. If the change is medium, then 4B, and finally for major route changes, only PACT 4A could be allowed. This is an example of PACT escalation based on the degree of change of plan. A review of all such transitions was made and these are outlined in detail in Para. 5.2.

3.3.4.2 Authorisation of Controls

Although the architecture is designed to generate, maintain and implement plans, initially, the authorisation protocol was envisaged at the control or action level. For each switch in the UASs virtual cockpit[73], a separate authorisation was considered necessary. As the architecture was implemented, it became clear that this was unwieldy and uncoordinated. In implementing the Flight Phase Manager, which uses checklists to transition from one phase to another, it became clear that a generic approach was to consider plans as the basic construct for authorisation. In fact, eventually it became obvious that the entire architecture should be predicated on the selection, authorisation and actioning of plans alone, even if those plans contain only a single “atomic” action such as a Sense and Avoid manoeuvre. This is in alignment with the original generic 4-layer architecture construct.

However, in sequencing these plans using a “heavyweight” but conventional programming language such as ADA, it was obvious that quite complicated code could be generated from applying even simple plans. That complexity increases the level of debugging and testing, and therefore, cost. Even a simple assessment (and there were several detailed Hazard Analyses done) indicates that the vast majority of the architecture will be Safety Critical. Safety Critical

[73] Originally, the controls were seen to be identical to those in a manned cockpit i.e. switches.

code can be 10 to 30 times the cost of conventional code. Although not part of this Thesis, a review of the theoretical background of agent programming languages indicates that the architecture is fully aligned with their approaches and indicates a way ahead for the implementation of a future system.

Moving to a plan based system of control also points the way to alternative means of specifying behaviours by the human operator. It was mentioned earlier that for an autonomous system, the operator is a manager of the autonomous system which is controlling the vehicle. Managers of human teams (well, good ones do) tend to specify objectives that need to be achieved or maintained rather than issue micro managing orders. Plans provide the ideal mechanism to do so since they are primarily concerned with the sequence of events that can lead from a current state to a desired or goal state. If that goal state can be identified by an objective, then the HMI dialogue can be reduced to setting objectives for the machine. On receiving an objective, the machine will back chain to link together a sequence of plans to get from the current state to the goal state and achieve the set objective. Again this concept is in line with current Agent Programming languages such as JACK, and Agent Factory.

The current way of delivering behaviours is by a (crisp) rule based system. It was noted that improved human understanding and more appropriate response could be generated by moving to a fuzzy rule based system with linguistic hedges providing a “soft” transition from one belief (and therefore behaviour) to another and reducing the occurrence of flip-flopping of behaviours as beliefs swap. None of the above (agent programming or fuzzy rules) has been implemented or tested at this stage and the benefits remain unproven.

3.3.4.3 SEBA Validation

As part of a separate strand of research into system development processes, the development of the architecture provided an opportunity to test the theoretical advantages of the SEBA process (see Appendix D). SEBA is predominantly concerned with the construction and management of an evolving set of Synthetic Environments (SEs), Models and Simulations (SEMS), of

increasing complexity. The aim is to mitigate integration risk by simulating, modelling and emulating the necessary variables to ensure the equipment target remains within the specified boundaries (ordinarily of time, cost and performance).

Consequently, it was decided to develop the AIMS architecture by following this approach. Initially, a small program, written in Java, and contained within a simple environment was written. This only had simple navigational and flight phase functionality[74] but had a separate Master Executive to control the vehicle. This program effectively proved the concept and basic architecture. This program was then expanded in scope, re-designed into the partitions previously outlined and re-written in ADA. This prototype, at Spiral 1, along with a detailed and representative vehicle model, was then embedded into a far more sophisticated SE. This SE could represent other vehicles, and for the Spirals 2/3 system, a high fidelity simulation of the UK Air Traffic Control system. The functionality at Spiral 3 included a Sense and Avoid System and an adaptive Navigation and Mission Planning System (called the Mission System) all controlled by the Master Executive suite. The Mission System created and constantly maintained a suite of flight and mission plans for use in normal flight and for contingencies such as: Return to Base, Divert to Nearest Suitable Airfield, Divert to Nearest Available Airfield and Emergency Ditch. These plans comprised: Routes, Fuel, Times of Arrival and Navigation data. In addition, the router could generate alternative plans to re-route around bad weather.

Although a Ground Control Station was yet to be developed, the Master Executive was designed to respond to the Air Traffic Controllers commands generated by the ATC workstation. A screenshot of this ATC workstation and SE is given below.

[74] It would command the (very simple) vehicle model to take off, waypoint steer along airways and land.

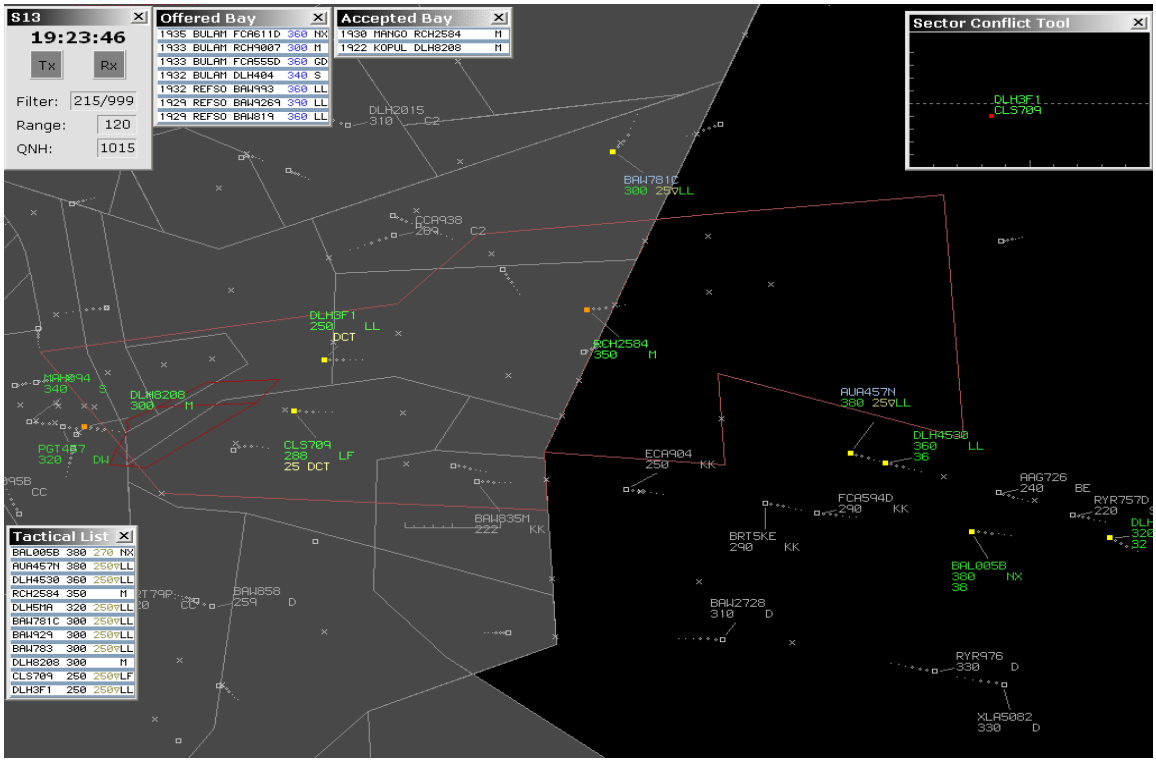


Figure 16: UK Air Traffic Control Simulation Environment

At Spiral 4, the functionality embedded in AIMS was further developed and included a Health Management System for identifying failures which informed the Emergency Manager (EM) in the Master Executive suite. The EM could then follow appropriate checklists to handle the emergency and if necessary invoke the appropriate contingency route. Spiral 5 (re-designated Capability 1 or CAP 1) was designed but never implemented. However the current system, CAP 2, now incorporates a research Ground Control Station (GCS)[75]. The GCS/AIMS interface has been implemented as an extension of the NATO Stanag 4586 protocol and is capable of providing the full range of authorisations, using the PACT levels, to the Master Executive.

This system development process using the SEBA template has proven highly successful and, in general, has validated the theoretical advantages outlined in Appendix D. In particular, it has demonstrated the reduction of integration risk by the adoption of evolutionary acquisition and spiral development.

[75] This has been designed by Human Factors researchers.

4 The Analysis of UAS Accidents – Causation and Prevention

4.1 Aircraft Safety and Regulations

What is deemed to be safe, or more accurately, tolerably unsafe, for flight operations is defined by the appropriate regulatory authority. In the USA, for civil aviation, this is the Federal Aviation Administration (FAA) and they publish regulations, known as the Federal Aviation Regulations (FARs) that cover the whole arena of flight operations. In the UK, the equivalent body is the Civil Aviation Authority (CAA), however, they are part of the Joint Aviation Authorities (JAA) of European states and they publish, broadly equivalent, Joint Aviation Requirements (JARs). A new system of European aviation regulation (EASA – the European Aviation Safety Agency) was introduced in September 2003, and will gradually supersede the JARs. In this document, FAR and JAR are regarded as the same and the term JAR is used consequently.

Military aircraft in the UK are certificated by the MoD and the approach used is largely based on the analysis of safety cases. In recent years, the CAA has concluded that UASs, in the UK at least, will be required to comply with defined codes of requirements, rather than a Safety Target approach [33].

General airworthiness regulations for aircraft are covered in Sections 23 (Light/Commuter) and 25 (Transport) of the JARs. For UASs, briefly, each aircraft is covered by Section 23 or 25 according to a variety of criteria, e.g. size, weight, kinetic energy, use etc. As this thesis is aimed at medium to large UASs, we will assume that medium UASs are covered by JAR 23 and large ones, JAR25.

A particularly relevant paragraph in the JARs is Para. 1309[76] . Put simply, "*this requires justification that all probable failures, or combinations of failures, will not result in unacceptable consequences"* [33]. This requires the identification of

[76] This is so important that it is frequently referred to as "1309" as if it were a separate document.

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. This is set out in the Table below[77]:

Severity of Effect	Prob.	Definition or Example
Catastrophic	$< 1 \times 10^{-9}$	Multiple fatalities[78] – usually with the loss of the aircraft
Hazardous	1×10^{-7} to 1×10^{-9}	Reduce the capability of the aircraft or the ability of the crew to cope with adverse operating conditions. Specifically: A large reduction in safety margins or functional capabilities Physical distress or excessive workload such that the flight crew cannot be relied up on to perform their tasks accurately or completely Serious or fatal injury to a relatively small number of the occupants other than the flight crew
Major	1×10^{-5} to 1×10^{-7}	Reduce the capability of the aircraft or the ability of the crew to cope with adverse operating conditions. Specifically: A significant reduction in safety margins or functional capabilities A significant increase in crew workload Discomfort to the flight crew Physical distress to passengers or cabin crew, possibly including injuries.
Minor	$>1 \times 10^{-5}$	No significant reduction in aircraft safety. It may involve crew actions that are well within their capabilities Specifically, a slight reduction in safety margins or functional capabilities

[77] The table has been provided by the Safety Regulation Group, Civil Aviation Authority. The equivalent FAA table includes the category "No Safety Effect".

[78] The CAA refuse to formally define "multiple" fatalities. Informally, they have commented that greater than four would be likely to be categorised as "multiple".

		A slight increase in crew workload (e.g. flight plan changes) Some physical discomfort to passengers or cabin crew

Table 7: Acceptable Probability of Failure, their Effects and their Definition

There are a few problems with the above when UASs are under consideration. 1309 is specifically concerned with maintaining safety by keeping the aircraft airborne or landing safely. It is also clearly aimed at passenger aircraft (according to the above definitions). A UAS is unlikely at this present time to carry passengers, nor, by definition, does it carry a crew. In addition, there is a distinct possibility that UASs can be designed to crash safely (some are already), and they can certainly be designed to land automatically and safely. By these definitions and considerations, a UAS failure cannot in theory have a Catastrophic outcome. This point has been discussed with a Senior Surveyor of the Safety Regulation Group, Civil Aviation Authority (CAA). The official view is in accordance with the above, however, she agreed that, notwithstanding this, a UAS accident that resulted in the UAS impacting the ground (or a ship) killing several people would be surely construed as Catastrophic, even if only on societal grounds. In fact, there are probably many outcomes that would involve multiple fatalities, which should therefore be classified as Catastrophic.

4.2 Aircraft Accident Analysis

The major, some would say the only, difference between manned and unmanned aircraft accidents and their causes is the fact that, for unmanned aircraft, the pilot is remotely situated. Although this statement could be considered obvious to some, this view is not held by all, particularly some pilots. During an interview with three BAE SYSTEMS UAS pilots[34], it was quite clear that they did not accept that they were in any way less in control of the UAS by being remotely situated, than would be the case for a normal manned aircraft. Their (unanimous) view was that, although not actually flying the aircraft directly, their situational awareness (SA) was undiminished due to the quality of the information available to them at the Ground Control Station. During discussions, where it was pointed out that many ways of achieving SA were

denied to them, notably peripheral vision, tactility ("seat of the pants flying") and force feedback, it was clear that they felt these aspects to be of minimal importance. When they were shown that the accident rates between manned (GA) and UAS accident rates is of an order difference, roughly 6:80 (as will be shown), they were at a loss to explain this in terms of their skills or SA and genuinely thought that this was due to reliability problems in UASs. Whilst the latter is true to a degree, but the detail of this is outside the scope of this work, the analysis below gives strong indications that there is plenty of evidence that this remoteness is a major cause of UAS accidents.

However, to understand the effects of this remoteness in detail, the following analysis is broken down into several components, which when brought together at the end, will hopefully justify the above statement. In doing so, the primary concentration will be on the Human Factors of UAS control and its impact on the prevalence of UAS accidents

To do this initially, an understanding of Human Error is presented. This naturally lends itself to a discussion on the taxonomies for classifying Human Error types. Following this, an analysis of manned aircraft accidents is first given, with a particular emphasis on the Human aspects. This is then followed by an analysis of unmanned aircraft accidents. From a comparison of these, conclusions can then be drawn highlighting the differences.

4.2.1 Reason's "Swiss Cheese" Model of Human Error

There are many models describing Human Error in various forms such as a breakdown in Situational Awareness, proposed by Endsley [35] or as a wrong formulation of, and persistence in retaining[79], "mental models, Besnard *et al* [36], and Burns [37]. However, a widely accepted model, primarily because of its abstract nature, and therefore broad applicability, is that proposed by James Reason. In his book "Human Error" [3], Reason argued that accidents occur because of a breakdown in the interactions and operation of the elements within a "productive system". He identified these elements as follows:

[79] Often described as The Confirmation Bias.

- Decision Makers – plant and corporate management.
- Line Management – operations, maintenance and training etc.
- Preconditions – reliable equipment, motivated workforce etc.
- Productive activities – integration of human and mechanical elements.

He particularly differentiates between errors that precede a fault, failure or problem and which he refers to as *monitoring failures* and those that follow it, referred to as *problem solving failures.* The former is seen as a latent failure or "an accident waiting to happen" whilst the latter is regarded as an active failure.

When the integrity of each or any of the above is degraded, the system as a whole is more susceptible to catastrophe. Thus, these elements can be seen as barriers to failure and lapses of integrity are depicted as breaches or holes in these barriers, the whole system can be viewed in the following manner:

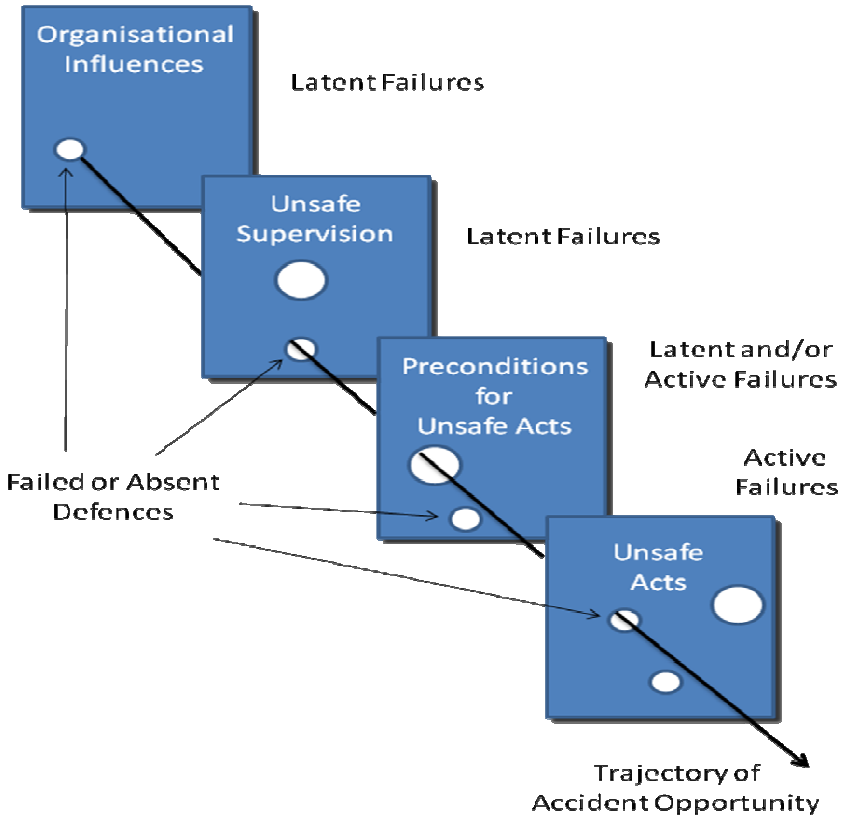


Figure 17: The "Swiss Cheese" Model of Human Error

Reason's approach is attractive because it identifies and acknowledges the latent factors, which may be around for weeks or years that lead up to the final

unsafe act of the operator. The approach thus forces investigators to consider the accident sequence in its entirety. However, as it is by nature an abstract model, it does have some drawback in that it does not identify what the “holes” are. For this a taxonomy for the particular work environment is required.

4.2.2 Taxonomies of Human Error in Aircraft Accidents

Just as for models of Human Error, there are a wide variety of taxonomies or classification systems that have been used for the analysis of Human Error. A system used throughout the US Army is DA PAM 385-40, “Army Accident Investigation and Reporting”. In this approach, accidents are classified according to either materiel failure, environment or human error. The latter is further classified into individual failure, leader failure, training failure, support failure or standards failure. Another, which has gained prevalence in the civil aviation community and the USAF is the Human Factor Analysis and Classification System (HFACS) of Shappell and Weigmann [2]. As will noted later aligning these different taxonomies for the continued analysis of UAV accidents presents considerable difficulties. The attractiveness of HFACS is that it supplies what was acknowledged to be the missing link in Reasons theory – identification of the “holes”.

Since HFACS is more widely used and because it is heavily based on the accepted human error model of Reason, it will now be discussed in detail.

4.2.3 The Human Factor Analysis and Classification System

The Human Factor Analysis and Classification System (HFACS) builds directly upon Reason’s work has been developed to assist in the classification and analysis of human error. The work so far has been primarily aimed at aircraft accidents but has been extended for other areas, such as Air Traffic Control. The system has also been used to analyse human error in Unmanned Air Vehicle accidents, but has not yet, as far is known, been used to investigate the interactions and acts of the air based command component (the autonomous part of the UAS).

HFACS builds on Reason's model by formally identifying and classifying the natures of the "holes", whilst keeping the organisational elements previously identified. In doing so, it provide the means to analyse the causal nature of human error in accidents. The formal structure of HFACS is shown below.

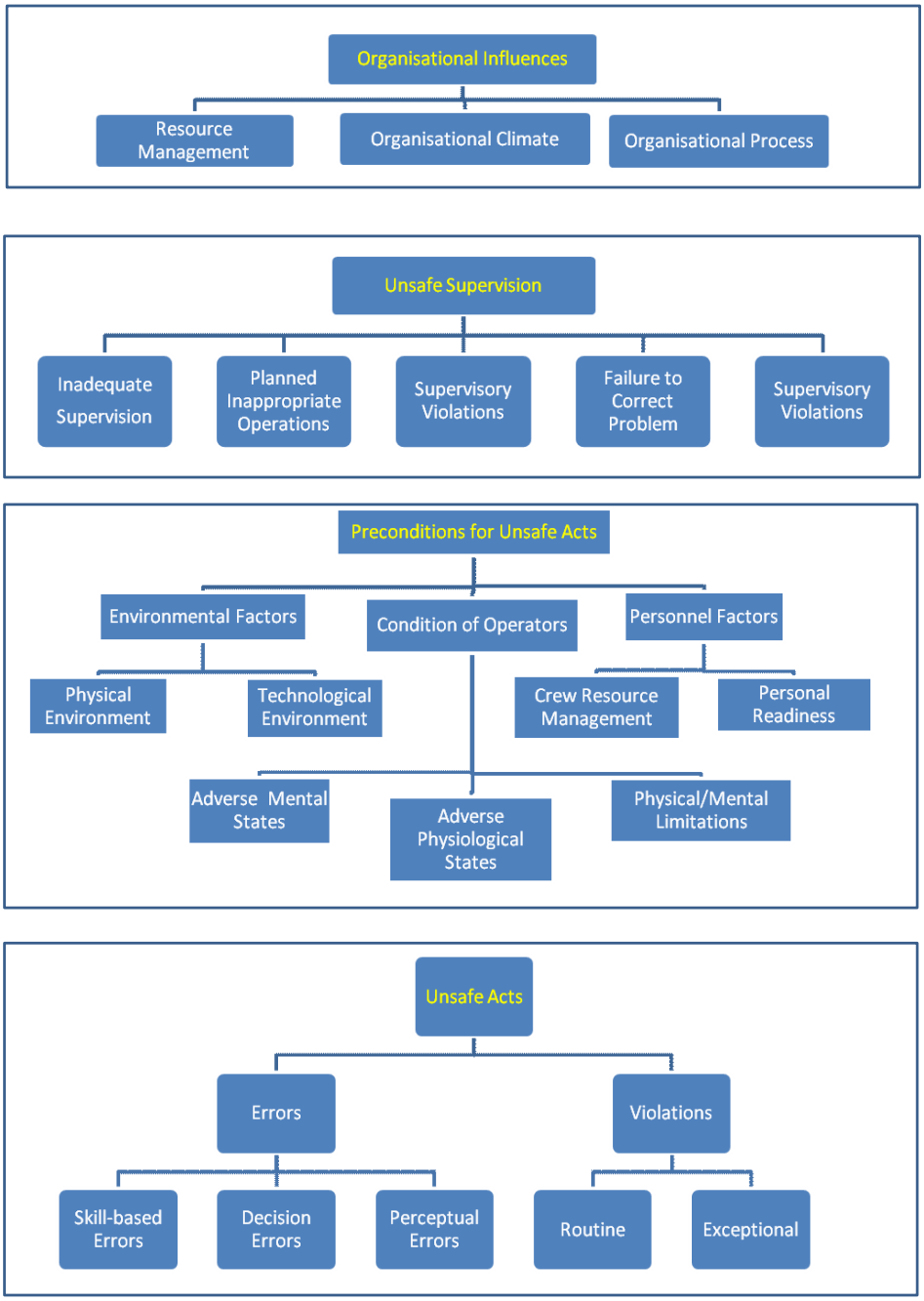


Figure 18: The Human Factors Analysis and Classification System

HFACS was developed empirically and has been refined over many years by analysing a vast amount of military and civil aviation accident reports each of which contained several human causal factors. As noted, although HFACS is one of several taxonomies of human error, it has become established as the leader in its field. In fact, it is now used by the US Navy, Marine Corps, Army, Air Force and Coast Guard [2].

4.2.4 Manned Aircraft Accident Analysis

4.2.4.1 USA General Aviation Accidents

Each year, the Aircraft Owners and Pilots Association (AOPA) Air Safety Foundation publish facts and figures for General Aviation (GA) operations in the USA – a report known as The Nall Report[80]. General Aviation covers all flights for civil aircraft below the empty weight of 12,500lbs. This report therefore excludes the larger airliners but does include light commercial aviation and multi-engine aircraft. Since this report covers 1385 accidents for a total of 21.4 million flight hours, the results can be considered statistically significant. Additionally the results are broadly comparable with the UK.

Since, the body of statistics for GA is sufficiently large, and the accident rate of airliners is at least 2 orders of magnitude lower which makes their body of statistics commensurately lower, together with the fact that the target level of safety for UAVs is likely to be (initially at least) comparable with GA, only GA accidents will be considered in this analysis.

In 2008, the authors reported that the accident rate for GA had fallen over the previous 10 years from 6.81 to 6.47 per 100,000 flying hours.

Of these accidents[81], 71.9% were attributed to being caused by the pilot. The following is the distribution of these pilot related accidents in 2006 according to

[80] So called in memory of Joseph Nall, a National Transport Safety Board official who died in an airline accident.

[81] AOPA defines an accident as: "An occurrence incidental to flight in which, "as a result of the operation of an aircraft, any person (occupant or non-occupant) receives fatal or serious injury or any aircraft receives substantial damage."

their flight phase and/or main factor. It should be added that these are broadly comparable with earlier, though recent, years.

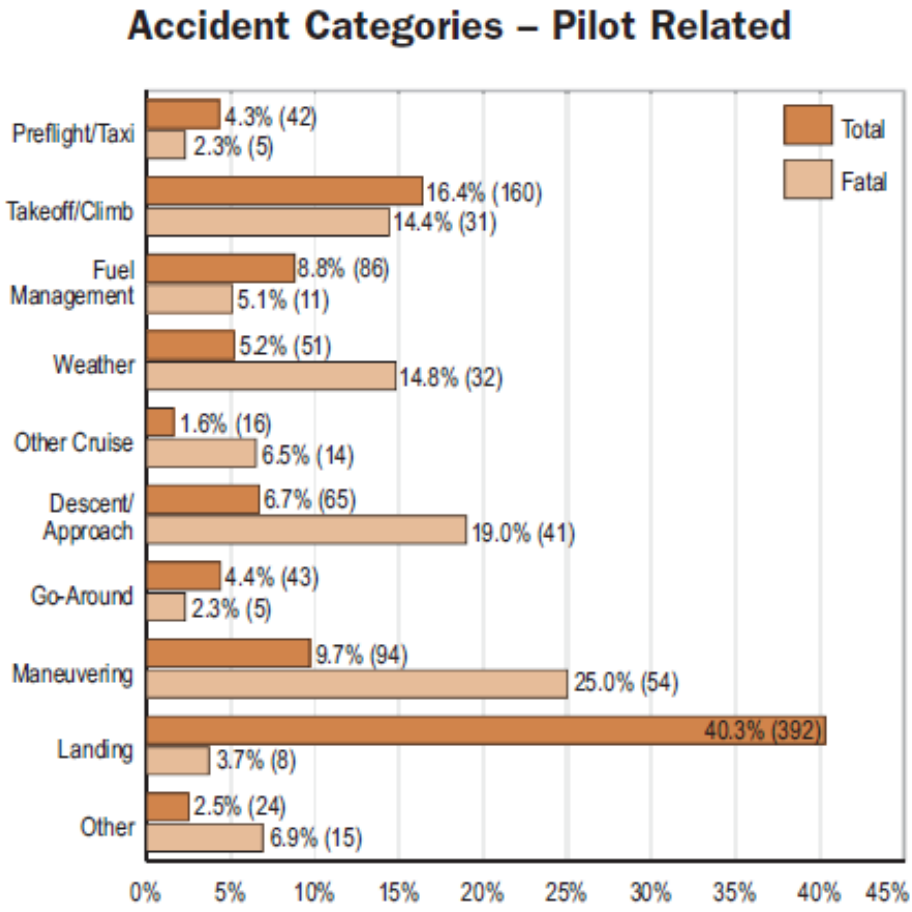


Figure 19: 2006 USA Pilot Related GA Accidents in Flight Phase[38]

It can be seen that, Landing (the most dominant, 40%), Take off/Climb and Manoeuvring are the major factors contributing to the overall accident rate. However, the number of accidents involving Fuel Management[82] is fairly significant and, which the report notes, should be "*easily preventable."* This subject is dealt with in detail later.

In discussing these factors, the report highlights that "*Take-off and climb accident statistics continue to show gradual improvement in both total and fatal*

82 Fuel Management (or more accurately the lack of) covers loss of fuel (i.e. running out of through bad planning and operation), fuel starvation (the inability to use on board fuel due to failure of some part of the fuel system) and fuel contamination.

crashes. Loss of directional control is a frequent causal factor in these accidents". In fact, the report highlights that 61% of all take-off and climb accidents are due to loss of control which will be highlighted later as skill based errors. For Manoeuvring flight, 64.8% of all accidents were due to loss of control, whilst the corresponding figure for approach and landing was 39.3%. Interestingly, the report concludes that inexperience - a high proportion of GA flight hours are by inexperienced pilots - was not a significant factor in any of these figures.

	All Accidents	Fatal Accidents
Human Performance Issues	1,372	275
Aircraft handling/control	990	229
Planning/decision	489	106
Use of aircraft equipment	148	22
Maintenance	87	14
Communications/information/ATC	69	8
Meteorological service	4	4
Airport	1	0
Dispatch	0	0
Underlying Explanatory Factors	116	57
Qualification	41	18
Physiological condition	31	25
Psychological condition	25	8
Aircraft/equipment inadequate	8	1
Institutional factors	8	6
Procedure inadequate	5	4
Material inadequate	2	0
Information	1	0
Facility inadequate	1	0

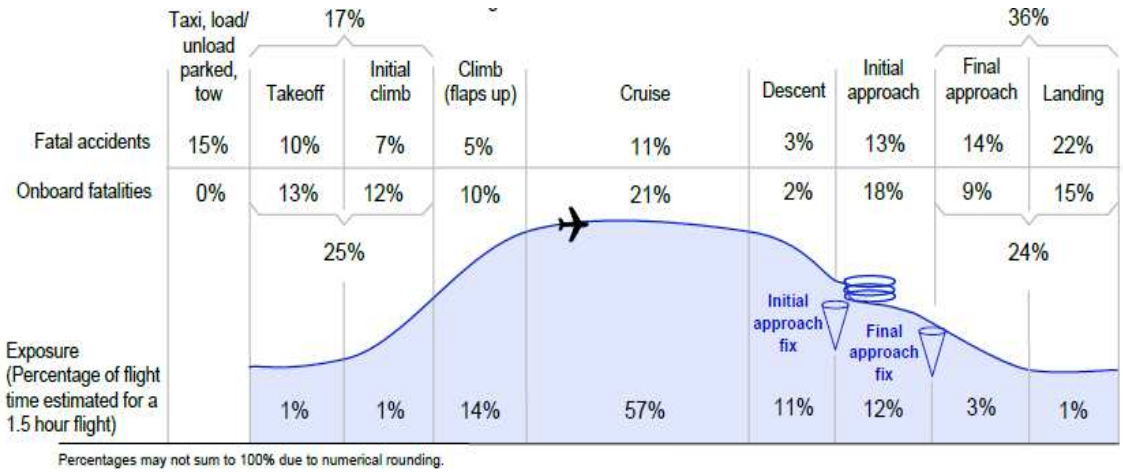
Table 8: Accident Factors and Issues

Table 8 above highlights that aircraft handling and control is a major influence on the overall accident rate. This is significant since the vast majority of GA aircraft are flown manually.

4.2.4.2 Commercial Manned Accidents

It is important to now consider manned commercial aircraft to see if there are any differences in the accident profiles with GA. The major differences of course are that large commercial aircraft are subject to more stringent regulations both for manufacture and operation. Additionally, the pilots are much more highly trained, current on type and they operate in a rigidly controlled airspace.

The table below indicates commercial aircraft accidents classified according to



Phase of Flight.

Figure 20: Commercial Aircraft Accidents in Flight Phase [*39*]

It can be seen that, again, Approach and Landing are the most dominant (36% compared to the GA figure of 40%)[83]. Also Take Off and Initial Climb in Figure 1Figure 20 is 17% compared to the GA figure of 16%. Therefore despite the differences in operation highlighted, there seems to be a broad similarity between the two manned aircraft types in terms of accidents by Phase of Flight. In addition, [39], shows that, for Fatal Accidents involving the Worldwide Commercial Jet Fleet (2001-2010), 23% were due to Loss of Control (LOC), 20% due to Controlled Flight into Terrain (CFiT) and 20% during Landing. The full range is shown in the figure below:

[83] There is a slight discrepancy in the use of the terms: Approach and Landing. For GA, Approach is often used to describe manoeuvring in the vicinity of an airfield whereas for commercial aircraft there is a clear distinction between initial approach, final approach and landing. The terms Landing (GA) and Final Approach and Landing (Commercial) are taken to be roughly equivalent.

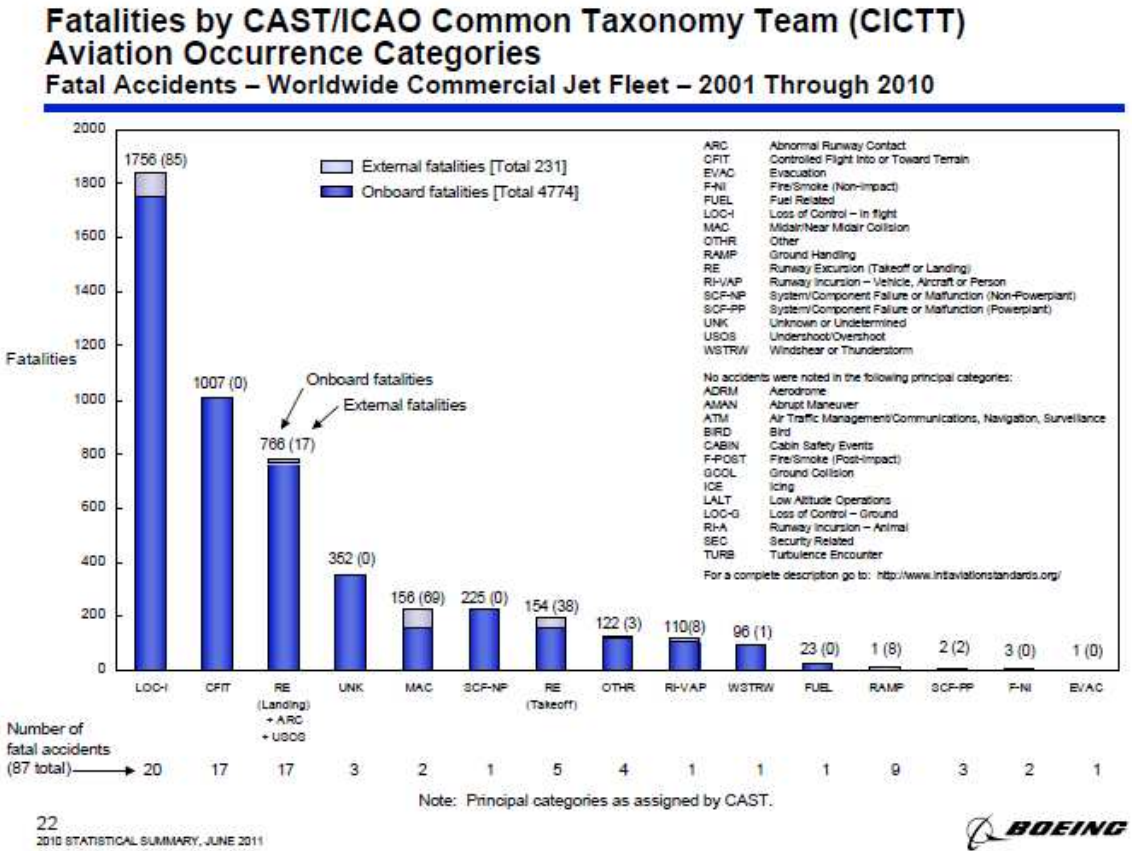


Figure 21: Fatal Accidents involving the Worldwide Commercial Jet Fleet 2001-2010

This reduction in accidents due to LOC is probably due to the far more prevalent use of autopilots, however this still remains the greatest category for manned commercial aircraft.

4.2.5 Unmanned Aircraft Accident Analysis

A review of unmanned aircraft accidents is now presented. As stated in the Introduction, the scope of this subject is focused on Medium to Large UASs such as *Global Hawk*, *Predator* and *Reaper.* In addition, the primary emphasis is on the Human aspects of accidents.

4.2.5.1 Background and Historical Aspects

There is a large body of information on accident trends for modern UASs, compared to their predecessor RPVs, but unfortunately much is incomparable

without significant effort for a variety of reasons. UASs came very much to the fore in the Bosnian war of 1992-1995 and the losses during that time could not often be confidently ascribed to being accidents or as the result of enemy action. In addition, these early, and therefore immature, systems were very much prototypes rushed into service. Finally, the weather conditions in which these prototypes had been developed (sunny California) were very much different to the far more hostile climate of the Balkans. All these factors therefore make the analysis of UAS accident prior to 1999 extremely difficult if not impossible. Subsequent analysis is somewhat easier but still difficult because of the variety of taxonomies used. An overview, with accompanying notes, of the more notable analyses and data sources available is given in the table below:

Ref.	Author(s)	Title/Data	Notes
1	A.P.Tvaryanas W.T.Thompson S.H.Constable	US Military Unmanned Aerial Vehicle Mishaps: Assessment of the Role of Human Factors Using the HFACS Classification System (March 2005)	Used HFACS and provides a substantial analysis of the differences between USAF, Army, Navy and Marine operators and their accidents
2	S.D.Manning C.E.Rash P.A.LeDuc	The Role of Human Causal Factors in US Army UAV accidents (2004)	Analysed US Army accidents using HFACS and the US Army's own classification taxonomy DA PAM 385-40
3	B.M.Rogers B.Palmer	Human Systems Issues in UAV Design and	Analysed US Army and Air Force accidents using a

	J.M.Chitwood	Operation (2004)	human-systems issues taxonomy
4	K.W.Williams	A Summary of Unmanned Aircraft Accident/Incident Data: Human Factors Implications (2005)	Analysed in all US Services using a novel two tier classification system
5	USAF	United States Air Force Class A Aerospace Mishaps	Accident Reports and Summaries for all USAF air accidents between 2000-2011

Table 9: A Summary of UAV Accident Analyses

4.2.5.2 Review of Recent Studies

Ref 1 above found that human causal factors were present in 68% of UAS accidents (mishaps in US parlance). However, there was a marked difference in patterns of human failure across the three forces. The USAF operators (who are generally trained pilots) generally failed when involved in instrumentation and sensory feedback systems, automation and channelised attention. Navy/Marines failures were generally associated with workload attention and risk management while the Army failures were more procedural, publications, training and organisation. Specifically, they identified a preponderance of skill based errors in the Air Force and violations in the Army, but no difference in decision based errors [40].

Ref 2 above provided a comparison of taxonomies, HFACS and the US Army methodology and found broad agreement but favoured HFACS due to its ability to further separate causal factors. The results are in general agreement with Ref1 above with decision and organisational errors dominating for the US Army accidents[41].

Ref 3 proved to be unavailable but a review of this was conducted by [40] which reported that they found the highest number of accidents occurred among those with the least UAS experience (<500 hours) and those with the most flight experience (> 1000 hrs.). This fact could be supportive of the view that pilot remoteness may be a key factor. Experienced pilots, but inexperienced UAS operators, may find the transition from cockpit to ground station a difficult one.

Ref 4 notes that many of the human factors issues identified are highly dependent on the system being flown, the type of automation employed and the user interface. Particularly of note is the observation that these aircraft (Predator, Reaper and Global Hawk) are not flown in the conventional sense but rather "commanded" (*cf*. Para. 2.2.5 and below) and that the user interface should focus on facilitating the task of issuing commands and verifying that those commands have been accepted and followed[84][42].

4.2.5.3 UAV Accident Analysis

Accident analysis for all UAV accidents, where data is available, is given at 0. Due to the concerns above regarding machine specific causes, a subset of this data has been used for the analysis. This subset consists of accidents for which the USAF has published an accident report and which cover only *Predator* and its larger cousin, *Reaper,* both of which are controlled in an identical manner.

In conducting the analysis, as most other researchers have found, it was noted that most accidents have several contributing factors. If these are all taken into account, the summation of causes will be greater than 100%. Since this was never intended to be an in depth analysis but more of an investigation to determine major trends, where possible, only the primary cause was selected. In addition HFACS, as presented at Reference, has some anomalies. Primary amongst these is the classification of inattention. "Failed to Prioritise Attention" is given as an example of an Unsafe Act, Skill Based Error. However, Channelised Attention is given as an example of "Pre-conditions for Unsafe

[84] An seemingly obvious point which makes one wonder at the design concepts the HF engineers followed.

Acts, Substandard Conditions of Operators, Adverse Mental States. To discriminate between these it is necessary to consider the underlying basis of HFACS, which is Reason's model. An Unsafe Act is the final act of the operator (to use Reason's use of the word) leading directly to the accident occurrence. It is an active failure. Pre-conditions for Unsafe Acts is a latent failure which may occur several seconds or even hours before an accident. Adverse Mental States also implies a degree of stress, tiredness or mental limitation. For this analysis, if the operator was unaware of a developing state requiring his attention or failed to direct his attention appropriately **and** that directly contributed to the accident, it was classified as a Skill Based Error. If it was due to stress or any other adverse mental states, it was classified as a Pre-condition[85].

Finally, it is important to make a distinction between types of Human Error. In this analysis, Human Error has four contributors: Operator Error (an unsafe act), Supervisory Error, Design Error (by the human designer, and Maintenance Error (by the human maintainer). The last two categories, together with Weather and Manufacturing Error, are not included in the HFACS part of the analysis. The results of the analysis are presented below.

Results by Primary Cause

Primary Cause	Number	% of Total
Operator Error	20	37.8
Manufacturing Error	3	5.7
Design Error	10	18.9
Mechanical Failure	9	17.0
Maintenance Error	7	13.2
Supervisory Error	2	3.8
Weather	2	3.8
Total	53	

Table 10: Primary Causes of Predator/Reaper Accidents 2000 – 2011

[85] As it turned out, all inappropriate attention directions were classified as Unsafe Acts.

From the above, it can be seen that 39 failures (74%) were directly due to human error, though only 19 of these can be ascribed to the operator as seen in the table below.

HFACS Primary Cause	Number	% of Total
Unsafe Acts	19	59
Pre-cursors to Unsafe Acts	0	0
Unsafe Supervision	2	6
Organisational Influences	11	34
Total	32	

Table 11: HFACS Analysis of Predator/Reaper Accidents 2000-2011

All but one of the causes due to Organisational Influences were due to a poor design of the vehicle. In many of these cases, the accident was directly attributable to a single point failure. For manned aircraft (military and civil), certification regulations, as described in Para. 4.1 (though civil), are specifically targeted at identifying, and mitigating by design, such single point failures. What is somewhat surprising is the fact there appears to be no Pre-cursors to Unsafe Acts **as the primary cause**. This again highlights the problem of not considering all the causal factors in an attempt to produce a probabilistic approach to accidents analysis.

Since the focus of this study is the reduction in accidents due to decision/control mechanisms, maintenance, design, and reliability factors were ignored. Only the identified Unsafe Acts were further analysed and the results presented below.

HFACS Unsafe Acts	Number		% of Total
Errors			
	Skill Based Error	11	58
	Decision Based Error	4	21
	Perceptual Error	3	16
Violations	Routine	1	5
Total		19	

Table 12: HFACS Analysis of Unsafe Acts-Predator/Reaper Accidents 2000-2011

A full list of these 19 unsafe acts is provided in the table below.

	Cause	Unsafe Act	Notes + POF	HFACS Cause
1	**Pitot static icing - Non-use of pitot heating**	**Skill Based Error**	**Cruise**	**Failed to follow procedure**
2	**Incorrect procedures during hand over of control**	**Skill Based Error**	**Climb**	**Failed to follow procedure**
3	Pilot loss of control. Landing attempted on wind gust limits	Decision Error	Manual Flying Landing	Accepted an Unnecessary Hazard
4	**CFIT- Failure to monitor altitude**	**Skill Based Error**	**Descent**	**Failed to Prioritise Attention**
5	**Pilot loss of control, Incorrect diagnosis of icing.**	**Decision Error**	**Cruise**	**Over controlled the Aircraft**
6	Late executed go around	Decision Error	Manual Flying Landing	Poor technique/airmanship
7	Failure of pilot to correct high flare	Violation (Routine)	Manual Flying Landing	Committed approach outside published command criteria
8	Crashed short of the runway	Skill Based Error	Manual Flying Landing	Failed to follow procedure
9	Pilot failed to control aircraft glide-path	Perceptual Error	Manual Flying Landing	Due to misjudged flight path
10	Pilot turned off Stability Augmentation	Decision Error	Manual Flying Cruise	Inappropriate procedure
11	**Engine Fuel shut off signal inadvertently sent**	**Skill Based Error**	**Cruise**	**Failed to follow procedure**
12	**Inadvertent engine shut down in flight**	**Skill Based Error**	**Cruise**	**Failed to Prioritise Attention**
13	Pilot induced oscillation on landing	Perceptual Error	Manual Flying Landing	Due to misjudged flight path
14	**Controlled Flight into Terrain**	**Skill Based Error**	**Cruise**	**Channelised attention of flight crew**
15	Crashed on Touch and Go	Perceptual Error	Manual Flying Landing	Due to misjudged flight path
16	**Crashed during Taxi**	**Skill Based Error**	**Taxi**	**Failed to follow procedure**
17	Crashed during Take Off	Skill Based	Manual	Failed to follow

		Error	Flying Take Off	procedure
18	Crashed during Hi AOA demo	Skill Based Error	Manual Flying Cruise	Failed to Prioritise Attention
19	Crashed in descent	Skill Based Error	Manual Flying Descent	Failed to follow procedure

Table 13: HFACS Unsafe Acts-Predator/Reaper Accidents 2000-2011

Noteworthy of the above are the facts that 7 “Failed to Follow the Correct Procedure” (Checklist), 4 “Failed to Prioritise Attention” and 11 (at least) were during manual flight control. In addition, and this has to a be a subjective judgement, the accidents in bold print are considered to be due in large measure, to operator remoteness. In fact, all of the accidents above were either whilst flying manually (which is difficult anyway due to remoteness) or due to pilot remoteness anyway being a major factor. The conclusion must be reached that remoteness is a major issue for the accident rate.

The conjecture raised by the above observations is that, if we move to fully automatically controlled flight **and** mitigate in some way the effects of pilot remoteness, we should be able to substantially reduce the number of accidents caused by unsafe acts.

In terms of Phase of Flight, the following table, which reinforces the prevalence of Landing accidents, summarises them:

Phase of Flight	Number	Notes
Taxi	1	
Take Off	1	
Climb	1	
Cruise	7	Two under Manual Control
Descent	2	
Approach and Landing	7	All under Manual Control
Total	19	

Table 14: Unsafe Act Accidents by Phase of Flight

4.2.6 A Summary of Accident Rates

Evidence was presented earlier that the accident rate for GA was about 6-7 per 100,000 flight hours. Large commercial aircraft are about 0.01 per 1000, 000 flights hours. The USAF[86] have produced the following tables for Global Hawk, Reaper and Predator:

	CLASS A		CLASS B		DESTROY			CUM
YEAR	#	RATE	#	RATE	A/C	RATE	HOURS	HOURS
FY99	1	375.94	0	0	1	375.94	266	**266**
FY00	1	221.73	0	0	0	0.00	451	**717**
FY01	0	0.00	0	0.00	0	0.00	486	1203
FY02	2	127.71	0	0.00	2	127.71	1566	2769
FY03	0	0.00	0	0.00	0	0.00	779	3548
FY04	0	0.00	0	0.00	0	0.00	1375	4923
FY05	0	0.00	1	34.99	0	0.00	2858	7781
FY06	0	0.00	0	0.00	0	0.00	3568	11349
FY07	0	0.00	0	0.00	0	0.00	5972	17321
FY08	0	0.00	0	0.00	0	0.00	6634	23955
FY09	1	13.75	0	0.00	0	0.00	7274	31229
FY10	0	0.00	0	0.00	0	0.00	8322	39551
5 YR AVG	**0.2**	**3.15**	**0.0**	**0.00**	**0.0**	**0.00**	**6354.0**	
10 YR AVG	**0.3**	**7.73**	**0.1**	**2.58**	**0.2**	**5.15**	**3883.4**	
LIFETIME	**5**	**12.64**	**1**	**2.53**	**3**	**7.59**	**39551**	

UPDATED 29-NOV-10

Figure 22: RQ004 *Global Hawk* Mishap History

	CLASS A		CLASS B		DESTROY			Cum
YEAR	#	RATE	#	RATE	A/C	RATE	HOURS *	HOURS
FY04	0	0.00	0	0.00	0	0.00	71	71
FY05	0	#DIV/0!	0	#DIV/0!	0	#DIV/0!	0	71
FY06	2	64.04	0	0.00	1	32.02	3123	3194
FY07	1	14.58	0	0.00	0	0.00	6857	10051
FY08	3	28.91	0	0.00	1	9.64	10378	20429
FY09	4	15.75	0	0.00	1	3.94	25391	45820
FY10	1	1.78	0	0.00	1	1.78	56109	101929
5 YR AVG	**2.2**	**10.80**	**0.0**	**0.00**	**0.8**	**3.93**	**20371.6**	
LIFETIME	**11**	**10.79**	**0**	**0.00**	**4**	**3.92**	**101,929**	

UPDATED 29-NOV-10

Figure 23: Q9 *Reaper* Mishap History

[86] At http://www.afsc.af.mil/organizations/aviation/aircraftstatistics/index.asp accessed on 10 July 2011

	CLASS A		CLASS B		DESTROY			Cum
YEAR	#	RATE	#	RATE	A/C	RATE	HOURS *	HOURS
FY97	2	75.27	0	0.00	2	75.27	2657	2657
FY98	0	0.00	0	0.00	0	0.00	3258	5915
FY99	2	38.95	0	0.00	2	38.95	5135	11050
FY00	1	15.56	1	15.56	1	15.56	6426	17476
FY01	4	52.83	1	13.21	4	52.83	7571	25047
FY02	7	36.25	0	0.00	6	31.07	19313	44360
FY03	2	9.75	0	0.00	2	9.75	20507	64867
FY04	6	19.12	0	0.00	5	15.93	31383	96250
FY05	10	24.38	1	2.44	9	21.94	41024	137274
FY06	5	8.65	0	0.00	3	5.19	57798	195072
FY07	7	8.84	0	0.00	5	6.31	79193	274265
FY08	10	6.76	3	2.03	9	6.08	147989	422254
FY09	13	6.94	4	2.13	10	5.34	187393	609647
FY10	7	3.40	3	1.46	6	2.91	206113	815760
5 YR AVG	8.4	6.19	2.0	1.47	6.6	4.86	135697.2	
10 YR AVG	14.4	18.04	2.4	3.01	11.8	14.78	79828.4	
LIFETIME	76	9.32	13	1.59	64	7.85	815,760	

UPDATED 29-NOV-10

Figure 24: RQ001 *Predator* Mishap History

The following points are noted:

- *Global Hawk* already has an excellent record despite losing 4 aircraft in its early prototype days. Big improvements in reliability and design have made a large contribution to this but at increased costs. Global Hawk cannot be flown manually.

- *Reaper* is newer than the others and its rate is about twice as much as GA. It has more automation than *Predator* but can still be flown manually.

- *Predator* is the most mature but also the least sophisticated and its current rate is about the same as GA.

4.3 A Bayesian Model for UAS Accidents

4.3.1 Introduction

It has already been noted that that humans not only cause accidents but, by use of their special skills in reasoning and experience, plus their ability to successfully react to unforeseen and complex situations, they also are capable of preventing the propagation of errors and circumstances that could ultimately result in an accident. Evidence exists [43], which indicates that for every 300 incidents that could result in aircraft accident, humans can and do prevent 290 of these with a further 9 leading to a major incident and only 1 actually resulting in an accident. However, as has been highlighted, operator remoteness in a UAS, as compared to a manned aircraft, is likely to degrade that performance. The evidence presented in the accident analysis above, whilst acknowledged to be fairly approximate, would tend to indicate a reduction of about 50% in that performance.

4.3.2 Requirements, Limitations and Assumptions

In developing a model to help identify, understand, and improve the decision architecture of UASs, the accident model had the following requirements, limitations and assumptions:

- The model should only encompassed human and machine control errors – reliability modelling was not addressed.

- The model should reflect the fact that humans directly in the loop do prevent accidents.
- The model should reflect the accident evidence presented above.
- The model should be based on the Reason Model of Human Error using the HFACS taxonomy. Organisational and supervisory errors are not included at this stage.

In order to understand the contributions made by humans alone and to reflect the 1 in 300 accident evidence presented above, a human error model was developed first. It should be noted that no attempt has yet been made to analyse the relative probabilities for the underlying causes – they are all currently set to be equal for each area. However, the probabilities of the possible outcomes do reflect the evidence presented in Para. 4

4.3.3 Human Error Bayesian Model for Aircraft Accidents

The human error model for aircraft accidents is shown below and indicates the relative probabilities for an accident given that a major incident has occurred. This model reflects the evidence and the requirements stated earlier.:

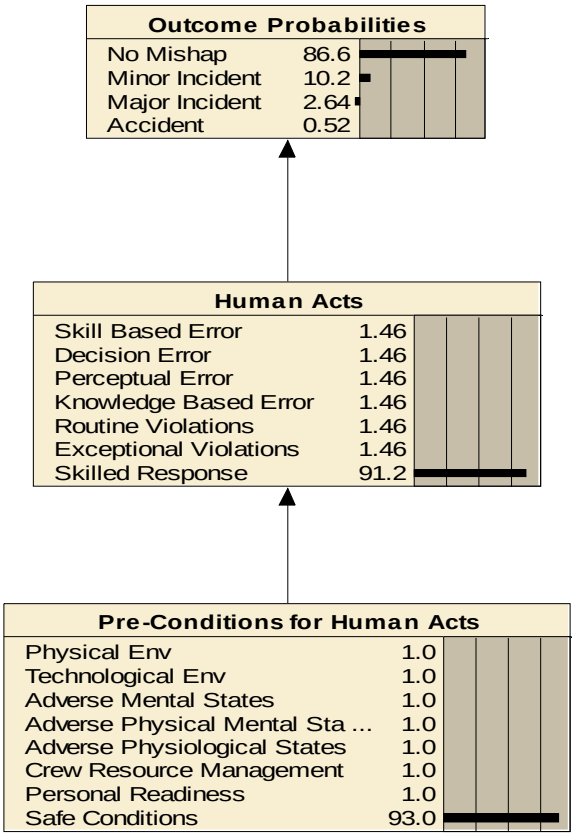


Figure 25: Human Error Bayesian Model for Aircraft Accidents

4.3.4 The Preliminary Model for UAS Accidents

The Human Error model was then degraded to cater for operator remoteness, extended to include machine acts and the outcome probabilities modified to reflect the accidents rates in Para. 4.2. The four control operation types, RPV, RPV with automatic take-off and landing, fully automatic and semi-autonomous, were added and subjective assessments made to determine relative outcomes. Again it should be noted that this is a preliminary model and indicates relative probabilities for an accident given that a major incident has occurred. The model is shown below:

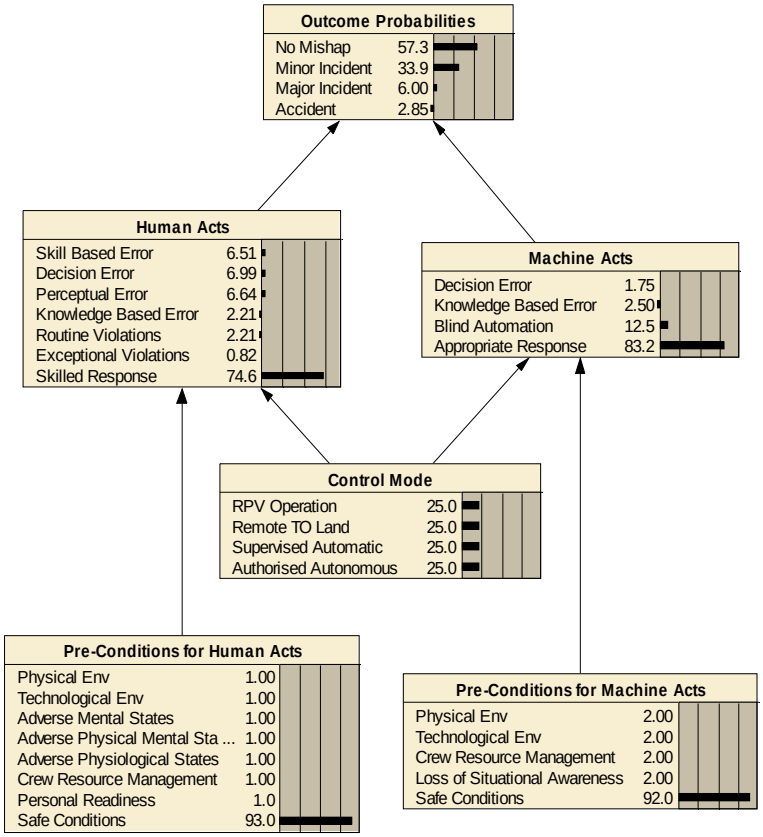


Figure 26: Preliminary UAS Accident Model

The variation according to the different operation types is shown in the following four diagrams:

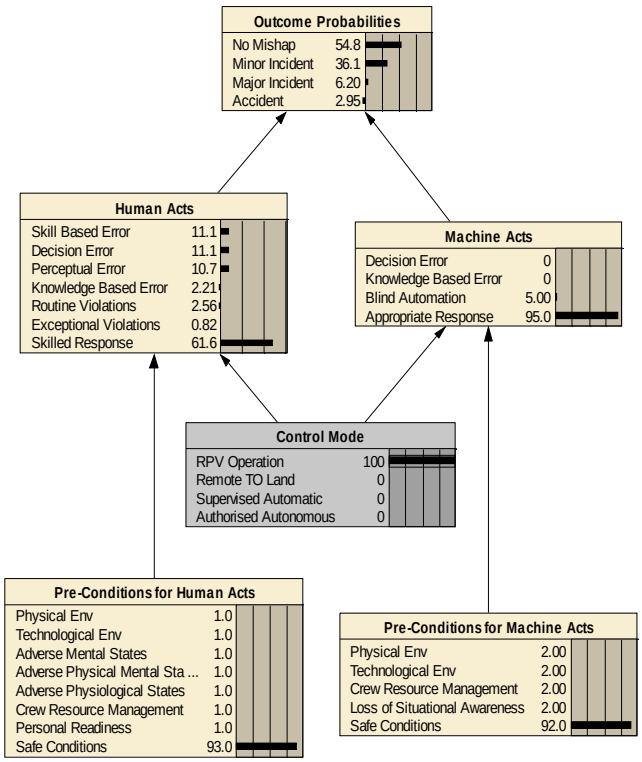


Figure 27: Remote Pilot Vehicle Operation

The RPV operation model results reflect the facts that skill based, perceptual and decision errors predominate for human acts whilst decision and knowledge based errors are zero for machine acts.

For RPV operation with automated take-off and landing, there is a reduction in skill based errors but an increase in “blind automation” – the acts of a machine which blindly follow an inappropriate control sequence such as continuing a take-off when it would be better to abort. The results for this operation type are shown below:

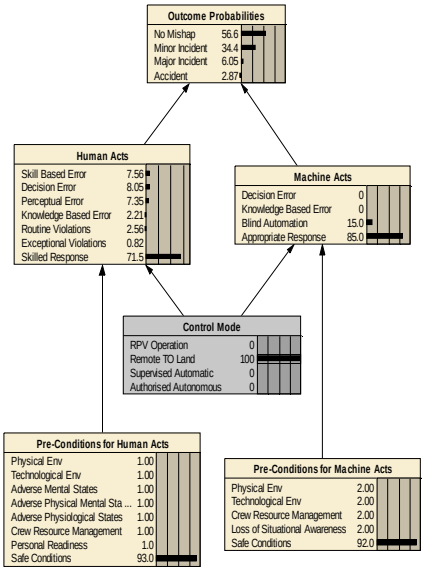


Figure 28: RPV Operation with Automated Take-off and Landing

For highly automated UAS types, such as Global Hawk, the model is further modified showing a reduction in skill based error and increase in automation errors. This shown below:

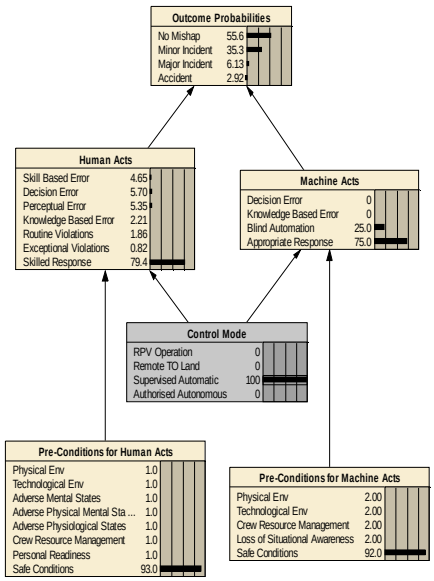


Figure 29: Fully Automated Operation

For autonomous operation, decision and knowledge based errors in machine acts are introduced. Those of humans are reduced accordingly. The model for this operation is shown below:

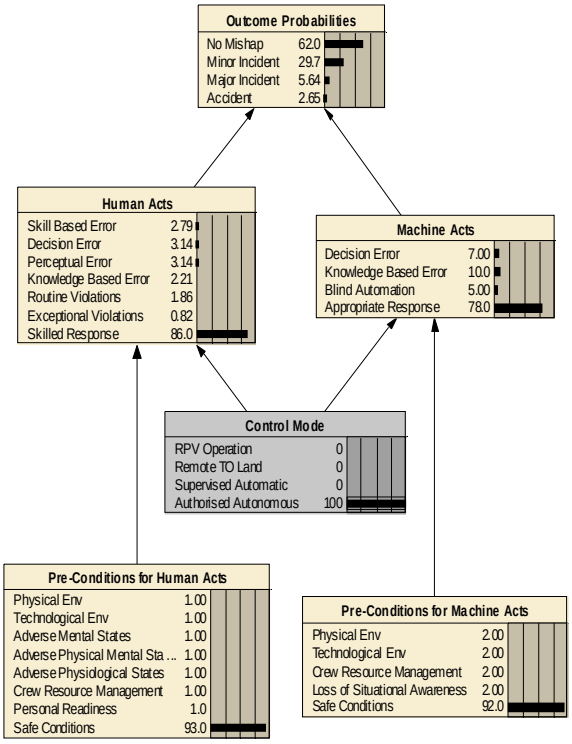


Figure 30: Autonomous Operation

4.4 Results and Inferences

The main thrust for the work undertaken to date is to understand how the decision architecture for autonomous operation is affected by the need to shift control acts from the human operator to the machine whilst maintaining, or preferably reducing, the accident rate. Whilst noting that this work is on-going, the initial results indicate the following:

- Introducing machine decision making, in itself, will increase the accident rate.
- Inevitable increases in machine decision errors in autonomous system operation must be matched or bettered by reductions in human errors,

particularly skill, perceptual and decision errors.

- Human errors can be reduced by increasing human situational awareness (SA). This in turn may be achieved by improving the dialogue from the machine to the human, cross-checking apparently irrational commands.
- Machine decision errors and “blind automation” errors must be reduced to a minimum. This can be achieved by substantially increasing machine SA. Again, this could be used to cross-check human commands and improve human SA.
- Useful gains can be made by reducing machine knowledge errors.

4.5 Conclusions and Future Work

A preliminary model of the causes of accidents for UASs of differing control operation has been presented. This model already indicates where improvements in the decision architecture must be made if the benefits of autonomous operation, namely workload reduction, are to be realised without a commensurate increase in the accident rate.

It was originally intended to quantify these to show a theoretical level of safety that can be achieved. Completion of the research however, indicates that it is unwise to state such a quantification. The reasons for this are various. Primarily, it was hoped that the quantative nature of the accident analysis using Bayesian methods would facilitate such an approach. However, accident analysts, fairly universally, suggest that no two accidents are the same and that hindsight is very much a wonderful thing. It is precisely because no two accidents have the same causal chain that the HFACS approach finds multiple causes for each accident and as such the probability of their occurrence never, or at least rarely, adds up to 1, thereby negating a probabilistic approach. Wonderful hindsight can give a false illusion that, from the investigation of past accidents, we could predict accidents in detail. If we could do that, we could design against every future accident which we clearly cannot. All that can be done is to investigate, assess our vulnerability to such a causal chain that has been learned from

experience and put in place safeguards to negate or reduce the probability of history repeating itself. And that is what has been proposed here.

Accident researchers, particularly those investigating UAS accidents [40], have commented that the statistics are very much vehicle specific. A particular design flaw in the HMI for *Predator* is assessed to be culpable for many accidents but that flaw does not read across to other vehicles where the design is different.

5 Strategies and Implementations for UAS Safety Improvements

5.1 Safety Improvements to the Proposed Architecture

In the previous Chapter, the conjecture was raised that we move to fully automatically controlled flight in all phases of flight **and** mitigate in some way the effects of pilot remoteness leading to unsafe acts, we should be able to substantially reduce the number of accidents.

When the operator and autonomous system are separated, there is scope for having separated situation awareness and many examples are captured in Table 13: HFACS Unsafe Acts-Predator/Reaper Accidents 2000-2011. If this separated situation awareness differs, perhaps by only a small but critical factor, there is increased opportunity for an accident trajectory to progress. Since autonomous systems are data driven, any method of restoring situational awareness to a common and consistent level will reduce the probability of that trajectory progressing through to an accident.

To generate that consistency requires:

- effective communication between operator and AS of each other's intent, situational state and belief set
- identification of belief set and state differences
- assessment of the potential consequences of a lack of consistency
- recovery action needed to re-align the difference

Recognising that the above needs to be achieved with a minimal increases in bandwidth and preferably none, ways in which the above can be realised using existing dialogue must be researched. Some of these are identified below:

5.1.1 The SA Abduction Loop

There are several models of Situational Awareness (SA) but very few ways of measuring it effectively. One method is that proposed by McGuinness and Dawson [44] and is called the Quantitative Analysis of Situational Awareness

(QUASA). The basis for this method is as follows. If a person believes something in his environment to be true and it is, then that is good SA. Similarly, if something is believed false and it is, it also good SA. Conversely, If something is believed to be true and it is false, then that is bad SA and so on. This can be summarised as:

Scenario Facts Beliefs	True	False
True	High SA	Low SA
False	Low SA	High SA

Table 15: QUASA SA Beliefs

The overall SA metric is formed by summating those environmental aspects and their results, true/false, as follows:

$$SA = (\sum (T/T + F/F)) - (\sum (T/F + F/T))$$

The QUASA concept can now be used for shared SA. In our case, we actually do not know the absolute truth, only our individual beliefs, man and machine. However, what we are interested in is shared SA. In other words, what do we agree to be true/false and what do we disagree to be true/false.

Thus the basis becomes now:

Operator Beliefs Machine Beliefs	True	False
True	High Shared SA	Low Shared SA
False	Low Shared SA	High Shared SA

Table 16: SA Shared Beliefs

If the beliefs have different levels of importance, weightings (W) can be applied to generate the metric as follows:

$$\text{Shared SA} = (\sum (T/T + F/F) * W1) - (\sum (T/F + F/T) * W2))$$

The problem now translates to one of aligning these beliefs. The operator should be able to find out what the machine believes to be true/false (and, ideally, degrees in between) if his GCS is set up to highlight these. How though does the machine discover the operators beliefs? Well, firstly these can be explicitly passed by the operator but this may prove to be cumbersome. Another way suggested here is by abduction and an example is given to illustrate this based on a real UAV accident[87]. In this accident, the pilot thought the aircraft was on the ground, and as it was in difficulty, shut down both engines to stop it. Unfortunately, it was, in fact, airborne. Using this example, the process to be followed is:

- Command received –"Shut down both engines". This is flight safety critical and would attract a high weighting.
- A database of all checklists is scanned with the result that this command is only normally used when the vehicle is on the ground as part of the post flight shut-down checks
- Therefore, **abduct** that the operator's belief is that the vehicle is on the ground.
- Compare with own belief and generate the metric as detailed above.
- If the metric is below a set threshold, the command is queried.

It is interesting to note, that if such a process had been in place for the Predator/Reaper accidents described earlier, particularly Numbers 2, 11, 12, 14, 16, it is entirely possible they may not have progressed through to becoming an accident. The abduction and belief checking would represent a new barrier in

[87] *"Crashed after take-off from N'Dolo. Apparently the pilot was afraid the aircraft would not get airborne in time before the end of the runway, and he switched off both engines, but as the aircraft was already in the air it crashed a few hundred metres after the end of the runway on Boulevard Triomphal. One woman on the ground was killed and at least three others were injured".* Widely reported by quoted here from *http://belmilac.wetpaint.com/page/IAI+-+Eagle+B-Hunter+UAV+(Unmanned+Aerial+Vehicle) accessed 8 July 2011*

the Swiss Cheese model to the final unsafe act by the remote operator. Note that such a system is unnecessary, or at least far less important, in a manned aircraft.

5.1.2 Critical Belief Set Handling

So what beliefs are important to either cross check, monitor or abduct from? Earlier the rule-set for autonomous machine operation of the vehicle was presented. All of these are in the form:

IF (belief + certainty) THEN (decision, action)

The belief that leads directly to a rule being fired into action can be considered a critical belief as without it being present the behaviour would be absent. This belief that leads to a control function (or in the AIMS architecture, a plan) can then be assessed for safety criticality, impact or involvement as a contributory cause. Such an assessment procedure is described below.

5.1.3 Hazard Analysis

In addition to any extra, and for autonomous systems, novel, mitigations that may be employed to reduce the likelihood of an unsafe act, a conventional Hazard Analysis should be adopted as a matter of course. To demonstrate this, it is useful to introduce a non-exhaustive List of Hazards as an example, including their severity and possible mitigations. Note that these are linked by the regulations for certification set out in Table 7.

Hazard	Severity	Mitigation	Short Label
Collision with another aerial object	Catastrophic	Safety-critical aerial collision avoidance	Air collision
UAS collides with terrain or ground-based obstacle	Catastrophic	Safety-critical ground collision avoidance	Ground collision
UAS Flies into Dangerous Weather Conditions	Hazardous	Weather Detection and Avoidance Routing	Weather

Hazard	Severity	Mitigation	Short Label
Engine(s) Failure	Hazardous	Twin Engine or High Engine Reliability	Power Failure
UAS runs out of fuel and consequently collides with terrain	Hazardous	Safety-critical fuel control	Fuel depletion
Failure to lower undercarriage before landing	Major	Safety-critical control	Gear not lowered
Attempted take off when not aligned on appropriate runway	Hazardous	Safety-critical control	Unaligned take off
Loss of flight control	Hazardous	Safety-critical control	Loss of control
Collision with airfield objects or static/taxiing aircraft	Major	Safety-critical control	Airfield ground collision
Running engine hazard to ground crew	Major	Ground operating procedures	Running engine hazard to ground crew
Failure to inform operator	Minor	Autonomy of control is expected to determine safe action without operator comms.	Failure to inform operator
Failure to follow the Rules Of the Air (ROA)	Major	Reduction of safety margin will not of itself cause an accident	Failure to follow ROA

Table 17: List of Identified Hazards

Several methods are available for a Hazard Analysis and they are all routinely done in systems design. One is Fault Tree Analysis whereby the probability of failure of an item is assessed and the consequences of that failure is analysed. A good example of this is given in Chapter 6. Such an analysis is useful when failure rates or probabilities of hardware components are available (it is of no use for software). Another method is Safety Case Assessment, whereby consideration is given to probable scenarios and an analysis of likely outcomes in the event of certain circumstances is made. When a formal system design is

under consideration, a common analysis tool is a Functional Failure Analysis (FFA). This considers end event functions[88]. If these are deemed to have an effect on safety, an analysis is performed by considering the consequences when the :

- Function is not provided when demanded
- Function is provided when not demanded
- Function that is provided is wrong or in error.

If the consequence or effect is unsafe in any of the above categories, a mitigation is required. This could be to have a back-up function always available (standby altimeter, compass etc.), to improve the integrity of the function by reducing the probability of the effect occurring to the appropriate hazard level (as discussed in Para. 4.1), or to have cross monitoring or "common sense" applied, usually by the pilot. As will now be obvious, if the pilot is remote, that monitoring may well be not be done effectively and the consequences lost or misunderstood. Just as importantly, the automation applied may well be hiding the output of the function, in which case a similar outcome will exist. In the case of an autonomous system being in control, it is obvious that such a control system will have to perform the monitoring and mitigation normally done by the pilot. This can be done by introducing Plausibility Checking.

5.1.4 Plausibility Checking

Plausibility checking is a simpler alternative to continuous monitoring. It involves assembling separate facts to form a belief that can be directly compared to belief state under consideration. The key aspect is that the facts assembled must be independent in every way to the variable under consideration. A good example, and one which will explained in more detail in the next Chapter, is fuel quantity. Fuel quantity is a critical belief (for powered flight at least). Fuel quantity is provided by fuel gauges of which there may be several. These are

[88] An end event function is one that is delivered externally i.e. to the environment, from the system under consideration.

not normally duplicated in a manned aircraft; it is usually left to the pilot to assess their veracity. In an unmanned autonomous aircraft, where critical mission decisions such as routing, alternate diversion options, flight altitude etc. need to be made, it is essential that an equivalent to the pilot function is provided. This can be achieved in several ways and one is presented here. By the passage of time and a knowledge of fuel flow, or look up table for average consumption per minute, an assessment of approximate fuel quantity can be made. All of these variables are separate to the gauges and are therefore suitable for plausibility checking. In formal terms such a check forms an AND gate in the fault tree and therefore the probabilities of the fuel gauge failing AND the plausibility check being wrong can be multiplied. This will provide a lower probability of overall system failure than that of the fuel gauge alone.

In short, for manned aircraft, we take the plausibility checking and cross monitoring function normally undertaken by the pilot, for granted. It is not known whether these functions have been introduced to date for current unmanned automated UAS[89] but future autonomous aircraft will certainly have to have them and if so, it is likely that their accident rate will improve over that achieved to date.

5.2 Human Machine Interaction for Safe Autonomous Systems Operation

A recurrent theme throughout this thesis is the relationship between the system and the Human and although a complete review and analysis of the HMI for autonomous systems is beyond the scope of this work, there are certain important aspects which must be addressed into order to complete the picture. It has already been seen in the accident analysis that the HMI, in itself, can cause accidents due to confusion and misinterpretation. A move to autonomous system operation will require a fundamental change to how such systems are managed in order to release the potential of these systems whilst at the same

[89] I think this is highly unlikely for reasons formulated in the next paragraph.

time ensuring that such systems will have an improved safety record. Before this though, a brief review of those aspects covered so far is given:

5.2.1 A Review of Control Modes for UASs

The following control modes for UASs in general were previously presented and the general aspect of their nature highlighted:

- **Remotely Piloted** – The pilot is in direct (immediate and constant) control of the vehicle. This mode of operation will not be discussed further because, as noted earlier, the trend is towards increasing the level of automatically piloted flight due to the need for Beyond Line of Sight operation (no immediate control feedback available) and to drive down (human) costs.

- **Supervised Automation** – The pilot is a remote supervisor of the automation which is directly controlling the vehicle.

- **Autonomous Operation** – The pilot is a remote manager of the autonomous system which is supervising the automatic operation of the vehicle, where authorised.

These latter two modes will now be discussed in some detail in order to examine the differences, pros and cons.

5.2.1.1 Supervised Automation

Increasing levels of automation, supervised by the pilot in manned aircraft, has been the overwhelming trend over the last 10-20 years for cockpit design. The fact that humans are psychologically ill equipped for this (supervision) task, is noted by virtually every textbook on Human Factors.

This automation has been introduced into the manned aircraft cockpit in successive levels [45] each one of which tends to further remove the pilot from the basic control loop. This is not in itself a bad thing. The manned aircraft accident rate has been slowly but surely reducing over many years and some of

that, or perhaps a lot, can be attributed to increased automation [46]. Certainly automatic systems are more reliable, more accurate and faster than humans. However, there are problems. One of these is called the “Ironies of Automation” introduced by Bainbridge in 1987 [47]. She notes that:

- Systems designers, who regard humans as unreliable and inefficient, strive to replace them with automation. In doing so they make a significant contribution to accidents. This particularly manifest when the designer is not an expert – particularly relevant to the air environment.

- Design engineers tend to automate those functions which are simple and easy to do, not those that are complex and which the pilots need most. In short, the designer still leaves functions to be handled by humans because they cannot think of an effective way of automating them

Norman [48] makes the same point*: “automatic equipment seems to function best when the workload is light and the task routine: when the task requires assistance, when the workload is highest, this is when the automatic equipment is of least assistance—this is the 'irony' of automation”* .

Unfortunately cockpit automatic systems, such the Airbus Flight Management System (FMS), as noted in [46], possess no intuition, no intelligence, no discernment and no decision capacity. In fact the majority of such systems comply with the Principles of Occam’s Razor [29], they are limited, by design, in their knowledge to the control at hand. As such they do not have, “The Big Picture”, which means that their operation is not always appropriate. In short they are not generally “Situationally Aware” and rely completely for this aspect on the pilot, who, as we have noted, is sometimes out of the loop, especially in times of high workload, stress or problem fixation. As will be seen later, to have such awareness is fundamental to an autonomous system. As Norman [48]. says:

"The problem, I suggest, is that the automation is at an intermediate level of intelligence, powerful enough to take over control that used to be done by people, but not powerful enough to handle all abnormalities. Moreover, its level of intelligence is insufficient to provide the continual, appropriate feedback that occurs naturally among human operators. This is the source of the current difficulties. To solve this problem, the automation should either be made less intelligent or more so, but the current level is quite inappropriate".

The above comments also point out that automation is fine when everything is going well and is routine. When the "abnormalities" or emergencies occur, a situation that may well be not expected by the designers, the pilots have to cope. This situation is common in aircraft accidents as indicated in the previous Chapter. The conclusion to draw from this is any on-board system should also be capable of handling, or at least providing support, in emergency or abnormal situations. From the *Predator* accident analysis, there were several incidents of procedures not being followed correctly. An on-board system to provide assistance in such situations must therefore be considered highly beneficial.

5.2.1.2 Autonomous Operation

The details of autonomous operation and the fact that the operator is now in a managerial role, perhaps controlling a "team" of several vehicles, has already been made. The primary means of interfacing with the proposed decision architecture is by authorisation of proposed or modified plans and through them, setting objectives[90], either at a microscopic or macroscopic level. By operating at the plan level, there becomes a need for the human manager to be aware of intent of the autonomous system. The concept of intention is at the heart, or some would say, the consequence, of the decision process. Decisions, followed by intentions, are commitments to act, either now or in the future. These intentions are acted upon when the appropriate time or triggering event

[90] Plans can be simply seen as a sequence of one or more actions to either achieve, or maintain, a goal state. That goal state may be specified as an objective.

occurs[91]. Because the actions of the vehicle are specified by these intentions, a clear and unambiguous display of them to the manager is fundamentally required. In addition to intentions, the feedback requirements, identified by Norman above, of what the automation is doing is also required. Finally, it is highly important that the manager, like any other manager of human resources, needs to have confidence and trust in the system to which he has authorised the tasks.

5.2.2 Displays for Autonomous Air Systems

In Para. 2.3.5, it was noted that new models of HMI and appropriate design guidelines will be required to be developed for autonomous air systems. These can now be considered and some principles developed but first it is instructive to consider the transition from manned cockpits in the air, to control stations on the ground.

5.2.2.1 Current UAS Workstations

Shown below is the operator interface of the most advanced UAS currently flying, which of course is an example of supervised automation, *Global Hawk.*

Figure 31: Global Hawk Operator Interface

[91] In the morning, you may make a decision to have dinner that evening. Thus you **intend** to have dinner. When you have it may be a time, say 19.30, an elapse of time, say 30 minutes after arriving home, or a particular event, say when you are hungry, when it has been heated etc. The decision was made in the morning, the intention remains until it is either retracted or triggered.

It can be seen that this display bears every similarity with that of an aircraft cockpit despite the fact that the operator cannot manually fly it. This gives, of course, the illusion that the operator has flight control. This illusion is reinforced by the fact that the crews wear flight overalls (usually with wings) and are frequently referred to as pilots[92]. This concept of a "cockpit on the ground" reinforces the other illusion referred to earlier that the pilot is "in effect" in the vehicle. When it is considered that delays on the satellite link may be of the order of seconds, it could be quite dangerous to believe that the displays an operator is seeing are in real time, when they clearly are not[93]. There have been proposals to take this illusion to the extreme by redesigning the workstations in order to more fully immerse the operator. A picture of such a "cockpit" is shown below (note the control column).

Figure 32: Raytheon Immersive Cockpit

[92] I don't dispute the fact that in the RAF and USAF, they are, in fact, qualified pilots

[93] During a recent Sense and Avoid Trial, the "pilots" watched the TCAS display like hawks until it was pointed out that the display was several seconds in arrears. They were under the illusion that it was a real time display because that what was they were used to.

It is contended that current, and proposed as above, operator workstations are not only unsatisfactory to operate automated aircraft and may well be dangerous under certain circumstances, They are unquestionably unsuited for autonomous vehicles, even for those operating at the lowest PACT levels.

5.2.2.2 Concepts of Autonomous System HMI Displays

As noted before, the primary needs are to show intention of decision, the status of conferred authorisation, the current (critical) states of the vehicle, particularly with respect to the automation, and finally the likely future states of the vehicle. With these requirements in mind, a concept display for discussion is shown below.

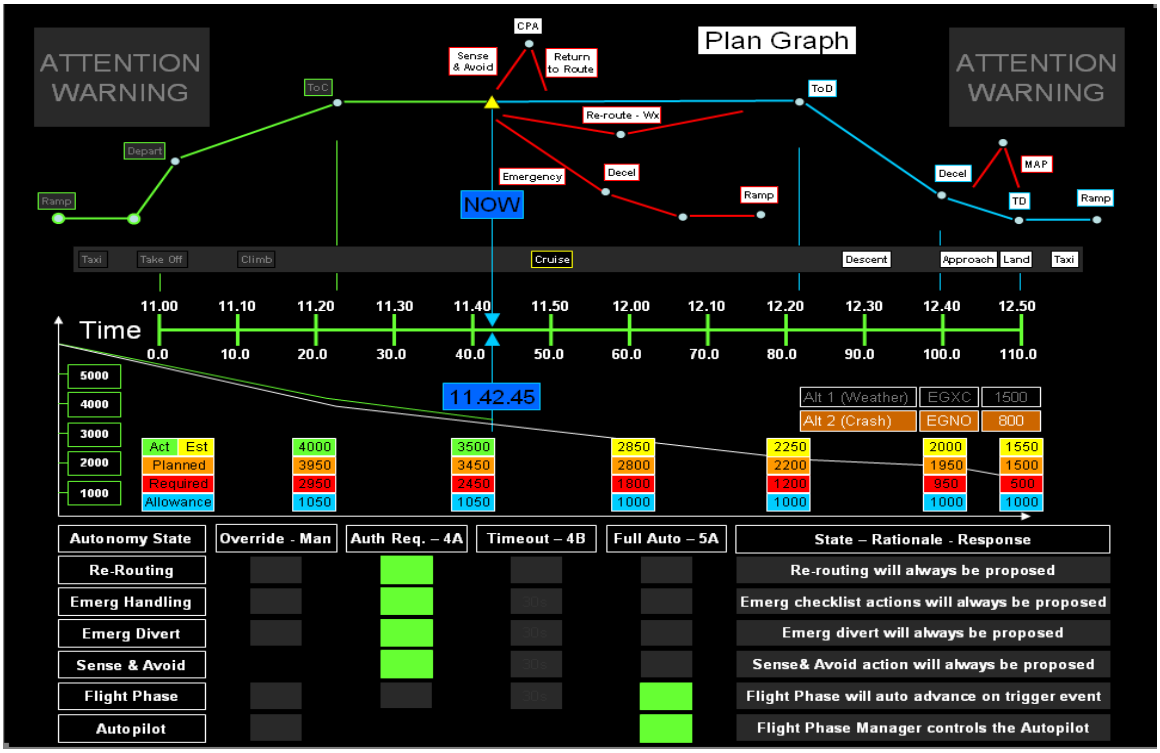


Figure 33: Concept Display for an Autonomous UAS

This is in fact three displays. A Plan – Intent Graph at the top showing past, current and future plans/intentions, a fuel graph showing current and predicted fuel states together with fuel requirements and reserves in the middle, with both joined by a common timeline. At the bottom is the status of the major planning and authorising functions in terms of their PACT level. These will now be explained in more detail.

The Plan – Intent graph is expanded below:

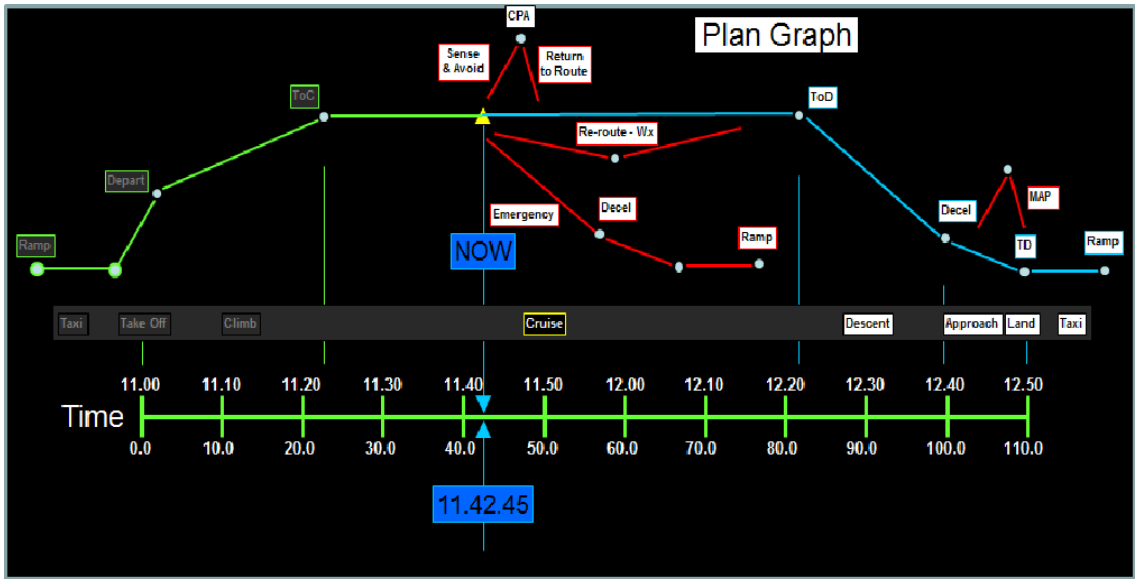


Figure 34: Concept Plan - Intent Graph

The time line at the bottom shows the take-off time (11:00) and predicted landing time (12:50) together with elapsed mission time (0 -110 minutes). Time NOW (11:42.45) joins the plan-intent graph with the fuel graph (not shown here). In the middle are the past (greyed out), current (yellow outline) and future (white boxes) Phases of Flight. The transition times of these phases are shown by vertical lines connecting the Plan-Intent graph to the timeline (green for past times and blue for future times). The Plan-Intent graph at the top shows the key events (and their times). The current plan is shown in green (past) and blue (future intent). The red lines show alternate plans that may be invoked. If they had in the past (perhaps on resumption of communication) they would be shown as green and if they are intended in the future or now, in blue. This display covers past actions and future intentions of the autonomous vehicle as required above.

The fuel graph is more traditional but nonetheless important and is shown below.

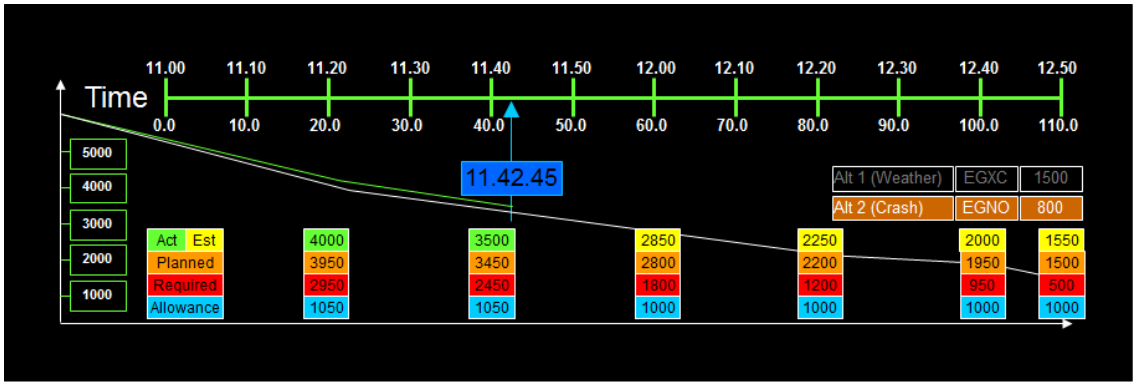


Figure 35: Concept Fuel Graph

Actual fuel used is shown on the green line and can be easily compared to that predicted before flight (white line). This particular graph shows that fuel is being used at about the predicted rate, indicated by the slope, (therefore no fuel leak) but is above that predicted for the current time (perhaps faster progress, more favourable winds or different air data). The coloured boxes, every 20 minutes (a normal fuel check point), show the actual (green) or estimated (yellow) fuel at those points, the planned fuel (brown), the required fuel (red) to complete the planned mission (which includes contingencies) and the allowance (blue), which is the difference between actual/estimated and required. The planned alternates[94] for landing, together with their fuel requirements are also shown, with the current selection highlighted in brown. This graph is a real time graph and will show updates if a different route is flown or different diversions are selected.

The major planning functions and their PACT level are shown in the graph below.

[94] Aircraft usually plan on having two alternates landing airfields. One is in case the weather at the destination falls below landing limits, sometimes called the weather diversion. The other is in case an aircraft crashes at the destination airfield thus rendering the runway unavailable.

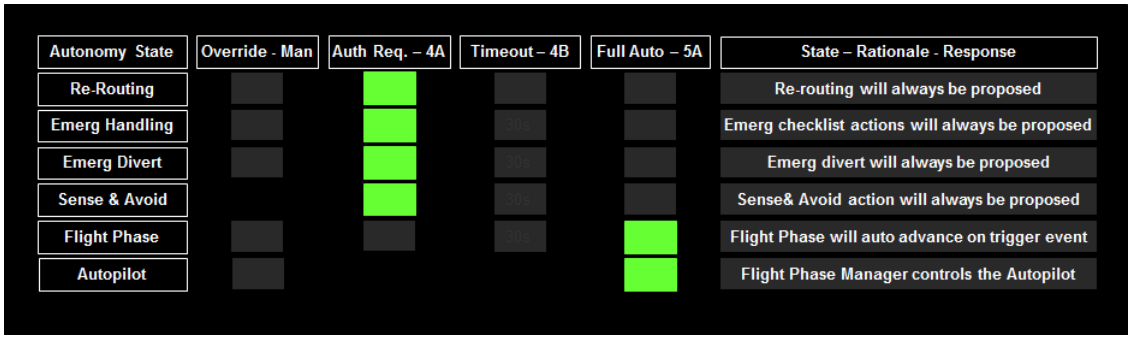


Figure 36: Concept PACT Graph

The concept PACT graph show the status and authorisation levels of the major planning areas, routing, emergency handling, diversion, Sense & Avoid, Flight Phase and Autopilot. On the right is an explanation of the status whilst in the centre are a range of buttons/captions indicating the state (grey indicates available, green indicates selected (by either the operator, or for Sense and Avoid, by the operator or Master Executive), black is unavailable). This particular graph shows that the autopilot control is being handled by the Master Executive at PACT 5A and that the Master Executive will transition the flight phase plans (also at 5A). The other functions at PACT4A are as stated and the Master Executive will only propose plans for these functions.

These displays when combined and used in conjunction with more conventional displays meet the requirements outlined above and should make a positive contribution to safe operation of the vehicle.

5.3 Summary of Strategies

To summarise, the strategies that can be assembled in order to reduce the accident rate of UASs and which must be employed in autonomous UASs, the following is proposed:

- All flight control functions should be automated with no manual control[95]. These flight control functions will then have to be implemented as a safety critical system in order to be certificated. In doing this, care must

[95] This has been done for the Global Hawk UAS. However, it is not known whether the controls have been implemented as a safety critical system.

be given towards the exact implementation. Directly linking the flight controls to a sensing system can be dangerous[96]

- Provide weather detection and avoidance mechanisms fully integrated within the decision architecture.
- Provide better fuel management procedures. These should be integrated with the overall flight management and routing planning centres.
- Provide a more efficient means of handling emergencies, particularly checklists and formal procedures.
- Introduce a range of cross monitoring, plausibility checking and abduction feedback mechanisms
- Provide improved HMI to show/provide :
    - autonomous system intent
    - increased feedback of aircraft state, particularly those deemed critical, either to the on-board autonomous system and/or the GCS.
    - status of authority for the key control mechanisms

In order to check the above and assess whether there are further mechanisms that need to be considered to improve safety, some exemplar scenarios are now analysed in detail.

[96] A tragic but exemplar case is that of Air France 447 whose air data sensors gave out conflicting signals and caused the flight control system to enter a steep descent from which it never recovered. In addition, and as will be seen, flight controls only have a limited awareness of the world around them and will quite happily fly into a hill if they have been so instructed.

6 Exemplar Scenarios Applied to the Proposed Improvements

In order to confirm, or otherwise, that the model findings and whether its inferred solutions are necessary and sufficient to support the thesis, the following exemplar scenarios were analysed and conclusions drawn:

1. Sense & Avoid
2. Mission Management
3. Take Off and Landing
4. Lost Link (Communications) Handling
5. Emergency Handling

6.1 Sense and Avoid

In a manned aircraft, the pilot has many responsibilities to other air users. Among the more primary of these is the so called “See and Avoid” requirement. This requires a pilot to make a constant and vigilant watch for other aircraft and, if safe and/or regulated separation with another vehicle(s) is in danger of being violated, to safely manoeuvre, according to the Rules of the Air, to increase separation to the safe or required level. That this is a prime requirement of any air vehicle is obvious, however despite this there are some 10 mid-air collisions, primarily involving GA aircraft, usually in clear skies, in daylight, in the USA each year; the majority proving fatal[38].

There have been 177 mid-air collisions between GA aircraft in the USA over the last 10 years which accumulated 200million flying hours[38]. This indicates that the probability of a mid-air collision, at least for GA aircraft, is approximately 8 x 10^{-7} per flying hour. The number of mid-air collisions for other aviation types (airliner, military etc.) is so small as to be statistically insignificant in comparison, so the quoted figure can be reasonably used as an input to the overall accident rate. Of course this figure includes the failed mitigation strategy of each pilot be able to first see, and then avoid, the collision (or else they would

not have collided). The baseline collision risk probability without mitigation is clearly higher. That is:

$$Prob_{\text{Mid-air Collision}} = Prob_{\text{Collision Risk}} * Prob_{\text{Safe Separation Failure}} * Prob_{\text{Mitigation Failure –Pilot A}} * Prob_{\text{Mitigation Failure- Pilot B}}$$

In developing the Traffic Collision Avoidance System (TCAS) control logic, the methodology called for generation of an Safety Encounter Model. Such models describe the probability of a mid-air collision. These outputs from these models are highly dependent on airspace type, aircraft type and traffic densities. Consequently, different models are used in the USA and Europe. The European model is a substantial revision of a model specified by ICAO and is described at [49]. The Encounter Models are usually modified to reflect real data. A generalised output of the European model is presented in the table below. This only gives an approximate view on safety encounter probabilities, but it is consistent with other encounter models and is suitable for consideration for the required level of detail here.

	vmd	hmd	rate
serious loss of separation	500ft	2NM	3 10^{-5} per flying-hour
critical airmiss	400ft	1500ft	3 10^{-6} per flying-hour
NMAC	100ft	500ft	3 10^{-7} per flying-hour
collision			3 10^{-8} per flying-hour

Table 18: Various Expressions of the Assumed Collision Risk without ACAS*[50]*

The figures in the above table are broadly in line with that of the actual GA accident rate due to mid-air collisions presented above.

There are many mechanisms employed to reduce the probability of a mid-air collision, such as the Traffic Collision Avoidance System (TCAS), Air Traffic Control (ATC) surveillance and monitoring, etc. however none of these devolve the air user from meeting the above basic requirement in any way.

Clearly a UAS has to have an equivalent function in order to satisfy regulations and this is commonly called “Sense and Avoid[97].

6.1.1 Application of the Decision Architecture to the Sense and Avoid Scenario

Collision avoidance in manned aircraft comprises two basic mechanisms: identification that a collision will occur, and a subsequent manoeuvre initiated to generate safe separation, either vertically, horizontally or a combination of both. Collision identification can be achieved by a variety of processes and players, such as:

- Air Traffic Controllers when one or both aircraft are under their control or receiving advisory reports. For area controllers, their ATC consoles have a variety of automated devices alerting them to the possibility of a conflict and highlight the affected aircraft on the displays. When aircraft are in Controlled Airspace, the ATC controller holds primary responsibility for collision avoidance. The same is true for ATC controllers in the tower at airfields. He will advise the positions of aircraft in his visual (and sometimes radar) circuit.
- The pilots in either aircraft may identify that a collision possibility is occurring. This may be visual or they may have equipment to help identify this such as TCAS.

For unmanned aircraft, the process is no different except that the pilot is now remote. He therefore cannot use his eyesight directly, but will rely on sensors. These can be of many types such as Electro Optics, radar, datalink and data broadcasts. For UASs with conventional architectures, the pilots must maintain a vigilant search using these sensors for other air objects and respond appropriately. Using the proposed architecture should confer several benefits, most of which will decrease the chance of a mid-air collision. The processes that the architecture will allow are presented in the schematic below.

[97] There is an increasing trend to rename this function to be “Detect and Avoid”.

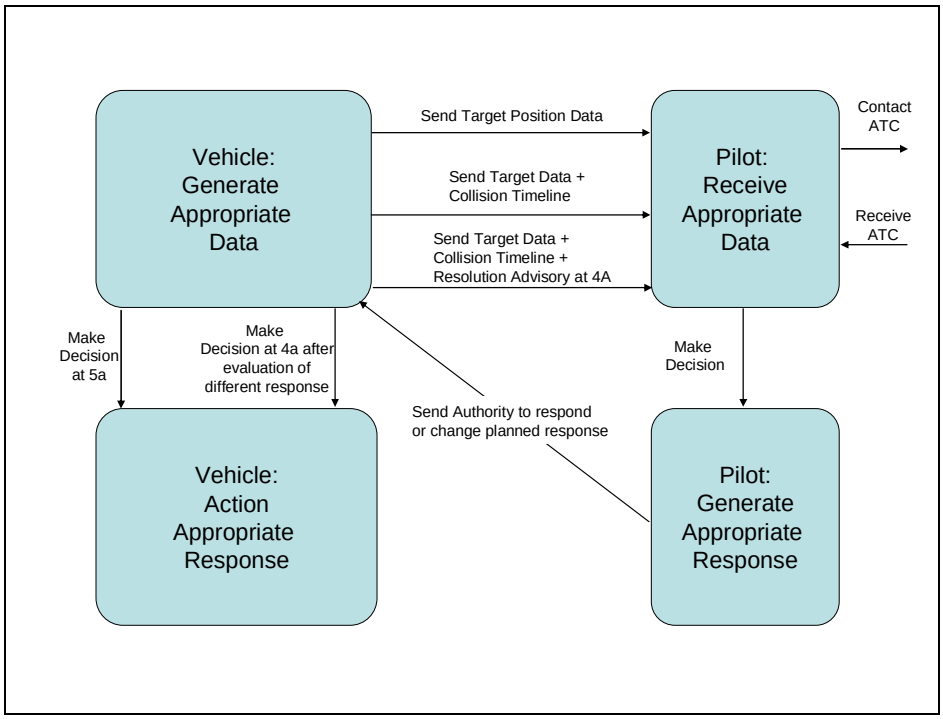


Figure 37: Collision Avoidance Process

The process can be described in terms of a timeline as follows:

- The Master Executive (ME) can be pre-authorised (at level 5a) to respond to Sense and Avoid resolution manoeuvres. This would be advisable if the expected false alarm rate was low. This would be outside the vicinity of airfields and if there was a low rate of manoeuvring. Such pre-authorisation would allow extremely fast response times (this is discussed in detail later). If the ME is authorised at level 5a, it would manoeuvre according to the plan generated by the Sense and Avoid sub-system planner following receipt of the conflict advisory message.
- When at level 4a, on receipt of a conflict advisory, the ME would report this using the communications link to the pilot by either sending:
    - intruder data
    - additional data including a collision timeline
    - manoeuvre plans for the resolution of the conflict
- The pilot would consider his response. If his aircraft is in controlled airspace, and there is plenty of time, he may wish to report this to the

ATC controller (who has the responsibility for collision avoidance. He can then either do nothing or send:

- authority to manoeuvre
- order a modified manoeuvre

- The UAS receives the message via the communications link and actions the response.
- The aircraft then manoeuvres to avoid the intruder achieving the appropriate safe separation.

This timeline can be modelled according to the following schematic:

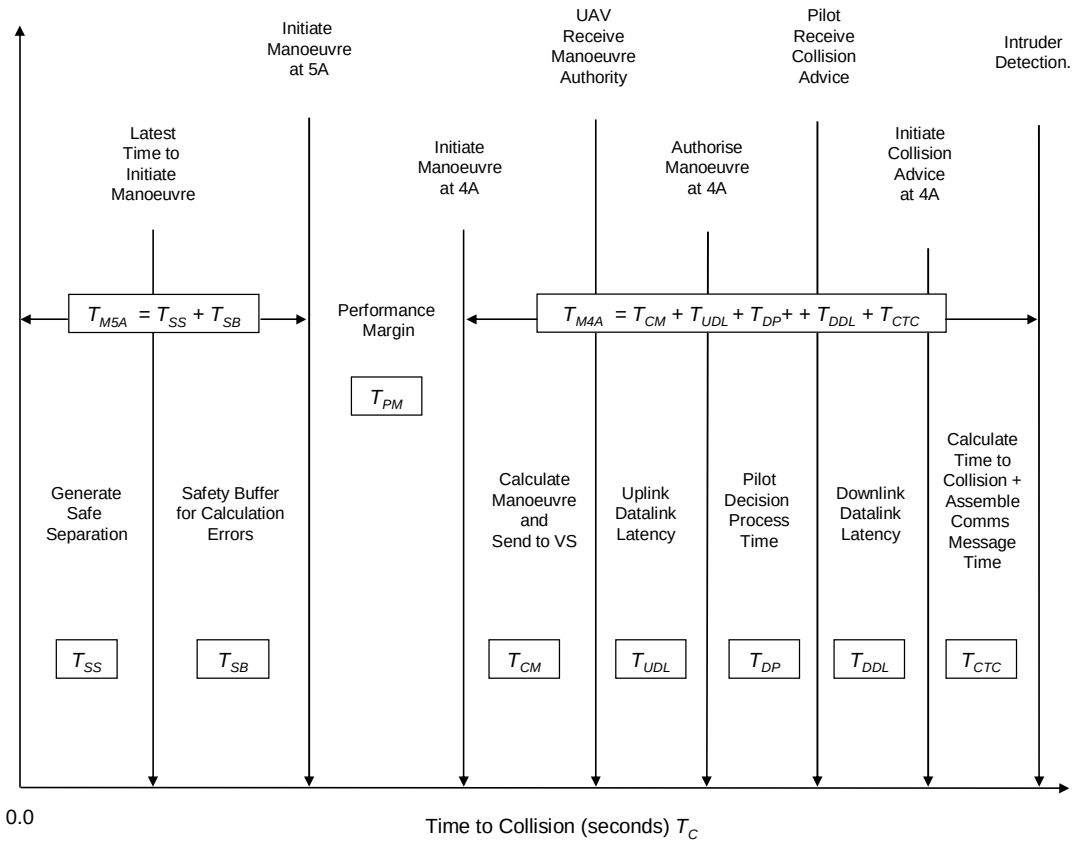


Figure 38: Collision Timeline Model

It can be seen that if T_{M4A} is larger than T_{M5A} then T_{PM} will be negative. This means that the time to include the pilot in the loop is larger than that required to generate a safe separation. In such a situation, the ME will wait until the last

safe time to act and then respond at Level 5A, by invoking an escalation of the authority levels as follows:

- Request authority to manoeuvre (at Level 4A)
- If no reply has been received by X seconds to manoeuvre, request authority to manoeuvre (at Level 4B with a time-out of X seconds).
- If no reply has been received by the end of the time-out, escalate the authority to manoeuvre to 5A
- Manoeuvre at 5A to avoid the conflict

The above is an example of ME authorisation escalation due to time pressure.

We can rearrange the formulas in the schematic above to calculate T_{PM} as follows:

$$T_{PM} = T_{TC} - T_{CTC} - T_{DDL} - T_{DP} - T_{UDL} - T_{CM} - T_{SB} - T_{SS}$$

where T_{TC} is the initial time to collision.

In practical terms, it is likely that $T_{DDL} = T_{UDL}$[98].

The above formula was modelled in Matlab using the following terms:

Time to Collision T_{TC} This is calculated using:

$$T_{TC} = \frac{DetectionRange}{Closing Velocity}$$

where Detection Range is modelled as a random Gaussian distribution with a Mean Detection Range and a standard deviation of 500m. The mean detection ranges used were 7km, 8km and 9km. These cover typical sensor ranges for electro-optic and millimetric radar sensors. The distribution for 8km is shown below:

[98] The justification for asserting that "$T_{DDL} = T_{UDL}$" is that the causes for delays to the uplink is quite likely to be the same as that for the downlink and therefore cause a similar delay.

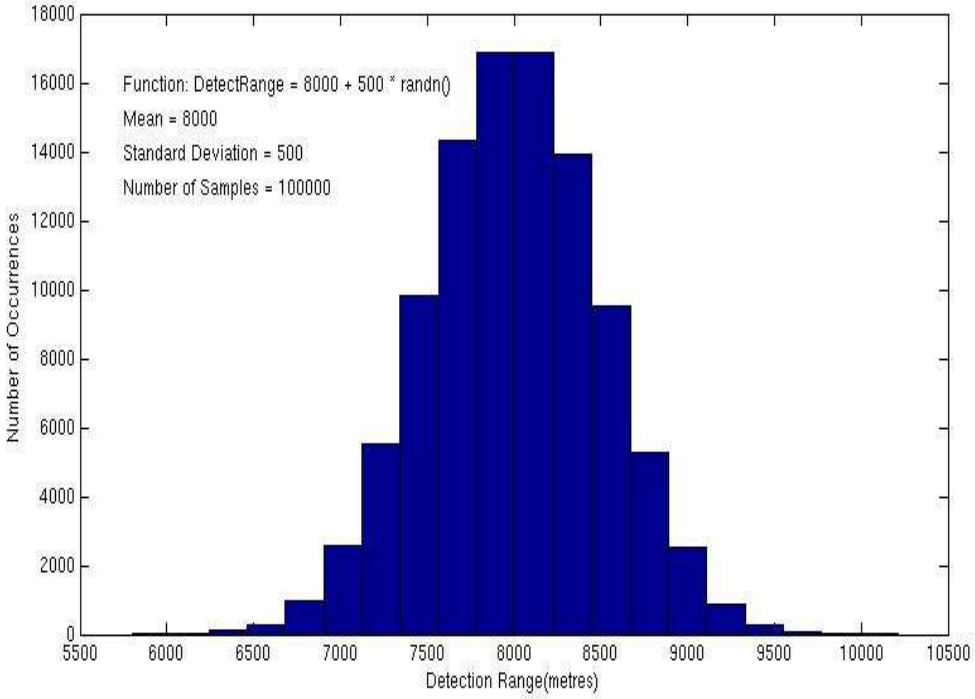


Figure 39: Gaussian Detection Range Model (Mean 8Km)

Closing Velocity is the vector sum of intruder/own aircraft velocities. These velocities were modelled as random Gaussian distributions with a mean of 250m/s and a standard distribution of 30m/s. This is modified to take into account the Track Crossing Angle[99] (TCA) which is modelled as a Gaussian random variable between $\pm$ 30 degrees of own aircraft's nose. The resulting closing velocity when TCA is taken into account has a mean of 240 m/s. This is shown on the figure below:

[99] The Track Crossing Angle is the difference in headings between two aircraft. It is conventionally described as TCA = 0 for the head on collision case, and TCA = 180 for the tail-chase scenario.

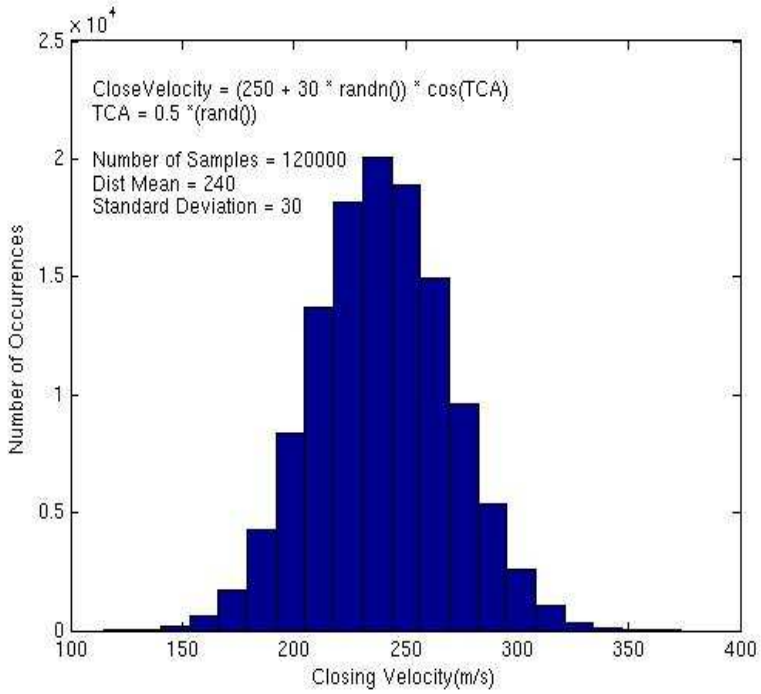


Figure 40: Closing Velocity Model

The closing velocity modelled thus represents a worst case (in terms of time to collision) Head On collision scenario.

Taking the distributions for Detection Range and Closing Velocity, a distribution of Times to Collision can be calculated. This is shown below:

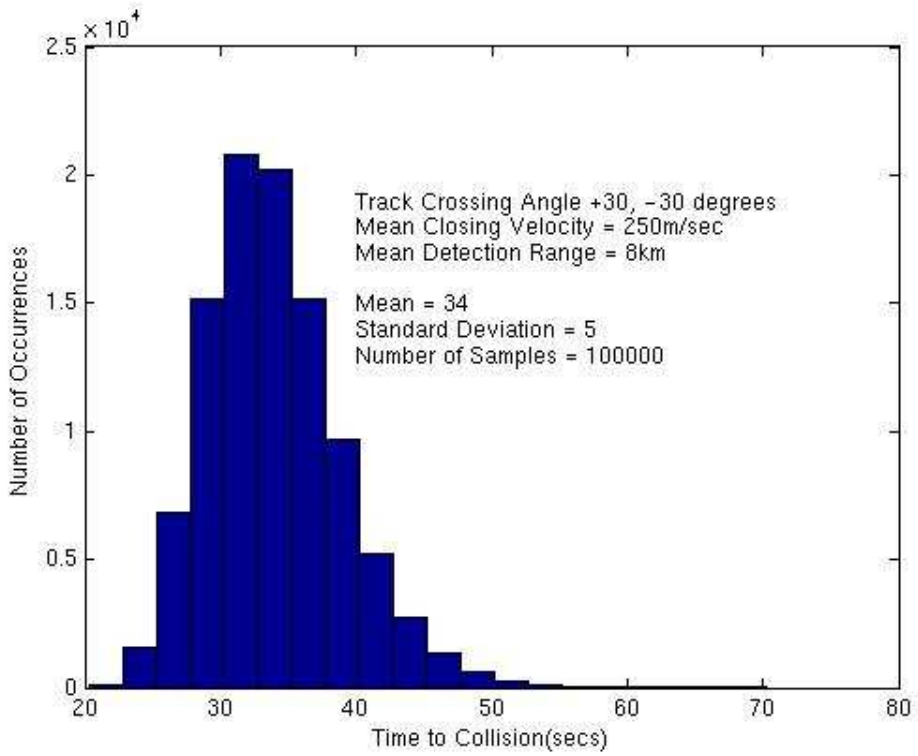


Figure 41: Distribution of Modelled Time to Collision

- **Time to Compute and Communicate, T_{CTC}, T_{CM}** In the AIMS architecture, the time from initial detection to assembling the communication message for transmission can be calculated as follows: For computing the initial detection, calculating the collision geometry and generating the communication request for authorisation, there are seven virtual channels connecting the seven computing applications all of which are running at 10Hz. It seems reasonable to infer that this should take a least a minimum of 0.7 seconds. Similarly, on receipt of the authorisation, there are six virtual channels between six computing applications before the autopilot can be set to initiate the manoeuvre. Again, these applications are all running at 10Hz. Accordingly, T_{CTC} was set at 0.7secs and T_{CTC} set at 0.6 secs.
- **Downlink /Uplink Delay Time, T_{DDL} , T_{UDL}** A full (or even partial) discourse on the delays, or latencies, involved in UAS communications is beyond the scope of the thesis. The main contributors it appears are not the physical aspects but rather the system aspects[51] such as: Line of

Sight Vs. Beyond Line of Sight (Sat-comms) topologies, number of jumps between satellites, distance of receiver/transmitter from satellite's nadir, geo-stationary Vs. low earth orbit, and many others. For satellite communications, used when the UAS is Beyond Line of Sight, then in terms of typical values, a generally held view[51] was that a figure of the order of 1-2 seconds was appropriate. In support of this, Eurocontrol conducted a study to generate Communications Operating Concept and Requirements (COCR) for the Future Radio System (FRS)[52]. A table which details the acceptable latency for the Air Traffic Service (ATS) is reproduced from that document below. This shows that the one-way acceptable latency value at the 95 percentile is 1.2 seconds.

Service & Phase	Service Type	Confidentiality	Latency (sec)					Integrity FRS	Availability Of Provision FRS
			APT	TMA	ENR	ORP	AOA		
ATS Phase 1	Broadcast	Medium	0.4	1.2	1.2	1.2	-	5.0E-8	0.9999975
	Addressed	Medium	3.8	3.8	3.8	26.5	-	5.0E-6	0.9995
ATS Phase 2	Broadcast	Medium	0.4	1.2	1.2	1.2	1.2	5.0E-8	0.9999975
	Addressed	Medium	1.4	0.74	0.74	5.9	1.4	5.0E-10	0.9999999995
AOC 1+2	Addressed	Medium	13.60	13.60	13.60	26.50	26.5	5.0E-10	0.9995

Table 19: Most Stringent Future Radio System Allocated Data Requirements

As only a general conclusion was required as an output of the model, the latency times, T_{DDL} and $T_{UDL,}$ were modelled as a random Poisson distribution[100] which gives the probability mass function of the expected arrival of the signal with a mean of 1.2 seconds.

[100] The justification for using this distribution is based on the study conducted by Eurocontrol to define acceptable latency requirements for a variety of communication systems. To quote" For most services, the COCR assumes a statistical allocation of latency based on a Poisson distribution" taken from Communications Operating Concept and Requirements for the Future Radio System (COCR) Ver 2.0

Safety Buffer Time, T_{SB} The Safety Buffer Time is inserted into the calculation to cater for the fact that it imprudent to wait until the last possible millisecond to start generating safe separation due to:

- the fact that the dynamics of the calculation may not be stable over time
- the input sensor data will have errors in it
- the calculations may also have errors due to rounding and approximations
- aircraft have to roll in the direction of manoeuvring before they can start generating lateral separation

The Safety Buffer Time used in the modelling was fixed at 5s.

Safe Separation Time, T_{SS} The Safe Separation Time is the time in which the aircraft can generate sufficient separation to avoid collision. The Safe Separation Time used in the modelling was the time for an aircraft to complete a 45degree Rate One[101] turn viz. 15s.

- **Pilot Decision Process Time, T_{DP}** The time for the pilot to make a decision and respond is based on evidence generated in deriving the ACAS Performance Model. This model has the safety encounter model, modified by a pilot response model. The ACAS Control logic assumes that the pilot will react to an ACAS generated Resolution Advisory (RA) within 5 seconds by commanding an acceleration of 0.25g to achieve the required 1500 fpm vertical velocity. For our model, it seems reasonable to suppose that a ground based pilot would not be any dissimilar in his performance, and arguments can be made for it to be worse, based on the fact that his situational awareness would be poorer due to his remoteness. Initially therefore, the model used a value of T_{DP} = 5s. However, evidence was found that, in order to validate this figure, the EUROCONTROL ASARP project has analysed on-board data recorded

[101] A Rate One turn is a turn that completes a 2π turn in 2 minutes. This corresponds to 30 degrees of bank angle and is considered to be a normal manoeuvre of an aircraft.

by some contributing European airlines during a period of three years. To support a comparison with the ACAS Control logic standard pilot response, the actual pilot responses were quantified in terms of:

- Time between the issuance of the RA and the beginning of the manoeuvre,
- Vertical acceleration taken to perform the manoeuvre,
- Vertical speed achieved by the manoeuvre.

The data gathered to validate the 5s figure in fact showed that the mean pilot response time was in fact 6.7s and also indicated that on 20% of occasions, no response was made at all. The figure below provides an overall picture of the observed pilot responses and the frequencies for each of the different response types. This distribution defines the ACAS pilot response model. In line with the figure commonly observed for the European airspace, this response model includes a 20% proportion of non-responding pilots. The model makes no mention of why a “no response” occurs. However since the main reason for the RA is to cue the pilots for a visual check, it is entirely possible that the pilots felt that no response was the correct action.

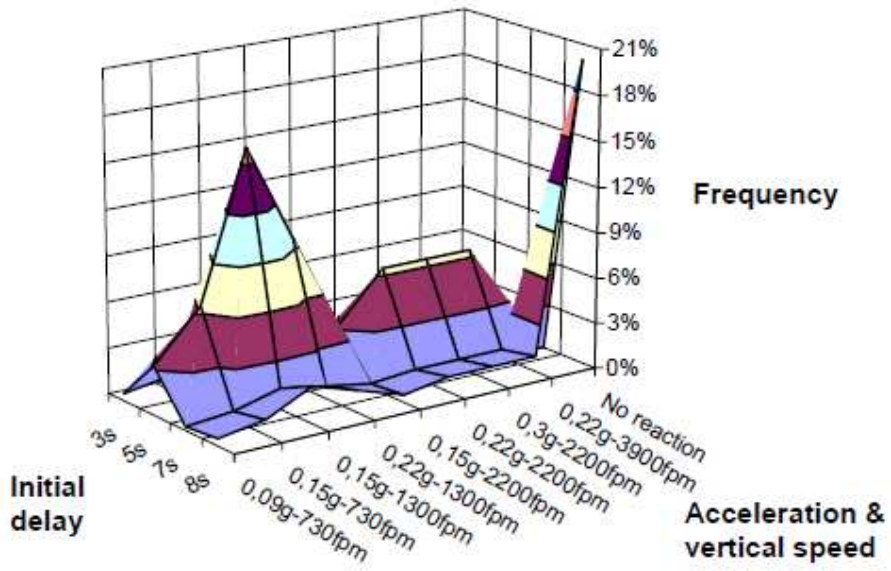


Figure 42: ACAS Pilot Response Model Distribution

In accordance with the above model, and in the absence of any other evidence, the figure used for $\mathbf{T_{DP}}$ was a random Gaussian distribution with a mean of 7s, an SD of 1s. However due to the fact that there were no reasons for a "no response" given above, this aspect was not included in calculating $\mathbf{T_{DP}}$. The calculated $\mathbf{T_{DP}}$ therefore represents a "best" case.

6.1.2 Results for the Sense and Avoid Decision Model

Initially, the basic Performance Margin outcomes were computed and these are shown below for detection ranges of 8km and 10km. Sample sizes were varied from 10,000 – 40,000 and little significant variation in mean or SD was found.

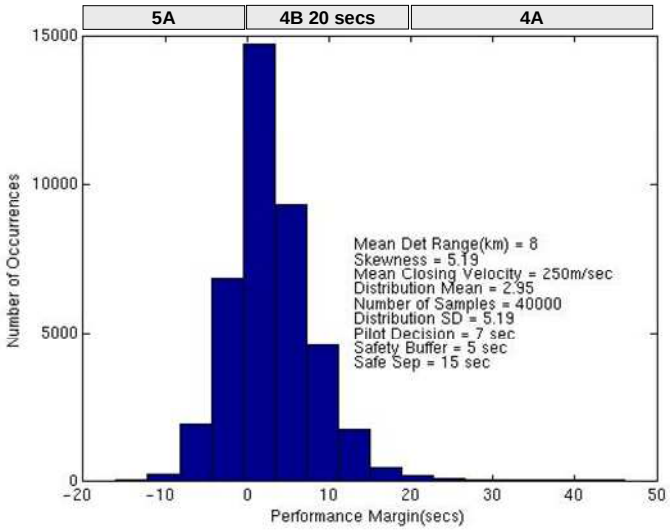


Figure 43: Performance Margin for Detection Range = 10km

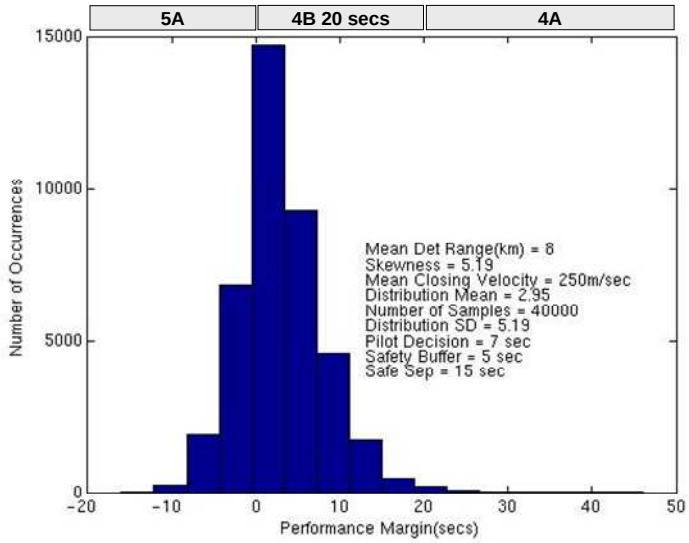


Figure 44: Performance Margin for Detection Range = 8km

Using an average closing velocity of 234m/s (a mean of 250m/s modified by the average TCA) over the difference in detection range between the two models of 2km, the time difference would be 8.5s. The difference between the modelled means for 8km and 10km is 8.1s. The results therefore appear believable.

Experimental Results

In order to further validate the Sense and Avoid Performance Margin model, an experimental system was set up. This used a fully working Sense and Avoid[102] system operating in a Synthetic Environment and integrated with a representative vehicle flight system. The trial scenarios used a head on collision vector (Track Crossing Angle = 180 degrees) with detections according to the sensor model outputs. The system operated according to the timeline as described above and the timings of the dispatch and receipt of authorisations together with the manoeuvre initiation time were logged.

Thirty nine runs were recorded. The raw data collected is shown at Appendix G. Only those runs where full results were logged, the TCA was 180 deg. and a desired CPA of 926m was set were further analysed for this summary – a total

[102] The system was fully representative of a working SAA system. Details of this system are provided at Appendix G.

of 21 runs. These results are also shown at Appendix G and a summary presented below:

Detection Range	Mean = 9676.2 m	Stud Dev. = 924.6 m
Closing Speed	Mean = 250.96 m/sec	Std Dev = 9.38 m/sec
Down/Up Link Latency	Mean = 2.048 sec	Std Dev = 0.805 sec
Notional Pilot Decision Time	5 sec	
Required CPA	926 m	
Achieved CPA	Mean = 926.8 m	Std Dev = 11.55 m
Performance Margin	Mean = +1.97 sec	Std Dev = 3.69 sec

Table 20: SAA Experiment - Parameters for CPA926 runs

The recorded Performance Margins for the 21 runs are shown below:

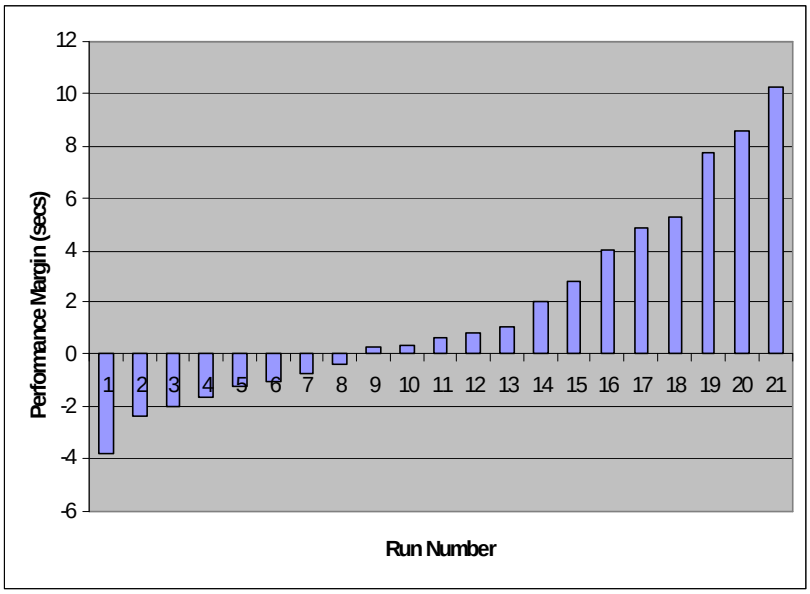


Figure 45: Sense and Avoid Experimental Results (CPA-926)

In order to determine whether these results are consistent with those obtained from the SAA Timeline Matlab model, the Matlab model was modified to use the experimental parameters recorded above. However, it should be noted that the manoeuvre time the model uses, together with the safety margin time, were both notional at 15 and 5 seconds respectively and based on “reasonable” assumptions regarding turning circles. In addition, the model performance

margin is based on the collision point at the time of first detection and assumes no manoeuvring.

In the case of the experimental results though, the manoeuvre initiation time was a function of the algorithms determining the latest time to avoid a collision and of course this varies according to the precise geometry of that particular run. The algorithms also have a notional fixed safety buffer time of 7 sec. In order to rationalise these differences, a spreadsheet model determining the differences between using a fixed collision point and a point of closest approach (CPA) was developed. The spreadsheet results are at Appendix G and a graphical output is shown below:

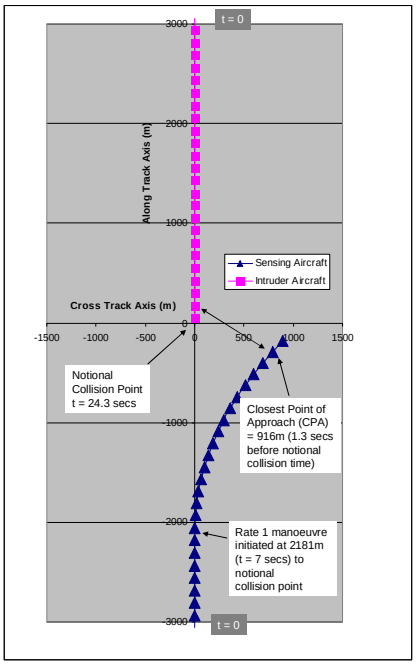


Figure 46: Collision Point and Closest Point of Approach Graph (CPA-926)

Each aircraft (the SAA aircraft and the intruder) is started at time, t = 0 at a range of 3060m from the notional collision point, each with a velocity of 125.6m/sec, which they will reach at, t = 24.3 sec. However, to achieve the desired CPA of approximately 900m, the SAA aircraft commences a Rate One Turn (3 degrees/sec) at time, t = 7.0 sec (17.3 sec to notional collision). The CPA is reached at time, t = 23.0 secs which is 1.3 sec earlier than the notional

collision point. From this we can deduce that the latest time to manoeuvre is 16.0 secs (17.3 – 1.3) from the closest point of approach.

The original Matlab model was therefore further modified to use a Pilot Decision Process Time, T_{DP} of 5.0 sec, a Safety Buffer Time, T_{SB} of 7.0 sec and a Safe Separation Time, T_{SS} of 16.0 sec. Furthermore, the speeds and detection ranges used were as the figures in Table 20. This modified Matlab model gave the following Performance Margin distribution:

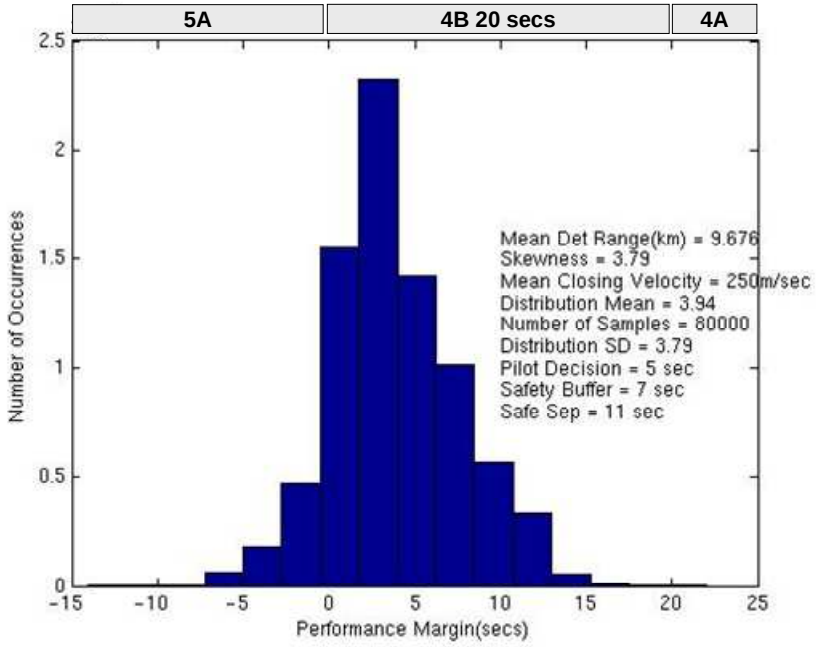


Figure 47: Modelled Performance Margin using Experimentally Derived Inputs (CPA-926)

This can be compared with those achieved experimentally in Table 20 (reproduced below):

Mean = +1.97 sec	Std Dev = 3.69 sec

As a further comparison, 9 experimental results were available with a desired CPA of 463m. The above process was repeated for these runs. and the results are also shown at Appendix G and a summary presented below:

Detection Range	Mean = 8978 m	Std Dev = 1363 m
Closing Speed	Mean = 261.4 m/sec	Std Dev = 3.3 m/sec
Down/Up Link Latency	Mean = 3.0 sec	Std Dev = 0 sec
Notional Pilot Decision Time	5 sec	
Required CPA	463 m	
Achieved CPA	Mean = 463.75 m	Std Dev = 3.45 m
Performance Margin	Mean = +3.0 sec	Std Dev = 5.4 sec

Table 21:SAA Experiment - Parameters for CPA-463 runs

The recorded Performance Margin for these runs is below:

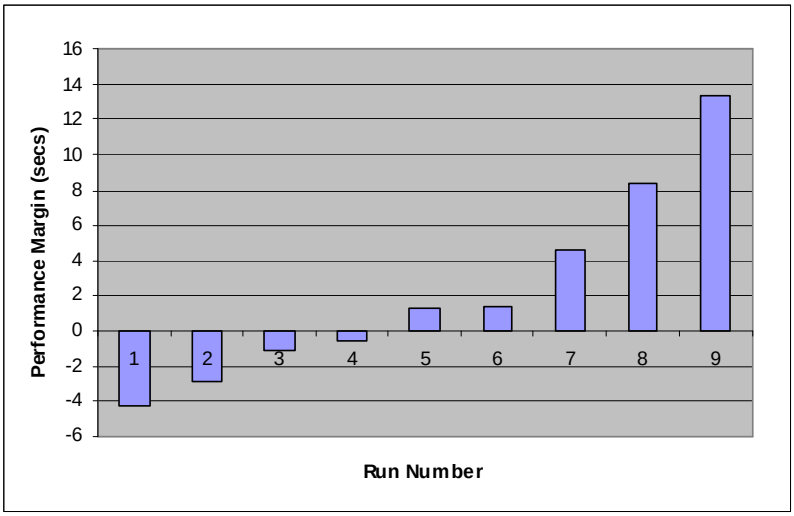


Figure 48: Sense and Avoid Experimental Results (CPA-463)

Again the collision plot was repeated with these figures as shown below:

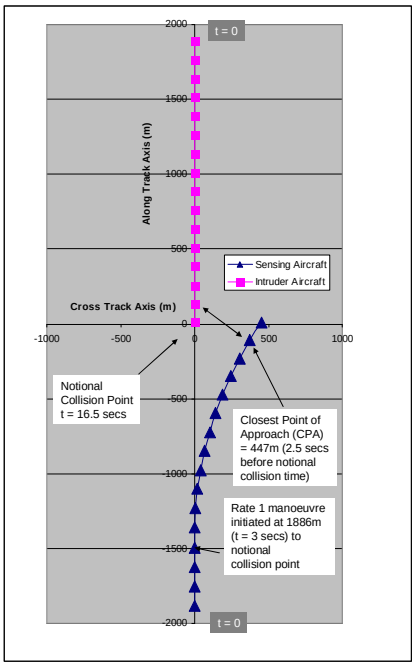


Figure 49: Collision Point and Closest Point of Approach Graph (CPA-463)

Again each aircraft (the SAA aircraft and the intruder) is started at time, t = 0, this time at a range of 1886m from the notional collision point, each with a velocity of 130.7m/sec, which they will reach at, t = 16.5 sec. However, to achieve the desired CPA of approximately 460m, the SAA aircraft commences a Rate One Turn (3 degrees/sec) at time, t = 3.0 sec (13.5 sec to notional collision). The CPA is reached at time, t = 14.0 secs. which is 2.5 sec earlier than the notional collision point. From this we can deduce that the latest time to manoeuvre is 11.0 secs (13.5 – 2.5) from the closest point of approach.

Note that in the first scenario, a 45 degree turn was sufficient to generate the required separation, whilst in this scenario, a turn of 39 degrees was sufficient.

Again the original Matlab model was therefore further modified to use a Safe Separation Time, T_{SS} of 11.0 sec. This modified Matlab model gave the following Performance Margin distribution:

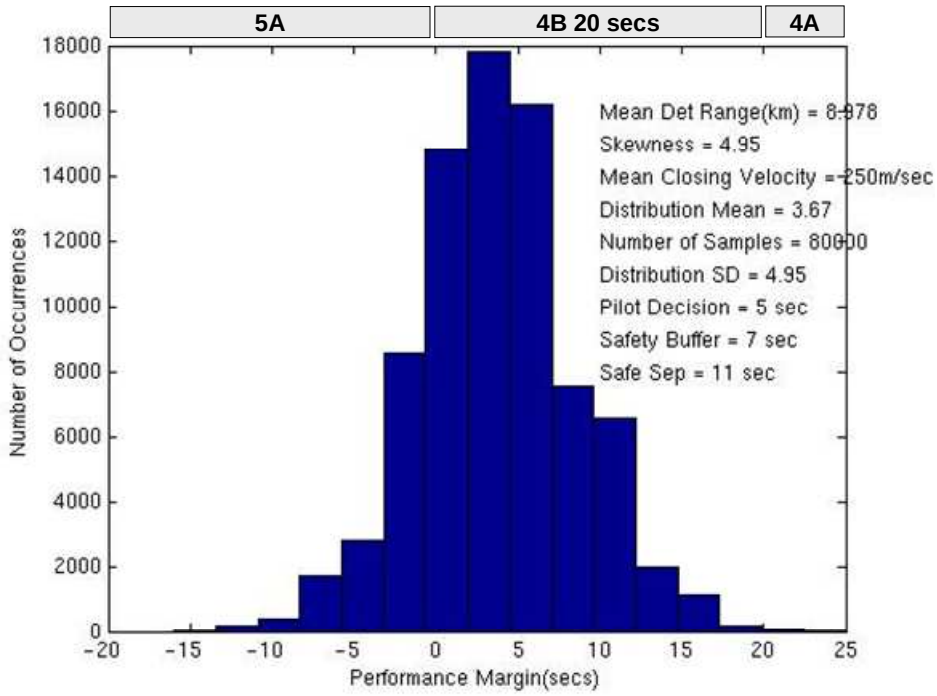


Figure 50: Modelled Performance Margin using Experimentally Derived Inputs (CPA-463)

These results can be compared with those achieved experimentally in Table 21 (reproduced below):

Performance Margin	Mean = +3.0 sec	Std Dev = 5.4 sec

Discussion

Sensor Performance - The first thing that is apparent is to do with sensor performance. The results show that, if the mean detection range of the sensor is less than 8km, then more than 50% of Resolution Advisories will result in the decision architecture operating in an autonomous mode at the PACT Level 5A. There will be literally no point in contacting the pilot as impact will occur before he can respond (at least 50% of the time). As previously noted, a detection range of 8km for millimetric wave radar and electro-optic devices is by no means pessimistic. Such a result indicates a major advantage for this architecture compared to others and highlights the benefit postulated earlier, that the architecture allows human intervention right up to the point when

corrective action must be taken whether the human has intervened or not. It should be noted that this is a fundamental consequence of the architecture's general operation and not a feature applied for a special case.

Distribution Tail Results – The number of samples used was 40,000. A small, but significant number of results gave Performance Margins of less than 20s for the 8Km case. This figure represents a combination of input distributions at their extremities and is such that almost instant reaction is required upon detection. That the architecture can support such a reactive operation again demonstrates its versatility. Again, it should be noted that the architecture is not designed specifically to have a reactive mode that is switched in when appropriate, but rather that the architecture can operate seamlessly from reactive, through to deliberative and finally to subordination to the human controller without any special invocation of a particular control mode.

Comparison with other control/decision architectures:

- Remotely Piloted Vehicles – Notwithstanding the fact that there are no RPVs currently, or expected to be, fielded that are fitted with a S&A system, the limitations previously highlighted clearly show that, given similar detection ranges, 50% of potential collisions cannot be avoided by the pilot in control since he will have insufficient time to respond.
- Automated systems – the only automated system currently fielded that has an S&A system is Global Hawk, which is fitted with TCAS. TCAS has been declared unsuitable for use on many UASs due to the following considerations:
    - It has been primarily designed for use by airliners operating on the airways systems of the world. As such, all the safety studies that were conducted to certify TCAS assumes that the pilot is on-board the vehicle[53] and the Operating Procedures for its use call for the pilot to attempt to make visual contact with the intruding aircraft and manoeuvre according to their judgement (and not the proposed RA message). This is possibly an attempt to reduce the false alarm rate.

- TCAS requires that the intruder aircraft is fitted with either a Mode S or Mode C transponder. Not all aircraft are so equipped.
- The climb rates of UASs and Global Hawk in particular are different from that expected in the TCAS logic. This fact alone has prevented the FAA certifying the use of TCAS on Global Hawk.
- TCAS on Global Hawk has not been integrated with the Flight Control System and is not certified for automatic use.

Moding - When considering the use of automation for a Sense and Avoid scenario, some thought needs to be given to its moding. If automatic responses are required in order to prevent collisions, the equivalent of operation at 5A, then the human must set this prior to any incident, say on climb out or pre-flight. Consequently, he is effectively now out of the loop and cannot intervene unless it is to turn the automation off. If, however, he insists on being in the loop at all times and responses can only be initiated when sanctioned by him, then the overall system will miss responding in 50% of potential collisions (given the above scenario.). Such an automatic system will inherently therefore have limitations in its moding. The autonomous, rather than automatic, decision architecture proposed is the only architecture known that will ensure that the human is kept in the loop until such a point where that situation is no longer beneficial and autonomous action to prevent an accident is initiated.

Assumptions and Constraints – It has been assumed throughout that the Sense and Avoid action, once initiated, will be perfect and prevent the accident. Any other consideration would require a full analysis of Sense and Avoid performance which is outside the scope of this thesis. The main thrust of this analysis is to model the point at which the Information - Decision - Action sequence is initiated, and by which of the two controllers, human or machine, within the general requirement to keep the human controller in the loop for as long as practicable.

Effect on the accident rate – The scenario outlined above is a worst case scenario for a time critical Sense and Avoid solution in that it only considers Track Crossing Angles (TCA) from 150 -180 degrees, which is clearly 1/6th of

the overall range of TCAs (0 -360). It is therefore reasonable to assume that the effect of having the proposed architecture, which is beneficial in 50% of the modelled cases, will be effective in 1/12th (i.e. 8%) of all cases, given that TCAs are equally and randomly distributed. The effect of not responding to a Sense and Avoid action can be regarded as identical to a pilot of an ACAS equipped aircraft not responding to a Resolution Advisory. In the ACAS performance analysis [50], the following is stated:

> "A pilot who never follows ACAS RAs faces a risk of collision that is 45.8%, rather than 27.8%, of that faced by the pilots of unequipped aircraft. **Thus she faces more than one-and a-half times the risk faced by typical pilots, and more than three times the risk she would face if she always followed RAs and followed them accurately".**

Is this a fair comparison? In the absence of some quite complicated encounter modelling, it may well be justified as a best case since ACAS is carried only by large aircraft and modelling assumes encounters typically of this sort. As presented earlier, the accident statistics for GA aviation are about twice as bad as those predicted by the ACAS encounter model. Therefore, it can be argued that initiating a Sense & Avoid manoeuvre will prevent at least 1 in 3 collisions (i.e. 33%) (based on the ACAS encounter model) and, for GA aviation, as much as possibly 1 in 1.5 (i.e. 66%) (based on the GA accident statistics). Since these are rough estimations only, a good working figure of 50% will be used.

6.2 Flight Management

The architecture makes specific provision for the management of each flight. This requires the generation and maintenance of a flight or mission plan[103]. This flight plan is generated by the Mission Planner. This planner updates the mission plan according to progress, generates contingency plans for action in

[103] The noun "Flight Plan" is used to denote the actual plan of the flight in terms where it is going, when it will arrive and how much fuel it will use. It is also used to describe the form (in the UK, Form CA42) that will be submitted to the Air Traffic Management System to either inform them of the flight or request clearance to enter the system. It is the former description that is relevant here.

the event of an emergency and ensures that a safe termination of the flight can always be made. This termination plan includes:

- actions to ensure navigation to land at the designated airfield with more than the minimum authorised fuel on board. This includes an adaptive routing algorithm which enables the aircraft to plan flights around dangerous weather such as thunderstorms and turbulence, which UASs are particularly vulnerable to.
- an algorithm to select the most suitable alternative airfield to land at if required.
- a forced landing sub-system which will identify a suitable site, ensure that it is clear of human inhabitancy and provide suitable guidance to it.
- a safe ditch function.

The above are incorporated into the architecture in order to reduce the probability level of a “Hazardous or potentially “Catastrophic” accident to an acceptable level. The acceptable level of probability of failure at the latter category is of the order of 1×10^{-7} per flight hour.

6.2.1 FLIGHT MANAGEMENT FAILURE DATA

In 2007, there were 90 accidents [38] in GA in the USA attributed to Fuel Management failures. Of these 66 were directly linked to faulty pre-flight planning and in flight monitoring of the fuel. Out of a total of 21.7m flight hours logged, this constitutes a failure rate of 3.1×10^{-6} per flight hour due to fuel mismanagement. There were also 15 accidents due to inadvertently encountering poor weather and icing which represents an accident rate of 7.0×10^{-7} per flight hour. Combining these gives a total for flight management failures for GA of 3.8×10^{-6} per flight hour.

From the UAS accident analysis in Para., the Predator type had one accident when the aircraft ran out of fuel following a loss of communications and two accidents when it was inadvertently flown into dangerous weather conditions. Up to and including 2009, the Predator fleet had accumulated 655,463 flight

hours. The above therefore represent a combined mission mismanagement loss rate of 4.6×10^{-6}.

First to consider is whether the figures for UAS and GA are comparable (other than that the figures are almost identical)? Predator flies long endurance missions, upwards of 20 hours on average, sometimes many miles from a recovery airfield. Conversely, GA flights are of the order of 1-2 hours, often within 10-20 miles of an airfield and with good visual (VFR operation) lookout for poor weather. It could then be argued that these three factors (flight time Vs. distance from base Vs. visual lookout) balance themselves out.

If we accept that the loss rate is of the order of 4×10^{-6} and that loss of the aircraft is classified as "Hazardous" (which requires a probability of $<1 \times 10^{-7}$), a mitigation strategy of weather avoidance routing and on-board flight management would only have to have a performance better than 25:1 to achieve an acceptable loss rate. If it achieved 250:1, it would be acceptable at the "Catastrophic" level. To investigate system performance, we have to also understand the failure rate of such a system. This situation can be summarised by the following equation:

$$\text{Prob}_{\text{Flight Management Accident}} = \text{Prob}_{\text{Flight Management Error}} * \text{Prob}_{\text{Flight Management Error Mitigation Failure}}$$

where we surmise that:

$\text{Prob}_{\text{Flight Management Error}} \underline{\sim} 4 \times 10^{-6}$ and $\text{Prob}_{\text{Flight Management Accident}}$ is required to be $<1 \times 10^{-7}$

6.2.2 FLIGHT MANAGEMENT FAILURE ANALYSIS

In the assessment above, failures due to poor flight management were subdivided into fuel and weather related. Only the fuel aspects are now assessed in further detail:

Fuel Management Failure Analysis

All powered aircraft including UASs have a self-contained fuel system. This system can be quite complicated in large aircraft but even in light aircraft the principles and architecture have common components, typically, tanks, lines, pumps, valves and gauges. A diagram of a typical fuel system on a light aircraft is shown below:

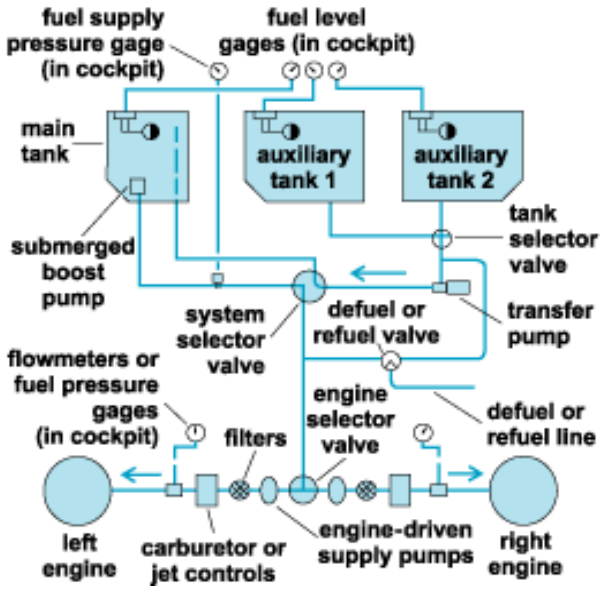


Figure 51: Simple Fuel System[104]

Failures in any of these components can lead to fuel system failures. In assessing fuel system failures it is important to recognise the difference between *fuel starvation* and *fuel exhaustion*. The former describes fuel being on board an aircraft but is unable to be burnt by the engine(s) and the latter describes the situation when there is no more fuel on board. A simple fault tree[105] below describes this situation and assigns possible (but not exhaustive) causes:

---

104 *"Aircraft Fuel Management"*: Experimental Aircraft Info website at http://www.experimentalaircraft.info/flight-planning/fuel-management.php

105 A fault tree is a tool that allows the analysis of an undesired state of a system. It traces the top level state, via intermediary causal states to bottom level causal events. These states and events are graphically linked by gates which define their logical relationships, (e.g. AND or OR gates). By specifying the these logical relationships, together will knowledge of the probability of the bottom level event occurrence, it is possible to derive the probability of the top level

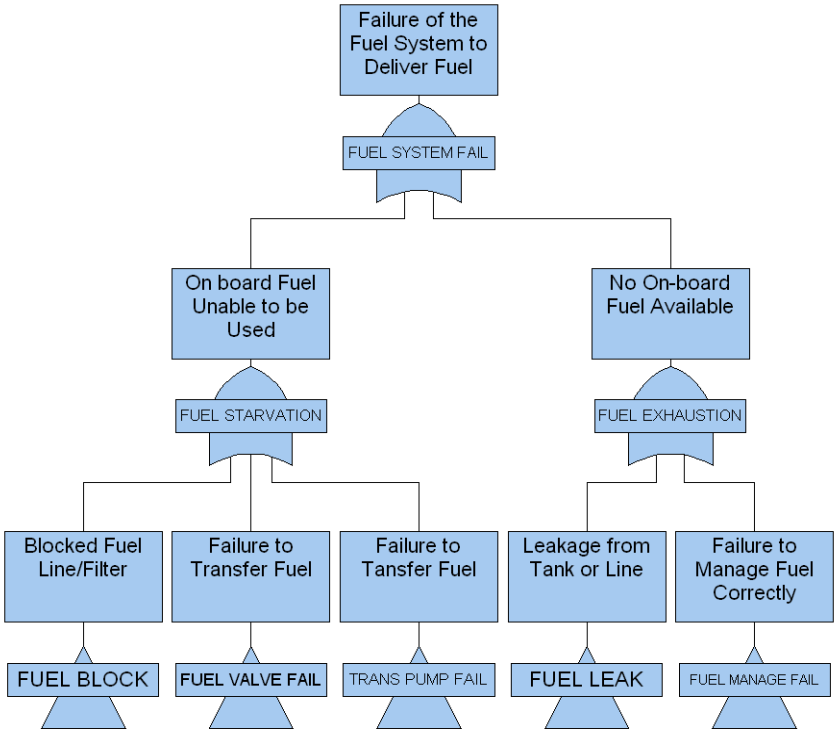


Figure 52: Fuel System Fault Tree (Top Level)

The failure on the above fault tree that is under consideration here is the Fuel Management Failure whereby flight termination is caused by running out of fuel before the aircraft can land safely. The other causes or failures come under the control of the Health Management System and are out of the scope of this thesis and not considered further.

Given the above, it should now be possible to identify what would cause a Fuel Management Failure and attempt to ascribe a probability to it. In order to do this, it is necessary to examine the architecture to determine the components which contribute to this process.

The module responsible for managing fuel during flight is the Mission Manager. This takes the pre-flight Flight (and Fuel) Plan and updates it during flight according to actual, as opposed to planned, routing and fuel consumption. As such it maintains an up-to-date estimate of fuel remaining at the planned

---

state occurring.

destination. However, a variety of changes to the flight plan may occur. These are:

- Change to the planned destination due to operational, weather or other constraints. In such a case, changes to the planned alternates[106] are also quite likely.
- Change of routing imposed by the operating authority, ATC or otherwise, such as routing around poor weather etc. This can include changes of Flight Level (altitude) and/or speed which will alter the fuel consumption rate.
- Changes in the environmental conditions encountered as opposed to that predicted, such as temperatures or winds. Again these will change the fuel consumption rate and/or extend/reduce the flight times.
- Changes to required time on task if that is the mission type (such as a Search and Rescue mission).
- Emergencies which require the UAS to land at the Nearest Suitable, or if urgent, the Nearest Available airfield.

All of the above will necessitate updates to the flight, and therefore fuel, plans. If the update indicates that the fuel remaining at the destination is outside of stipulated constraints[107], then a major revision to the flight plan is required. As in common with the rest of the architecture, an updated plan will be proposed to the ground based pilot (if available). In general, and in order to make the reaction to the critical belief immediate, the Mission Manager constantly calculates the following:

- Updates to the Flight and Fuel Plans according to actual conditions

[106] In general, alternates are considered for weather or unavailability at the planned destination whilst the flight is in progress. These are sometimes referred to as the Weather Alternate or Alt 1 (which is usually a fair distance from the original destination, for obvious reasons) and the Crash Alternate or Alt 2(should a crash at the planned destination close the airfield). The latter, of course, can be very close to the original destination. Therefore the fuel states to reach these alternatives are usually quite different. Which one is chosen depends on many things but primarily the likely weather at the planned or alternate airfields.

[107] The “Fuel at the Destination” constraint is a complicated sum composed of many parts, such as weather, type of approach, unusable fuel, safety margin etc. In the UK, it is mandatory for civil traffic to have at least 30minutes of fuel remaining at the destination.

encountered. These will provide estimates to the fuel remaining after landing

- Contingency routes to:
    - Return to Base
    - Divert to the Planned Alternative
    - Divert to the Nearest Suitable
    - Divert to the Nearest Available
    - Nearest Safe Forced Landing site
    - Nearest Safe Ditching area
- Algorithms to choose the best alternatives to the above sites based on selected criteria
- Updates to the actual weather at the above sites

The required behaviours for Fuel Management are generated by the rule-set below:

No	Rule Antecedent (Critical Belief)	Belief Partition	Rule Consequent (Decision/Action)	Decision Partition	PoF [108]
1	IF Fuel is insufficient for planned destination plus reserves	Mission Manager	THEN Declare Fuel Priority	Mission Manager	All
2	IF Fuel is insufficient for planned destination	Mission Manager	THEN Divert to Planned Alternative	Mission Manager	All
3	IF Fuel is insufficient for planned alternative	Mission Manager	THEN Return to Base	Mission Manager	All
4	IF Fuel is insufficient for Return To Base	Mission Manager	THEN Divert to Nearest Suitable	Mission Manager	All
5	IF Fuel is insufficient for Diversion Nearest Suitable	Mission Manager	THEN Divert to Nearest Available	Mission Manager	All
6	IF Fuel is insufficient for Diversion Nearest Available	Mission Manager	THEN Plan to Force Land	Mission Manager	All

[108] Phase of Flight. i.e. Take-off, Landing, Cruise etc.

Table 22: Rule–set for Fuel Management Behaviours

In Para. 3.1.7.1, the concept of safety critical chains were presented. These are initiated by control functions, normally by the pilot. In the AIMS architecture, these control functions are initiated by the appropriate manager, and in the above case, the Mission Manager. The control signals are routed via the Mission Controller to the Vehicle System for final implementation. As discussed in Para. 3.1.7.1, these control signals are themselves initiated, not now by a Pilot but by a Critical Belief in the Decision Partition. These beliefs, for Fuel Management at least, are the Rule Antecedents in

Table 22 above. Failure to generate these beliefs will therefore result in a potential Fuel Management Failure. These beliefs are themselves generated in the Information (or World Modelling) Layer at the appropriate Belief Partition. Thus we can show the full architecture for the generation and control implementation of the required behaviours for Fuel Management in the Figure below:

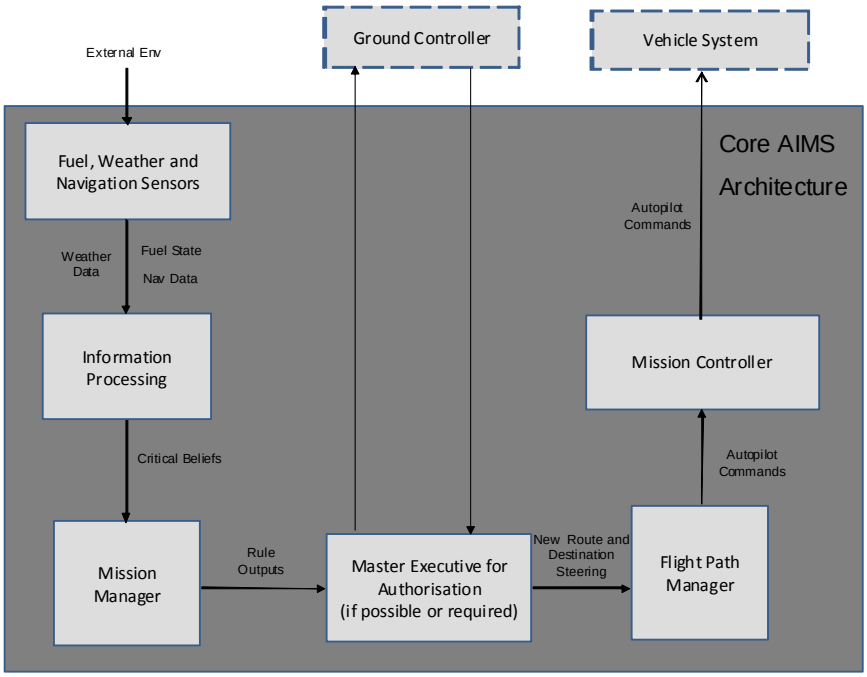


Figure 53: Control Chain for Fuel Management Behaviours

It can be seen that the decision chain in the architecture shown above allows for ground controller/pilot intervention in a fully autonomous manner. That is, the pilot, if available, can be included in the loop for authorisation or, if communications failure prevents this, a timely decision will made autonomously. This point is discussed in more detail further.

From the chain above, a fault tree for the analysis of the Fuel Management Failure state, as identified in the top-level fault tree in Figure 52, can then be developed. This is shown below:

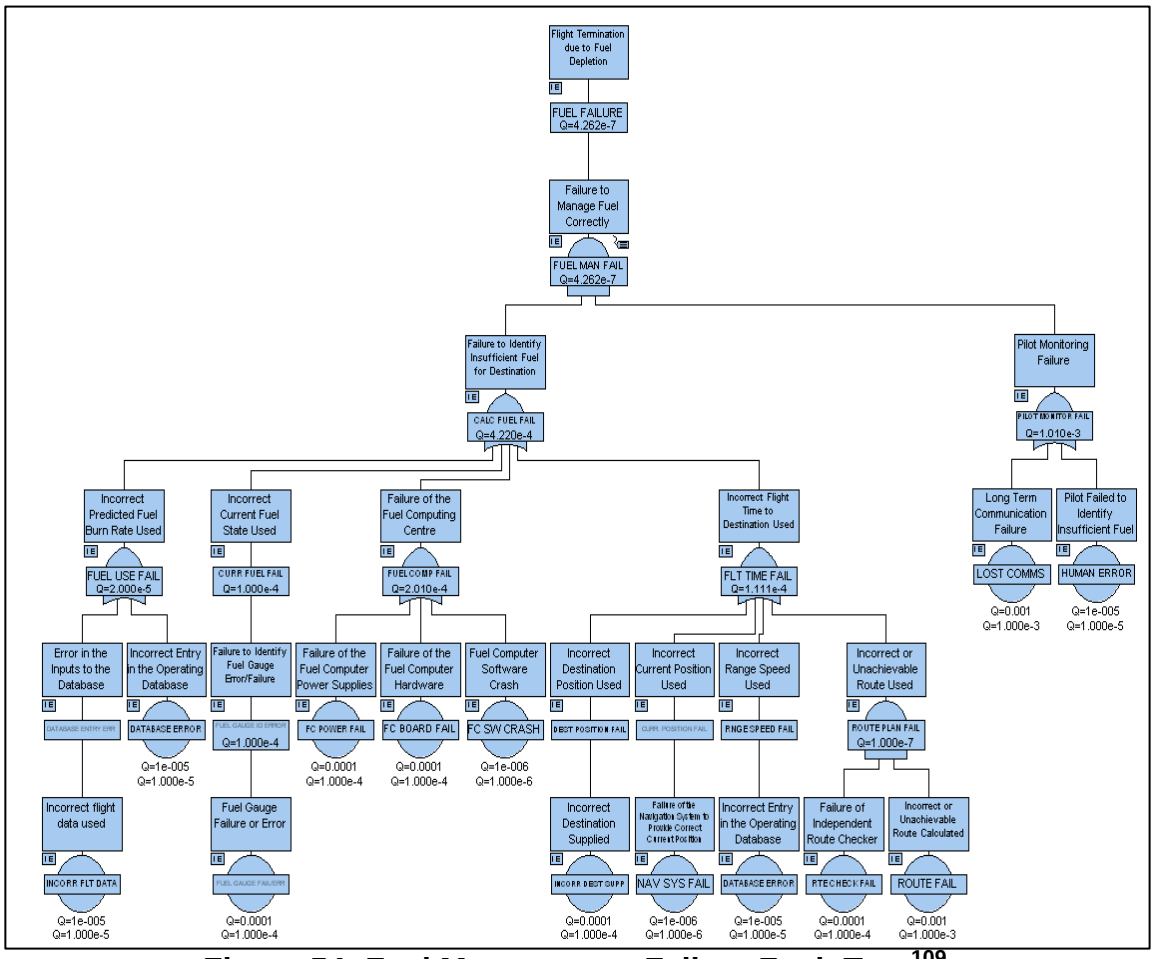


Figure 54: Fuel Management Failure Fault Tree[109]

The above fault tree shows the top level state, “Fuel Failure” which is caused by a failure of the system to identify insufficient fuel for the destination AND the pilot failing similarly. The latter is caused by either human error OR by losing communications. In failing to identify sufficient fuel, the system has either used the wrong data in its calculations, OR the computing centre has failed. The

[109] In this fault tree above, base level events are signified by a circle and bar (“London Underground” symbol), an AND Gate by the symbol AND GATE and an OR gate by the symbol OR GATE. A “Transfer” gate, i.e. one that has only one underlying causal state or event, retains only the bar. The symbol “Q” specifies or derives the probability of the state or event. A justification for these probabilities is given later. Note that an AND gate effectively multiplies the underlying probabilities whilst an OR gate effectively adds them.

events which can cause these effects are shown at the bottom of the tree. It can be seen, by inspection, that the overall top level state probability can be reduced to an acceptable level by introducing extra AND gates in the underlying tree structure or by reducing the probabilities of the major contributing base level events.

Summary

To summarise, a link has been developed which traces the hazard of a flight management failure leading to catastrophic loss of the aircraft which has an acceptable probability of occurrence of $<1 \times 10^{-7}$. One of the contributors to a Flight Management failure is that of a Fuel Management Failure. Using the architecture of the decision making system, a failure chain was developed from which a fault tree analysis was conducted. This fault tree specifies the link between the top level failure state and the underlying causal events and indicates its probability of occurrence.

Discussion

Now that a direct link has been made between the hazard and its underlying causal chain, a comparison and assessment can be made between different decision architectures. In addition, alterations and improvements can be suggested to the original decision architecture in order to reduce the probability of occurrence of the hazard to an acceptable level.

Decision Architecture Comparisons

The alternative architectures to be compared are:

- A manned aircraft decision architecture
- The decision architecture of the “Predator” UAS
- The proposed architecture as previously defined and illustrated

Each of the above have apparently only slight or subtle differences but in fact these differences can have a major impact which will be clearly illustrated using the above analysis tools.

Manned aircraft architecture

The crucial difference between the UAS architecture and manned aircraft is that there is no increase in probability of failure due to communication loss – the pilot(s) is always in communication by his very presence. In addition, the pilot will monitor the fuel gauges and should notice any anomalous readings. Ideally, fuel management will be conducted using a manual or automated fuel log[110]. In addition, pilots will use simple, mental rules of thumb for the fuel calculations which constitute a form of plausibility checking. Referring to the above fault tree, the probability of Pilot Monitoring Failure (PMF) is dominated by the Communications failure event. Eliminating this for manned aircraft suggests that the PMF probability is reduced to around 1×10^{-5}. For large commercial aircraft, with sophisticated Flight Management Systems, using this figure for PMF reduces the probability of overall Failure to Manage Fuel Correctly to something of the order of 1.2×10^{-10}. This figure may only be a coarse estimate but according to Airsafe[111], there have been only six such accidents in the last 45 years, and at least three of these were compounded by poor weather.

Predator UAS architecture

On Predator, there is no on board fuel management system and the protection for Fuel Management Failure resides solely with the flight crew. As such, the fault tree for this architecture is the RHS of the above fault tree. As discussed earlier, this is dominated by the probability of long term communication loss which interestingly was the circumstances of the Predator loss referred to earlier. A clue as to why there are not more Predator accidents due to Fuel Management Failure, since communication loss is quite frequent, is possibly

[110] A Fuel Log is a graph of expected fuel remaining (Y axis) versus flight time (X axis). During the flight, actual fuel remaining is plotted and compared with that originally expected at that time. Deviations can be attributed to several variables or causes such as extended/reduced estimated time of arrival, higher/lower fuel burn rates or unanticipated fuel loss. By extrapolating the actual fuel plots over time, estimates can be made of the fuel remaining after landing.

[111] Airsafe: A website for communicating safety matters for air passengers at http://www.airsafe.com/events/noengine.htm

that Predator has a lost link procedure which calls for it to climb to a safe altitude and conduct an alternative mission (which is usually to Return to Base (RTB)). Since Predator has a long endurance, and generally plans for a Return to Base anyway, the probability of running out of fuel after curtailing its mission prematurely is obviously greatly reduced. However, that operational doctrine (of planning to RTB) constrains the employability of Predator to those sort of missions and it would be expected that the loss rate would be much higher if long range transit missions were routinely conducted.

Conclusion

By including Fuel Management functionality into the architecture and having a decision making system that can act according to its predictions of estimated fuel at the planned destination, the following advantages are envisaged:

- The UAS loss rate due to fuel mismanagement will be reduced, perhaps by an order of magnitude
- Mission plans and operational doctrine can be more versatile and flexible, with more alternatives being available to ensure safe completion of the air mission.
- This latter advantage may help to convince regulatory and operating authorities to allow the UAS into non segregated and routine classes of airspace.

6.3 Take Off and Landing

It is bad to be low and it is bad to be slow. So never, ever, get low and slow[112] - Anon.

It is in the Landing phase of flight that aircraft have the most accidents and UASs are no different. In fact, they are more likely to be worse for reasons that will become apparent. There appear to be three main contributors as to why accidents are more prevalent in these phases. Firstly, there is nearly always a lack of time in which to initiate corrective action because of the proximity of the

[112] Ancient pilot homily (if pilots can be considered ancient)

ground. Humans are reasonably fast at the skill level but, as has been noted, can become ponderous or even frozen at the knowledge or planning levels. Machines, as has already been demonstrated, can be nearly instantaneous at the reactive level. In addition, humans can suffer from attention fixation where concentration on a particular aspect leads to loss of situational awareness of other aspects. This has been cited as the main reason for many mid-air collisions of GA aircraft which frequently occur in broad daylight and on the approach to landing[113] [38]. Secondly, the number of alternatives available to a decision maker during these phases is much reduced and it is frequently true that, of these alternatives, there may not be a fully satisfactory solution. Finally, aircraft in these phases are at the end of one extremity of their operating environment and invariably have low potential and low kinetic energy (the low and slow) with which to extract themselves from difficulties. Given the remote operation of UASs and the inherent lack of intimate control, it should be of no great surprise to note that UASs have an even poorer record of accidents occurring in these phases.

In order to overcome some of these difficulties, many modern UAS[114] now have automated take-off and landing modes thus relieving pilots of the skill levels required to remotely achieve these operations. However, it has already been noted that automation brings its own difficulties such as removing the pilot from the loop and being either inherently inflexible or complex in operation. So how could an autonomous UAS with the proposed decision architecture improve things? Several areas are identified below.

In reviewing the performance of an autonomous architecture in a Sense and Avoid scenario, two aspects were noted. The inherent ability to include the pilot in the decision loop for a long as possible to rectify an escalating problem and the ability to make reactive decision/control sequences. In addition to these aspects, the proposed autonomous decision architecture also has the ability to

[113] The reference cites concentration on the touchdown point and a high level of arousal as to the main reason why a good lookout during the approach, essential in all aspects of flying, particularly in the proximity of other aircraft, is often overlooked.

[114] Global Hawk, BAE Systems Herti and Mantis UAS and Reaper are good examples.

rapidly assess a series of alternative Courses of Action (CoA), selecting and, if authorised, implementing the most cost beneficial (literally).

To illustrate these aspects, an analysis of the an accident [54] occurring to an Airbus A320-214 when it ditched in the Hudson River has been made. The accident details and the timeline, taken directly from the official accident report, are given in the table below:

NTSB Accident Report — US Airways Flight 1549
Airbus A320-214, New Jersey January 15, 2009
Loss of Thrust in Both Engines After Encountering a Flock of Birds and Subsequent Ditching on the Hudson River

Timeline of Events

Ref No	Local Time	Time Diff	Elapsed time from start (secs)	
1	15:27:11	0	0	Multiple bird-strikes occur at 3,000ft in the climb out
2	15:27:13	2	2	Engines start to fail
3	15:27:18	5	7	Crew realise engine failure
4	15:27:21	3	10	Captain makes decision to start the APU
5	15:27:23	2	12	Captain makes decision to take manual control of the aircraft
6	15:27:26	3	15	Captain has manual control of the aircraft
7	15:27:28	2	17	Captain calls for checklist
8	15:27:33	5	22	Captain calls Mayday and starts to turn back to La Guardia airport
9	15:27:50	17	39	Ist Officer starts checklist
10	15:28:05	15	54	ATC comms Interrupts checklist
11	15:28:07	2	56	Crew start to realise that landing at an airport may not be an option
12	15:28:11	6	62	Captain replies to ATC
13	15:28:14	3	65	Ist Officer resumes checklist
14	15:28:31	17	82	ATC comms Interrupts checklist
15	15:28:35	4	86	Captain replies to ATC
16	15:28:36	1	87	TCAS Traffic advisory message
17	15:28:37	1	88	Crew discuss ATC message
18	15:28:45	8	96	PWS Wind shear alert
19	15:28:45	0	96	Ist Officer resumes checklist
20	15:28:46	1	97	ATC comms Interrupts checklist
21	15:28:55	9	106	Captain replies to ATC
22	15:28:59	4	110	Ist Officer resumes checklist
23	15:29:02	3	113	ATC comms Interrupts checklist
24	15:29:03	1	114	Captain replies to ATC
25	15:29:05	4	118	TCAS Clear of Conflict message
26	15:29:07	2	120	Ist Officer resumes checklist
27	15:29:11	4	124	Captain warns passengers to brace for impact

28	15:29:15	4	128	GPWS warning at 1000ft altitude
29	15:29:16	1	129	Ist Officer resumes checklist
30	15:29:21	5	134	ATC comms Interrupts checklist
31	15:29:33	12	146	Captain replies to ATC warning that the aircraft would be going to ditch
32	15:29:37	4	150	Ist Officer resumes checklist
33	15:29:45	8	158	Captain orders flaps; crew give up checklist and start to prepare for the ditch
34	15:29:53	8	166	ATC loses radar contact with aircraft but advises relative position of Newark airfield
35	15:30:01	8	174	Flaps at position 2, aircraft at 250 feet, 170 knots
36	15:30:17	16	190	Crew make decisions on further flap settings - aircraft at 150 knots
37	15:30:24	7	197	Three GPWS "Terrain, Terrain" warnings
38	15:30:41	17	214	Aircraft at 50 feet
39	15:30:44	3	217	End of transcript.

Table 23: Event Timeline of US Airways Flight 1549 Accident

There are several points worth noting from the above but not all are in scope for this thesis. What is worth mentioning is the exemplary crew co-operation and communication displayed. There can be little doubt that this crew demonstrated human performance, in all aspects, at its very best.

Diagnosis of Engine Failure The crew noticed the birds just before impact, felt the effect of the impact, heard both engines spooling down and correctly diagnosed a double engine failure within 7 seconds. There must be considerable doubt that a remote pilot would even be aware of the bird-strike, would certainly not have felt the impact and may have even taken considerable time to notice the engine winding down in the absence of any audio clues. In automatic flight, the first he would probably be aware of is the aircraft losing height for what would appear to be unknown reasons. There have been several instances of exactly this general sequence of events in the Predator accidents reports referenced at 0.

This highlights a key aspect in the strategy for improving the overall level of safety is to incorporate system health management and subsequent emergency handling in the decision architecture, since correct handling of emergencies has a direct influence on the accident statistics (as does incorrect handling!). If the diagnosis of a system failure is clear cut and there remains plenty of time for the human controller to put in place emergency actions following the failure, there

seems little that an on-board decision making system can do to improve performance. Unfortunately, and for a variety of reasons, such benign conditions are relatively uncommon.

Let us consider the process chain from the detection of an anomaly through to its safe mitigation. An example of such a process is well documented as Open System Architecture - Condition Based Maintenance (OSA-CBM) and, although developed primarily for aircraft maintainability, it remains relevant to a real time air incident. This process follows the steps outlined below:

Sensor Readings ⟶ State Detection ⟶ State Diagnosis ⟶ State Prognosis ⟶ Impact Assessment ⟶ Capability Assessment ⟶ Failure Mitigation ⟶ Failure Handling.

It can be seen from this that this process highlights that a simple single failure (such as a fuel leak or engine failure) can progress through to have a direct impact on operational matters such as diverting to an alternative landing site (handled in the proposed architecture by the Mission Manager. Even though the initial failure may be contained within a small sub-system component. The impact of that failure may require a co-ordinated response, such as airfield selection, fuel and route re- planning (and validation) and pilot authorisation, at the highest level.

In several demonstrations of the AIMS decision architecture operating in routine flight with an on-board health management system[115], the engine failure was diagnosed within 5 seconds of the triggering event and corrective action initiated.

The conclusion to be reached here is that there is a definite requirement for an effective Health Management System, capable of sophisticated reasoning to diagnose failures, predict the impact of that failure and propose mitigating plans

[115] This failure condition was demonstrated several times to VIP guests. All guests who were current or ex aircrew were astounded by the speed of response of the system. Having personal experience of such a situation, I can assure the reader that it can take several seconds for a crew to respond appropriately.

to the higher levels of authority. Such a system must of course be integrated within the decision architecture with the Health Management System informing the Emergency Manager with appropriate beliefs and plans. The Emergency Manager would then select and sequence the appropriate plan to the Mission Controller and task the Mission Manager (in this exemplar) to handle the operational consequences. The Mission Manager would then inform the Master Executive which would in turn seek the appropriate authority if required. Such a sophisticated and complex architecture would be very difficult to design as a stand alone automatic system. It is, however, a relatively simple addition to the proposed decision architecture since the required infrastructure and message handling is generic. This factor illustrates that the scalability of the proposed architecture.

Check List Procedures In a multi-crew aircraft, it is a standard response to any emergency to take control of the aircraft, carry out immediate emergency actions and then institute a formal "check and respond" of each subsequent action in the appropriate checklist[116]. In the Airbus 320, for failures that can be identified by the aircraft sensors, the appropriate crew actions are displayed on the Electronic Centralized Aircraft Monitor (ECAM). For others, the crew can refer to the Quick Reference Handbook (QRH), which contains abnormal and emergency procedures for such events The fact that the QRH checklist in this case was 3½ pages long and, as it turned out, inappropriately designed, is irrelevant to this analysis. What is of note is that the 1st Officer has to retrieve the (bulky) QRH, find the appropriate checks and be prepared to read them out

[116] The process followed in this case[54] was detailed in the US Airways A319/320/321 Pilot Handbook (PH), Chapter 9, "Non-Normal Operations". This states, in part, that, when a non-normal situation is evident, the pilots should methodically accomplish the following steps:

1. PF (Pilot Flying) - maintain aircraft control;
2. Identify the non-normal situation, PM (Pilot Monitoring) - cancel the warning or caution, if applicable;
3. PM - determine if situation requires an Immediate Action or if it is an ECAM Exception;
4. PM - accomplish Immediate Action Items, if applicable;
5. Captain - assigns PF;
6. PM - accomplish non-normal procedure; and
7. PM - accomplish ECAM follow-up procedures, if applicable.

aloud when called for. All of this takes valuable time. In this particular case, he actually never got past the first three checks in the list and was repeatedly interrupted by Air Traffic Control (ironically, trying to help). As the official accident report noted: "*Further, the flight crew never reached the ditching portion of the checklist, which most directly applied to the accident situation*". There are several instances of accidents where the crew become interrupted in the checks and either re-start at the wrong point or fail to complete them through lack of time. In the accident analysis, 7 Predator accidents were directly caused by the operator failing to carry out the correct procedure.

In the AIMS decision architecture, the Master Executive Emergency Manager is responsible for the checks[117] being carried out. These are initiated in response to a complex chain of events discussed earlier. To carry out these checks, the Emergency Manager sequences the checks to the Mission Controller in turn until all actions are complete. In the engine failure demonstration, the complete checklist was completed[118] in less than 2 seconds. In the Airbus accident, as an aside, the Captain remembered that one check was to start the Auxiliary Power Unit (APU) and this vital[119] check was completed within 10 seconds.

To conclude, autonomous sequencing of the appropriate plan (checklist) is easily achieved by the AIMS architecture and at high speed[120]. Additionally, once called for (or, in our ontology, proposed to and authorised by the pilot at a PACT level of 5A) it is uninterruptible by ATC and does not lose its correct sequence. In comparison to automated UASs, would there be much of a

[117] The Emergency Planner is the holder of the Emergency Plans (i.e. checklists) and provides them to the Emergency Manager on receipt of the appropriate request as detailed in Figure 13: Generic 4-Layer . This process mirrors human actions but of course can be done far faster and more accurately.

[118] The checks completed were: 1 - Correctly identify failed engine, 2 - Shut it down, 3 - Set transponder to the emergency code, 4 - Determine nearest suitable landing airfield, 5 – Plot new route to the airfield, 6 – Self-authorise new route, 7 - Turn onto new route. These demonstrations did not include the pilot in the loop as no Ground Control Station was then available and therefore it represents operation at the PACT 5A level

[119] As it turns out, had this not been done before ditching, a mode of the autopilot in this aircraft would have prevented a successful outcome by invoking manual flight control laws when it sensed they were below 100 feet without the undercarriage down and without power. It is believed that this autopilot mode was unknown to the aircrew.

[120] The decision architecture normally runs at 10Hz. It could easily be made to run at 50Hz.

difference? Well, once invoked almost certainly not – an autonomous architecture operating at PACT 5A is no faster or better than an automatic system switched on. However, the AIMS architecture selects the appropriate plan and proposes it to the pilot. If the pilot thinks the architecture has got it wrong, he can refuse to authorise it whereas an automatic system would just carry on. Again, the human remains in the loop at the appropriate level.

Alternative Options After 22 seconds, the Captain started to turn back to the originating airfield of La Guardia (LGA) but after 56 seconds realised they may not make it successfully. To quote from the report:

At 1528:46 (97 seconds in), the controller stated that runway 4 at LGA was available, and the captain responded, “I’m not sure we can make any runway. Uh what’s over to our right anything in New Jersey maybe Teterboro?” The controller replied, “ok yeah, off your right side is Teterboro Airport [TEB].” Subsequently, the departure controller asked the captain if he wanted to try going to TEB, and the captain replied, “yes.” At 1529:27 (140 seconds in), the departure controller then asked the captain which runway at TEB he would like, and the captain responded, “we’re gonna be in the Hudson.” It seems reasonable to assume that the option of ditching was selected at that point.

The AIMS architecture is specifically designed to create alternative plans and to propose the best one for selection. Not only that, it constantly updates its plan in the light of current beliefs (and these may rapidly change, as in this accident) and monitors progress of a chosen plan. If a better one is subsequently identified, then that plan is proposed instead. Additionally, it generates plan metrics thus highlighting how much better (or worse) the current plan is in relation to the identified alternatives. This is in stark contrast to an automated UAS or even one that is manually controlled.

An analysis of the decision options available to the Captain was made after the accident and concluded, after several simulations, that it was possible to return to La Guardia but not necessarily with a guarantee of safety. However, most experts concur that he did the “right thing” in terms of safety of his passengers.

Algorithms generating and analysing options for exactly this type of event have been developed[121] and could possibly be beneficial in presenting decision options to the crew.

6.4 Lost Link Procedures and Communications Handling

It has already been remarked upon that the communication link with the UAS is essential for effective human management (rather than control or supervision) of the on-board autonomous system. It has already been noted that this link is neither instantaneous (as in manned aircraft) nor guaranteed to be available (a failure rate of 1 X 10^{-3} per flight hour has been quoted). Most current medium sized UASs have, what is termed, a lost link procedure which is an automatic mode instigated when the link has been determined to have been lost. A description of the Predator lost link procedure, as described in the NTSB Accident Report of the crash at Nogales, Arizona on 25 April 2006 [55], is given below[122]. Global Hawk has a similar procedure but is also capable of flying to a pre-planned destination and carrying an emergency landing automatically even without ground support. However, the system output when communications are lost needs to be somewhat more sophisticated than merely safe navigation and landing if full airspace integration is to be achieved. The main drivers for this are equivalence (the need to conform to regulations) and safety. Without human management of the mission, the aircraft needs to be able to generate safe routes. These routes must be validated against flying into the ground at the chosen altitude, conforming to any airspace restrictions and avoiding dangerous weather. The UAS must also able to handle emergencies that may arise and be capable of avoiding other air users in emergency collision scenarios. Ideally, the

[121] These require a reference to be found

[122] To quote directly from the report, "..when the UA goes into a lost-link profile, it will initially turn to a pre-set lost-link heading, go to full power, and climb for 51 seconds at a commanded airspeed of 105 knots. If the UA is within 200 feet of the lost-link altitude (or higher), this first step is skipped. In the next step, the system generates a waypoint at the pre-set lost-link altitude, 2.5 nautical miles from the location where the UA started the lost-link profile, in the direction of the lost-link heading, and the UA proceeds to that waypoint. Once the UA reaches that waypoint and the lost-link altitude (or 30 minutes later, whichever comes first), it will proceed to fly the remainder of the lost-link profile. This portion of the lost-link profile consists of a predetermined series of altitudes and locations, which form a path that the UA will autonomously fly. If a data link cannot be re-established, the UA cannot land, and it will eventually run out of fuel and crash at some location along the lost-link profile route.

UAS should also be capable of interacting with the Air Traffic Control agencies and other air users in a transparent way. In short, and returning to familiar ground, the UAS must be self-aware, and in a fairly sophisticated way in order to be effective.

As mentioned, Global Hawk is the most sophisticated UAS currently in existence. In lost link conditions, it is capable of flying pre-planned routes to alternative landing airfields and carrying out a safe landing. However, the caveat is “pre-planned”. The routes are validated before take-off and are stored as contingency plans for several points (every 5 minutes has been suggested) along its mission route. This is why it takes some 24 hours to fully plan a mission (though there are believed to be unconfirmed plans that some form of on-board route planning will be introduced).

The AIMS architecture is capable of operating at whatever level of authority that has been invested on it. The level of sophistication of behaviour is high but this is only due to the number of autonomous functions it has been designed to have. When operating in a lost link situation, it will revert all authority levels to PACT 5B (full authority to act appropriately, but no feedback to the pilot because he is not available). If this is compared to an automatic system, then, provided that the functionality is the same, and the triggering mechanisms (beliefs for AIMS, variable states for automation), there can be nothing to choose between them in terms of improved behaviours. The AIMS architecture would be easier to program though, since it is predicated on selection of the best alternative plans rather than hard coded conventional methods and has been noted is fundamentally scalar.

In conclusion, AIMS offers only a slight, and arguable, advantage over an automated system for the lost link situation. However, in comparison to current UAS, even Global Hawk, it should prove to be highly responsive to handling a lost link situation.

7 Summary of Results

7.1 The Thesis

In Chapter One, the general thesis was proposed. This stated that the role, responsibilities and environment of an autonomous UAS, are sufficiently different from those of other conventionally controlled manned and unmanned systems to require a different architectural approach. Such an architecture must also demonstrate acceptable safety, performance and robustness to unforeseen events. This led to the following questions being posed:

1. What do the issues of role, responsibility and environment force the architecture to achieve that is not found in other architectures?
2. Can existing autonomous system architectures address these issues? If so, how, to what degree and why? If not, why not?
3. What effect does the fact that the operator is remote have on the underlying accident rate compared to manned aircraft? Are the accident mechanisms the same or different?
4. How does the need for operator interaction affect the design of the architecture.

A review of the subsequent work will now be made to assess whether these questions have been satisfactorily answered.

7.2 Problem Analysis

In evaluating the problem at hand, it was argued that the trend to reduce costs by introducing increasing levels of automation had upper limits when also attempting to increase UASs versatility, safety and performance. The argument was proposed that a move to autonomous operation would potentially remove this limit. However, it was noted that such an autonomous vehicle would have to be capable of operating in conjunction with a human controller at whatever level of authority was invested by him **or,** if he was unavailable, at whatever level

was appropriate to the situation. This latter requirement requires the architecture to be self-aware and identify, and commit to, appropriate courses of action – features that are inherent to autonomous systems and not necessarily to increased automation.

7.3 On the Decision Architecture

A review of several architectures was undertaken and discussed in terms of their suitability at Appendix C. It was noted that there were several that had favourable attributes but none that would fit all the requirements previously identified. A review of the nature of autonomy was undertaken, together with methods for suitable human control. From this a generic 4-Layer architecture was developed based on a fusion of robotic and avionic practices which could operated with mixed levels of authority by a human controller. This architecture was further developed to produce a hierarchical model with specific functionality to address the issues of versatility, safety and performance. This functionality was carefully partitioned according to the identified decision space and in so doing demonstrated the scalable nature of the architecture. By adding increased functionality where required, a route to increased versatility of operation is demonstrated.

In deriving the requirements from an analysis of the problem, together with the analysis of current approaches and the development of the proposed architecture, it is considered that Questions 1 & 2 above were answered by that study and proposal. Additional material in support of this was provided in summarising the lessons learned from implementing and operating the architecture in its synthetic environment. These lessons provided further evidence that moving to a plan based system of control whereby appropriate plans are generated as consequence of implementing a better HMI based on the setting of objectives. Further, enhancements were suggested by the move to a fuzzy rule based system.

However, to assess Question 3 and the issue of acceptable levels of safety, an accident analysis of manned and unmanned aircraft was undertaken.

7.4 Accident Analysis and Overall Safety Contributions

The accident analysis of manned aircraft, commercial and those in General Aviation, showed common trends. Namely that accidents frequently occur in the Landing Phase of Flight and that the primary cause of 70-80% of accidents was human error. Additionally, foremost amongst the causes were Loss of Control (though less so for commercial aircraft with autopilots). However, weather[123] detection and avoidance and fuel management were also important factors for autonomous control system to consider. Mechanical reliability, maintenance and design were important factors but outside the scope of a such a system's control and therefore this thesis. These factors were also present, and to a similar degree in relation to the overall accident rate, when considering unmanned systems. However, additional factors, attributable (in part) to operator remoteness, appeared to be present in 7 out of the 19 accidents analysed. In addition, loss of control was again a prevalent factor for the *Predator/Reaper* family of UAVs but of course entirely absent for *Global Hawk* (no manual control available and none lost so far in automatic flight control).

This analysis indicated several avenues for improving UAS safety and strategies for incorporating these into the decision architecture were proposed. These were:

- All flight control functions should be automated with no manual control.
- Provide weather detection and avoidance mechanisms fully integrated within the decision architecture.
- Provide better fuel management procedures integrated with flight management planning..
- Provide a more efficient means of handling emergencies, particularly checklists and formal procedures.

[123] The term weather is used here in its formal meteorological sense. e.g. icing, rain, thunderstorm, turbulence etc.

- Introduce a range of internally generated cross monitoring, plausibility checking and abduction feedback mechanisms

- Provide improved HMI to show/provide :

    - autonomous system intent

    - increased feedback of aircraft state, particularly of those automated functions and those deemed critical, either to the on-board autonomous system and/or the GCS.

    - status of invested authority (PACT Level) for the key control mechanisms

It was considered that incorporating these mechanisms into the decision architecture would substantially improve the accident rate and although no target figure could be formally calculated, a improvement of between 50% and 75% was postulated.

In developing and presenting these proposals, it is considered that satisfactory answers to Questions 3 & 4 above are completed.

7.5 Analysis of Exemplar Scenarios

In order to validate the above proposals and to further examine the architecture's inherent features, a detailed analysis of the architecture's response in exemplar scenarios was undertaken.

7.5.1 Sense and Avoid

A detailed performance analysis of the response of the architecture in a Sense and Avoid scenario was undertaken. This showed two main results. Firstly, that if detection ranges were less than 8km, impact would occur before an operator's response had been received in 50% of occasions. Unlike an automatic system, the architecture is capable of keeping the operator in the loop for the maximum time possible. This appears to be generally true of the autonomous architecture

and the analysis supports Norman's criticism's of automated systems in Para. 2.2.5.2. and at the same time provides a more robust alternative approach. In addition, the analysis highlighted than under certain circumstances when a nearly instantaneous response was required to avoid a collision, the architecture was capable of reactive behaviour. Under more leisurely conditions, the architecture appears to an outside observer as undertaking deliberative behaviours. This is a inherent feature of the architecture and requires no specific switches or modes to swap between the two.

The analysis concluded that having a Sense and Avoid System on board the collision risk would reduce by a factor of 50% or more.

7.5.2 Flight Management

Two aspects of flight management were considered. Adverse weather detection/avoidance and routing/fuel management. The preliminary analysis concluded that having such functionality on-board would reduce the probability of an accident due to those causes to an acceptable level.

The fuel management scenario was then analysed in detail This concluded by noting that including Fuel Management functionality into the architecture and having a decision making system that can act according to its predictions of estimated fuel at the planned destination, the following advantages could be realised:

- The UAS loss rate due to fuel mismanagement will be reduced, perhaps by an order of magnitude.
- Mission plans and operational doctrine can be more versatile and flexible, with more alternatives being available to ensure safe completion of the air mission. This advantage may help to convince regulatory and operating authorities to allow the UAS into non segregated and routine classes of airspace.

7.5.3 Take Off and Landing

In the Take-Off and Landing scenarios, it was noted that that accidents in these phases generally occur because of the limited time to identify and mitigate the offending cause. The event timeline of an exemplar real world scenario was analysed in detail. This was then compared with an analysis of what an automated and an autonomous system architecture, both with remote operators, could accomplish.

This analysis concluded that a remote operator would have considerable difficulty is diagnosing the scenario presented. This highlighted a clear requirement for a capable Health Management System, fully integrated into the decision architecture. The analysis suggested that an automatic system with the required functionality and human interaction would very difficult to design but for the proposed system, was merely a relatively simple architectural addition. This demonstrates the scalability of the proposed architecture.

The exemplar analysis also showed that autonomous sequencing of the appropriate plan (checklist) could be easily achieved by the AIMS architecture and at high speed. Compared to an automated system, it would generally keep the operator in the loop (potentially in a variety of ways) and help by proposing alternative courses of action (plans). Finally, it was noted that the architecture is designed specifically around the generation and handling of plans as the primary means of communication (unlike an automatic system) and as such is inherently capable of presenting alternative courses of action to the human operator.

7.5.4 Lost Link Procedures and Communication Handling

Many UAS accidents have been caused by lost communications and its consequences. The analysis considered mitigating responses to losing contact with the human operator and concluded that fairly sophisticated functionality needed to be incorporated into the design to accommodate this. In comparison with an automated system, no extra advantage could be found in terms of

performance in this situation, assuming that the automatic system had the same functionality. However, it was noted that, due to the scalability of the architecture, such functionality could be fairly easily incorporated into the design whereas for an automatic system, a complex design would be required.

8 Conclusions and Future Work

In Chapter 2, the notion was proposed that having an autonomous system controlling an unmanned air vehicle in conjunction with a human operator would improve the versatility of employment, safety and performance of that vehicle, particularly in comparison with an automatic system. The safety aspect has been sufficiently covered in the accident analysis and exemplar scenarios as summarised above. In addition, the performance and versatility that the autonomous system of the vehicle has been shown to be much improved, primarily by the integration and selective partitioning of increased functionality within a common architectural framework.

In developing the decision architecture and undertaking an analysis of its features, some common themes emerge and these are now brought together.

8.1.1 Human Control Integration

From the outset, it was noted that such an autonomous vehicle would have to be capable of operating in conjunction with a human controller and or if he was unavailable, at whatever level was appropriate to the situation. Implementation of the PACT levels, provided a suitable route to achieving this. In addition, improved HMI concepts were proposed to further enhance the operators shared understanding.

In making the comparison with automatic systems, the true nature of how an autonomous system can keep an operator in the loop for as long a possible emerged, particularly in the Sense and Avoid exemplar. This was quite unforeseen, as was the degree to which an automatic system keeps the operator out of the loop (although this had long been suspected by HF researchers).

The use of plans also enables a better HMI by proposing alternative courses of action which the architecture generates.

8.1.2 Scalability, Versatility and Functionality

The functional partition of the architecture and the common means of implementation (planner, manager, controller) provides the required scalability, which consequently confers versatility, if required.

8.1.3 Safety

The study provided strong evidence of the functionality required for improved safety. It also showed how this functionality could be partitioned according to the decision space (flight, emergency, health, HMI protocol, etc.). Because of the scalability of the architecture any new avenues for improved safety could be easily incorporated.

In addition, a method for identifying deficiencies in shared situation awareness was presented in the form of an Abduction Loop. Improved ways of achieving higher integrity in critical beliefs (which generate behaviour) was identified by using plausibility checking.

8.2 Recommendations for Further Research

8.2.1 Trust, Legality and Ethics

When considering possible negative aspects to the implementation of an autonomous control system for a UAS, the subject of investing sufficient trust for human devolvement of authority was raised. The USAF Chief Scientist also notes that "*In the near- to mid-term, developing methods for establishing "certifiable trust in autonomous systems" is the single greatest technical barrier that must be overcome to obtain the capability advantages that are achievable by increasing use of autonomous systems*" [56]. This study made no further effort to research this topic as it was outside the scope of the Thesis. However, it is clear that there is a pressing need for more work on understanding the factors required for acceptance of autonomous systems, particularly airborne ones, into routine life. The study also noted on similar concerns regarding

ethics and legality and whilst some work has been done in these areas, much remains to be done.

8.2.2 Fuzzy Rule Based Approach

It was noted that moving to a fuzzy rule based approach may deliver a better HMI by the use of linguistic variables more familiar to the human operator's situation awareness. It could also prevent rapid swapping of behaviours as beliefs change. A move to implementing such a system is recommended as further research.

8.2.3 Agent Programming

The final implementation of the architecture showed the beneficial impact of implementing the authorisations solely by using a plan based structure. This is in line with agent programming languages. Unfortunately, all of these currently available will be hard to certify for the airborne environment due to the languages used (mainly JAVA) and/or because they have developed as a research tool at a University. A High Integrity language, such as SPARK ADA, would be capable of certification and research is recommended to understand the factors that would enable certification of such a programming language or library toolbox.

Appendix A On the Definition of Autonomy and Automation

A.1 The Definition of Autonomy

The word “Autonomous” is derived from the Greek *autonomos* : *auto*-, auto- + *nomos*, law.

An internet search on the definition of ***Autonomy*** rendered the following:

A person’s ability to make independent choices.
www.alz.org/Resources/Glossary.asp

An autonomous being is one that has the power of self-direction, possessing the ability to act as it decides, independent of the will of others and of other internal or external factors.
www.filosofia.net/materiales/rec/glosaen.htm

Freedom from all external constraints. Independence consisting of self-determination.
www.carm.org/atheism/terms.htm

Independence, self-government
www.imuna.org/c2c/app_a.html

The amount of control an individual has over his or her working life. Autonomy can relate to performance goals (the outputs of a role) and performance methodologies (the way in which goals are achieved). Increased autonomy is normally associated with higher levels of job satisfaction. However, too much autonomy can involve a high level of role ambiguity and role uncertainty which can be potent sources of stress for many individuals.
www.oup.com/uk/booksites/content/0199253978/student/glossary/glossary.htm

The ethical principle that independent actions and choices of an individual should not be constrained by others.
www.setnlegalservices.org/glossary.htm

Government agencies will be more effective when they have higher levels off autonomy in relation to external stakeholders, but not extremely high levels of autonomy ·Autonomy to manage its mission and tasks tends to enhance an agency's performance of the mission and tasks ·Autonomy does not mean leaving out stakeholders
fs.huntingdon.edu/jlewis/Syl/PA/306StillmanStudOuts.htm

An aspect of the responsiveness of health systems whereby one enjoys the freedom to decide for oneself on alternative treatment, testing and care options, including the decision to refuse treatment, if of sound mind.
www.emro.who.int/mei/mep/Healthsystemsglossary.htm

The quality or state of being self-governing; especially the right of self-government.
www.historyteacher.net/EuroProjects/DBQ1998-1999/glossary24-99.htm

A term that refers to the independence of the moral or ethical agent in decision-making.
www.texascollaborative.org/Urban_Module/glossary.htm

Immunity from arbitrary exercise of authority: political independence

Personal independence
wordnet.princeton.edu/perl/webwn

Ability to operate on one’s own. (See AUTONOMY) (MP)
www.biol.tsukuba.ac.jp/~macer/biodict.htm

Separate, independent.
www.anbg.gov.au/glossary/webpubl/lichglos.htm

Functioning independently of other components or systems; self-governing or self-controlling; possessing virtually complete closure in normal operation.
www.islandone.org/MMSG/aasm/AASMGlossary.html

Refers to an economic variable, magnitude, or entity that is caused independently of other variables that it may in turn influence; exogenous.
www-personal.umich.edu/~alandear/glossary/a.html

Describes a self-contained system that carries out programs or performs tasks without outside control by acquiring, processing and acting on environmental information.
www.rcmicroflight.com/library/glossary.asp

Independent; self-contained.
www.mariner.org/chesapeakebay/native/vocab.html

(of political bodies) not controlled by outside forces; "an autonomous judiciary"; "a sovereign state"

existing as an independent entity; "the partitioning of India created two separate and autonomous jute economies"

(of persons) free from external control and constraint in e.g. action and judgment
wordnet.princeton.edu/perl/webwn

From the above certain nouns are repeated several times such as: independent [12], operate/action/decide/control [14], self [10], so we could conclude that ***controlling oneself independently*** is a fairly good starting point for defining

autonomy. If there is such variety in the definition of autonomy, it comes as no surprise that there is even more so when considering the definition of *autonomous agent*. However, it would be reasonable to argue that an autonomous agent is ***an agent that controls itself independently***.

Throughout the vast library of documentation on agency, there are a few recurring themes and it is worth elucidating these as a basic framework or blueprint for what constitutes an agent. This blueprint draws particularly on the work of Nwana [57] and, Graesser and Franklin [58].

The basic concept of an agent, and nearly all, if not all, the definitions of agency require the agent to be **situated**. That is, it must exist in an environment and be *de facto* a part of that environment. Thus we can argue that it controls itself within an environment. Clearly, it must do something in that environment and this requires it to sense and act. It is obviously no use doing just anything, so the actions it performs must be in furtherance of some agenda or objective(s). This agenda could change over time or because of some other influence in the agent or the environment. So a basic list of properties is:

- To exist and be in, and part of, an environment
- To sense objects or attributes of or within that environment
- To independently act in order to influence or change that environment and/or its objects in accordance with an agenda.

So, if these properties are accepted we can now re-define our notion of an autonomous system as:

An autonomous system is one that operates within an environment and is capable of independent decision and action in pursuit of its objectives.

The UK MoD have definitions for autonomy and automation and these are reproduced below:

The MoD's definition, from JDN 3/10 [6], is:

'An autonomous system is capable of understanding higher level intent and direction. From this understanding and its perception of its environment, such a system is able to take appropriate action to bring about a desired state. It is capable of deciding a course of action, from a number of alternatives, without depending on human oversight and control, although these may still be present. Although the overall activity of an autonomous unmanned aircraft will be predictable, individual actions may not be.'

A.2 On the Definition of Automatic

Automation or an **automatic** machine is generally defined slightly differently although some views are uncomfortably close to those for autonomy. Automatic comes from the Greek *automatos* : auto-, auto- + -*matos*, willing. The Free Dictionary[124] gives three views:

operating with minimal human intervention; independent of external control – this is quite similar to that for **autonomy** but without requiring associated decisions or objectives.

like the unthinking functioning of a machine; "an automatic `thank you'" – this view is at the heart of the meaning of automatic; a machine that does not "think", but just does when instructed.

[124] The Free Dictionary by Farley. Accessible at http://www.thefreedictionary.com.

without volition or conscious control; reflexive – again this view implies a lack of connection between action and reason.

This is the MoD's JDN 3/10's definition for an automated system:-

'In the unmanned aircraft context, an automated or automatic system is one that, in response to inputs from one or more sensors, is programmed to logically follow a pre-defined set of rules in order to provide an outcome. Knowing the set of rules under which it is operating means that its output is predictable.'

In summary, if we consider autonomous and automatic modes of operation, especially where a human controller is involved we can consider the following differentiators:

- Automatic Systems are pre-programmed in their actions - the operator invokes a control mode, the machine follows the control mode program until it is instructed to do something else or finishes
- Autonomous Systems have choice in their actions - the operator invokes an authority to make a decision, the machine invokes the control mode it has decided upon

The above encapsulate the views of an automatic machine as being something that does not think, does not decide, nor acts according to an agenda, whereas an autonomous system should be capable of at least one of these, and preferably, all three. It also re-forces the view that autonomous machines should be authorised to make decisions. This may mean real time, operator-in-the-loop control authority, or pre-authorised control, perhaps even hard-coded at the design and implementation stage.

Appendix B Theories of Decision Making

B.1 Classical Decision Theory

Classical decision theory, variously known as Utility Theory or Rational Decision Theory, is based on the concept that a decision maker will choose the course of action, from a set of two or more alternatives that maximises the expected utility. Expected utility is a function it is based on a combination of pay-off, the likelihood of occurrence and the risk attitude of the decision maker. Keeney and Raiffa [59] identify classical (and simple) decision analysis as a 5-step problem:

1. **Pre-analysis.** Identification of the problem and the viable action alternatives.
2. **Structural Analysis.** Information gathering on the choices available and the appropriate structuring of these into, say, a decision tree. This results in identification of decision nodes (under the control of the DM), and chance nodes (under the control of external influences).
3. **Uncertainty Analysis.** The assignment of probabilities to the branches emanating from the chance nodes. This may be by subjective assessment, past empirical data, expert testimony etc.
4. **Utility or Value Analysis.** Utility values are assigned to consequences associated with paths through the tree. Each path will have a consequence. These are ranked in preference by the DM. By combining each preference with its associated probability, a statement of expected utility for each course of action is determined.
5. **Optimisation Analysis.** The analysis is completed when the DM calculates the optimum path through the decision tree. This sequence of preferred actions is the one that maximises the expected utility. In terms of decision theory, it is his optimum strategy.

B.2 Naturalistic Decision Making

Newer models of decision making try to capture aspects of human decision making such as: creativity, adaptability, impulsiveness, confidence etc. some of these ingredients as well as overcoming some of the shortfalls of Utility Theory. These models which try to solve real world problems involving human decision processes are often referred to as n*aturalistic*.

Naturalistic models reflect the fact that in the real world, data is incomplete, problems are ill-structured, goals are shifting, ill-defined, or competing, a multitude of interdependent decisions are required, time stress is often intense; stakes are typically high and multiple players are involved [60]. The research into naturalistic decision models has shown that in the above circumstances, the decision maker falls back on using his experience and the available data, to build a mental model of the environment. Projection of this model into the future on a 'what-if' basis and subsequent evaluation in terms of immediate objectives form the basis for the decision for a course of action. In addition, the best course is not always chosen. Frequently, decision makers have been shown to accept the first satisfactory course of action to be considered; a process known as *satisficing*.

All this indicates that the first main objective for a decision maker is the achievement of the correct mental model of the relevant world; a state which has come to be known as *Situation Awareness (SA)*. The next step is to project that into the future and predict outcomes given possible actions and re-actions.

B.3 Situation Awareness

There is no single unified view or theory of SA. One of the most quoted definitions comes from Endsley [61] who suggested: "the perception of elements in the environment within a volume of time and space, the

comprehension of their meaning, and the projection of their status in the near future". This definition can be considered to encompass three levels of SA, viz.:

- **Level 1 SA:** The identification of key elements, events or facts. Endsley uses the word, *perception,* which implies a belief in the in the mind of an observer. Therefore, each fact or element can be ascribed an underlying level of truth or uncertainty.

- **Level 2 SA:** The generation of appropriate explanation(s) for the existence of the above elements. Endsley describes this as comprehension. Using a simple metaphor of *cause and effect* , this can also be seen as hypothesising about the cause of the above effects. As such, each hypothesis will also have an associated level of belief or uncertainty.

- **Level 3 SA:** The prediction of future events and situations based on the perception of the current situation. Projection of current beliefs or prediction of the future based on beliefs of the present can also be ascribed levels of uncertainty, this time based on conditional probabilities; i.e. given a belief of Condition A, what is the probability of Condition B occurring or *Bel (B|A).*

If we take Endsley's view, the process is one of Perception, Comprehension and Projection or tuple (Pr, C, Po).

There are two major and influential models describing naturalistic decision making. They both relate to dynamic environments where the knowledge is incomplete and the decision maker is very experienced.

B.4 Recognition Primed Decision Making

This concept was produced by Gary Klein in 1987 [62]. He characterises Recognition-Primed Decision Making (RPDM) as the fusion of two processes: situation assessment and mental simulation. People use situation assessment to generate a plausible course of action and mental simulation to evaluate that course of action. Courses of action are evaluated relative to their applicability in a given situation, rather than to their preferability with respect to other possible courses of action. That is, the first satisfactory course of action is selected, rather than the "best" course of action. Such a process is known as *satisficing*.

In other words, the major effort of the decision maker is focused on achieving good SA; by weighing the evidence, seeking new facts and forming a mental model of the situation. Once this achieved to a satisfactory level, the decision maker draws on his experience to select the appropriate course of action. The model also explains the notion of 'jumping to conclusions' where action is initiated without achieving satisfactory SA.

In utility theory, high quality SA is assumed at Stage 1. RPDM can be considered as a complementary precursor to the decision theoretic approach Therefore it may well prove to be that that RPDM, combined subsequently with utility theory, if time is available, is a more complete basis for decision making, particularly machine based decision making.

B.5 The Observation – Orientation – Decision – Action Cycle

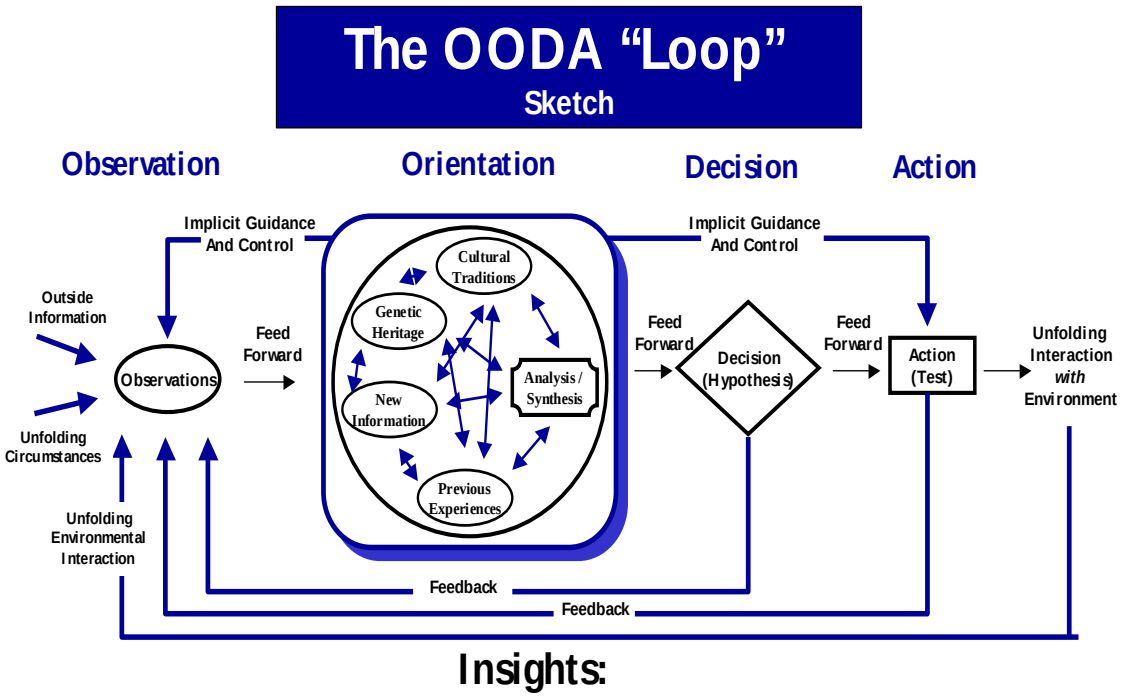
The concept of the OODA Cycle was first introduced by John R.Boyd [63]. He was top rate fighter pilot in the Korean War, a military historian, an important influence on U.S. (particularly the Marine Corps) military doctrine and, if he had ever formally published his work, he would have been a distinguished

academic. The cycle provides a simple, though impressively complete, model of the process of decision-making.

The cycle has since been used as the basis for the analysis of air combat effectiveness by Bartolamasi [19] where it is alternatively referred to as the 'Information – Decision – Action (IDA)' cycle.

In recent years, poignantly almost immediately after Boyd's death in 1997, the OODA cycle has also been applied to business decision making. In fact, the cycle applies to any system in a dynamic environment, capable of some form of situational cognition, selecting and deciding on an appropriate response and implementing the chosen course of action.

An original Boyd version of the OODA cycle is shown, in its final incarnation dated 1996, in Figure 6, below:



Insights:

Note how orientation shapes observation, shapes decision, shapes action, and, in turn is shaped by the feedback and other phenomena coming into our sensing and observation window.

Also note how the entire "loop" (not just orientation) is an on-going many-sided implicit cross-referencing process of projection, empathy, correlation and rejection.

Figure 55: Boyd's OODA Cycle

Information is collected and collated. Orientation then takes place, which results in the information being placed into a context. This context provides the decision maker with a mental model of the current situation which can be assessed with relation to his objectives and his current plan. At some point, the decision maker (DM) will be faced with the choice of selecting a suitable course of action from two or more alternatives.

Measures of overall effectiveness can be ascribed to the scope of an individual cycle, and the speed with which it is conducted. These are collectively known as the Tempo of Operation. Boyd suggests that the objective of a competitor is

to conduct operations at a tempo greater than his opponents and is a necessary condition for winning.

During earlier work on the performance of an intelligent agent [17], it was shown that operating the agent at significantly higher decision rates brought about a corresponding increase in performance. Whilst no claim was made for the proof of Boyd's theory, the findings represent strong evidence for this.

Appendix C Example System Architectures

C.1 Example System Architectures

The following provides examples of architectures that can be considered as contributing to, or as candidates, for the proposed UAS decision architecture. In each case a description of the architecture is given together with an assessment of their suitability.

C.2 Example Control Architectures

C.2.1 4D-RCS [64]

Description

The 4D Real-time Control System (RCS) architecture was designed for sensory-interactive robotics where the emphasis was on combining commands with sensory feedback so as to compute the proper response to every combination of goals and states. It has evolved from RCS-1 (mid 1970s), where the application was to control a robot arm, through RCS-2 (used for manufacturing control), to RCS-3, where the application was an autonomous undersea vehicle. 4D-RCS has been used for the control of Experimental Unmanned Ground Vehicles (XUVs) and were demonstrated in a series of field tests in 2002/3 [65]. The prefix “4D”, refers to the application of Dickmanns’ 4D approach to human vision [66] within the RCS control architecture.

RCS purports to be a cognitive architecture and models the brain as a hierarchy of goal-directed sensory-interactive intelligent control processes. As it is a reference architecture, these processes are not defined or specified and may be implemented in a variety of means such as neural nets, finite state automata or production rules [67].

The RCS architecture consists of a hierarchy of nodes, where each node represents and operational unit at different levels of granularity. Each node contains four processing units: Behaviour Generation(BG), World Modelling(WM), Sensory Processing(SP) and Value Judgement(VJ).

A schematic of an RCS node is shown below:

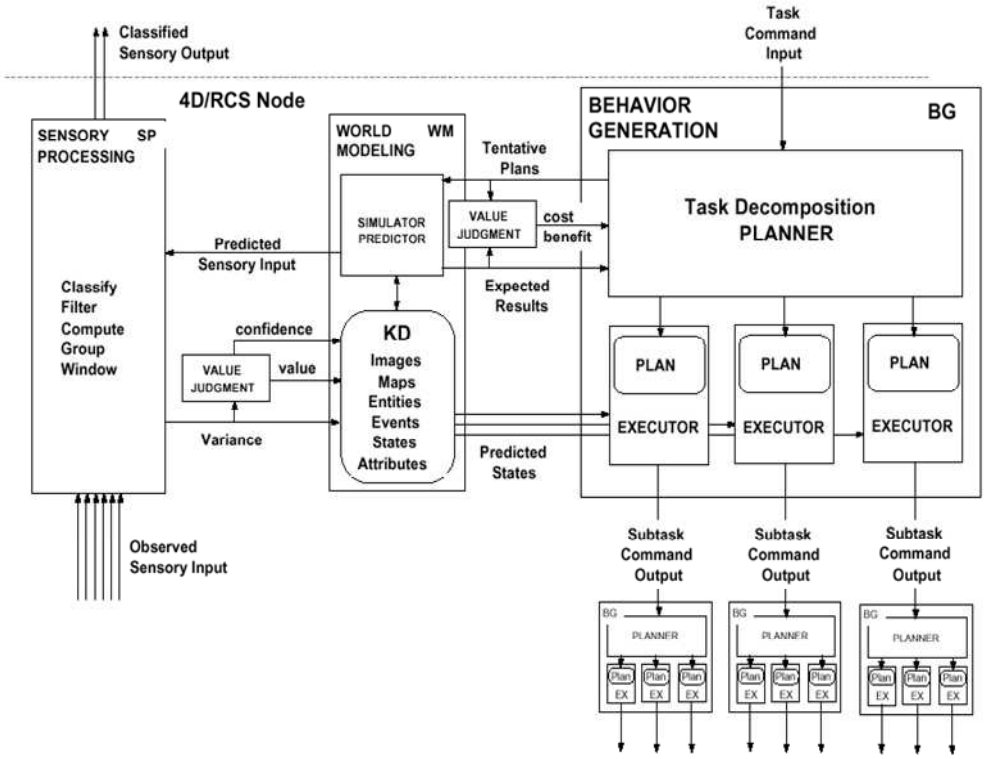


Figure 56: A 4D-RCS Node

The Behaviour Generator (BG) is the deliberative or planner process and consists of a planner and a set of executors. Higher level tasks are decomposed and sub-tasks are generated. These are cascaded down through the system until primitive tasks are generated and applied to servos. Thus planning and execution are tightly knit. VJ is applied in two ways: firstly to

sensory processing to determine value and confidence, and secondly to plans to determine the expected results of tentative plans. The WM maintains the KD at the appropriate level that is the best estimate of the external world. The hierarchy of WM processes in the architecture represent the overall system KD. Knowledge data is therefore accessible by design throughout the node hierarchy. The operator interface covers each level in the hierarchy.

A specific feature of this system is that, because of the nature of the hierarchy of nodes and the fact that similar processes exist at each level in the hierarchy, though at different granularities, each level corresponds to a different time/distance window. At the highest level, plans may be for a 24 hour window for, say, 5000sq km maps, whilst at the lowest level, plans for servo actuation may exist for only 20 milli-seconds or correspond to a single pixel. Again at the highest level, the task is defined by the top level mission goal. At successively lower levels, the task is further decomposed into sub-tasks and finally, at the lowest levels sub-tasks correspond to primitive actions for the controls, servos and actuators to respond to.

The basic model that runs whilst the planning period is open is as follows:

1. The BG planner hypothesizes a tentative plan,
2. The WM predicts the probable result of this plan
3. The VJ evaluates the probable outcome value
4. The plan selector within the BG planner checks to see if this result is greater than the previous probable outcome value of the plan that is already in the current best plan buffer, and,
5. If it is, then the tentative plan replaces the current best plan in the plan buffer. If it is not, the program continues to the next cycle.

6. At the beginning of the next cycle, the contents of the plan buffer is moved into the executor plan buffer and
7. Re-planning commences.

The above occurs at every level in the RCS hierarchy within the spatial/temporal horizon of that level. Because the plan and executor buffers are separate, the planning sequence can operate asynchronously (i.e. slower) than the execution process. This is claimed to provide not only deliberative and reactive processes, but also a pre-emptive process. Furthermore, it also allows for these processes to be distributed throughout the hierarchy.

The authors claim that the architecture addresses the three key issues which theoreticians [68], [69], [70] have proposed as major obstacles to meaningful machine intelligence:

- **Abductive Inference** – the ability to reason from consequent to antecedent. This is (allegedly) enabled by the top-down structure of the hierarchy driven by mission goals, and the in-built process of assessing consequences (or outcome values) prior to committing to a plan.

- **Symbol Grounding** – the establishment of a direct correspondence between internal symbolic data and external real world entities, events and relationships. This is apparently achieved (by analysis of RCS schematics, not by authors claims) by classification of sensor inputs, generation of confidence levels/values and prediction of next sensory inputs. In short, the establishment of coherent Situational Assessment in the accepted form of Endsley (see Appendix B).

- **The Frame** Problem - the problem of predicting what in the world changes as a result of action and what stays the same. The view is that this in itself is caused by modelling the world entirely by logical propositions. Humans (and other animals) of course do not do this and the RCS authors claim that their architecture, in its support for image analysis and models of human vision, do not also.

Suitability

RCS is an interesting architecture and apparently highly suitable for the control of a UAS for several reasons:

- It has been designed from the outset as a real time control system and has been used to develop the autonomous control system for an All Terrain Vehicle and an autonomous crane, the NIST RoboCrane [71]. These have both been successfully demonstrated.
- BG clearly contains the TLA elements of Planning, Sequencing and Control and these are augmented with information processing, SP and WM.
- At a higher level of abstraction, RCS is a methodology for designing a control system as well as a reference architecture. As such, it does not specify functionality or partitioning, software implementation or even make statements about what environments it is suitable for. It does not therefore specifically exclude or inherently rule out operation in airborne environments.
- RCS does implement certain standards, defines roles and responsibilities for its control levels and applies sound hierarchical and control principles. At a lower level it provides engineering guidelines for building and testing intelligent vehicle systems. It is therefore much more than an academic

exercise for the research of autonomous systems and clearly much thought has gone into its use for engineered products.

However, there are some negative aspects:

- Although the elements of a TLA are present, the way in which these elements is combined in terms of process are not wholly in keeping with the principles underlying a TLA. In fact, RCS is not a layered architecture in that sense but something quite the opposite. In RCS, each node contains the three layers and it is the nodes themselves that are layered at different levels of granularity (time, distance and image). In theory at least, even a servo operating at 50 Hz can have a deliberative planning layer and world modelling, which would normally be considered quite ridiculous. Such a system is fundamentally different from the precepts of the TLA. However, that is not to say it is unsuitable.
- There is no reference to the ability to make decisions in different ways. The fundamental reasoning system in RCS is deliberative and whilst it should be simple to closely couple the SP to the control executors by bypassing the WM and deliberative processes, it should be noticed that the scheduling, as described by the model design and node schematic above, is driven by the output of the planner. In short, the architecture is set up for involving planning at all stages. Having said that, the inclusion of a reactive process should be a simple affair. However, given that the architecture is designed specifically for intelligent control, it would have been preferable to see alternative decision processes, such as those outlined in Figure 8, to be far more explicitly embedded in the architecture.
- RCS is available as a class of libraries in C/C++, JAVA and ADA. There may be aspects of certification which could preclude the use of these

libraries within a UAS.

C.2.2 Coupled Layer Architecture for Robotic Autonomy (CLARAty)

Description [72]

CLARA is an architecture developed by the Jet Propulsion Laboratory (JPL) for developing robots to be deployed on planetary missions but can be applied to almost any vehicle. It has been developed in response to the following perceived needs:

- To achieve acceptance from a wide community, but primarily from the robotics and autonomy communities.
- To bridge the gaps between user, developers and academia operating to implement a diverse range of solutions to achieve differing types of problems.
- To leverage existing software in research and NASA flight efforts.
- To leverage standard practices in industry to avoid continued re-invention of the wheel and enable NASA robotics efforts to adopt techniques and solutions commonly used in commercial products.

The developers of CLARA are critical of the standard TLA, the layers of which they describe as “Functional, Executive and Planner” and which they perceive as increasing in intelligence from reflexive, to procedural, to deliberative. Specifically, the following criticisms are made:

- Since the limits of each layer are not strictly defined, they note that researchers tend to expand the capabilities and dominance of the

layers in which they particularly interested. This, they feel, results in systems which are unbalanced. They also note that there is still considerable research which blurs the line between the Planner and Executive and which questions the hierarchical superiority of one over the other.

- They feel that there is a lack of access from the Planner to the Functional level and that often means that two separate world models, which may not be consistent, are used.

- They are critical that researchers tend to associate the concept of increasing intelligence with that of increasing granularity in time and scope (such as that at the heart of RCS). This they feel, tends to obscure the hierarchy that can exist within each of the system levels.

To correct these perceived deficiencies; they have engineered a two-tiered architecture where the Planner and Executive are closely coupled, and have a common database, to form a single tier. These two layers are called the Decision Layer and the Functional Layer. Maximum use of object oriented approaches are made, particularly in the functional layer, and this is used to drive the concept of granularity, both in time and function, as a third dimension to the two layer hierarchy. Objectives are received and de-composed into sub-goals which are expressed as constraints. From these constraints, tasks are derived[125] which are sequenced into commands. These commands are passed to the functional layer for execution. Within the sequencer there is strong emphasis on resource management for prioritisation of activities. Clearly there

[125] Example: An objective would be to “make the joint angle not less than 20 degrees and not greater than 30 degrees. For goals, a similar example may be to make the joint angle 22 degrees, given the above constraints.

is no point in scheduling activities for which there is little or no resource availability.

In practical terms, CLARA merges the planning and sequencing layers (or blurs the boundary between them, depending on your viewpoint); the functional layer is fairly consistent with the conventional TLA control layer.

Suitability

Some of the aims of the CLARA project are laudable; in particular, in trying that of trying to bridge the gap between users, engineers and academia; in some ways they have been successful. CLARA has been used to build highly autonomous “planet explorers” yet clearly has a strong theoretical underpinning.

The main reason for merging the two top layers appears to have been driven by the planners need for ready access to functional resource information, (which would normally go via the executive). In the case of a small robot, with limited power and thousands of miles from Earth, this is clearly an important consideration. This is unlikely to be true for a UAS, particularly a large one.

However, on inspection of the documentation, it cannot be denied that the designers have really not produced anything startling or innovative. In fact, they have criticised robotic theorists for having uneven balances between the planning and executive layers and thus merged the two. They then go on to state that CLARA has considerable flexibility by having, at one end of the spectrum, a capable decision layer but basic functional level and at the other end, the opposite situation – i.e. a similar imbalance at the functional and (combined) decision layer. In addition, they freely admit that there is a fuzzy partition in the decision layer where the planner is dominant and the executive is dominant. Again an area which they themselves criticise others

The documentation, which is considerable but lacks detail and is readily obfuscating, in the CLARA project would require that a trial implementation would have to be built to substantiate (or otherwise) the designers claims. Unfortunately, although early software libraries (written in C++) are available to preferred US or Class B countries outside the JPL, the later versions are not.

C.2.3 Integration of Reactive behaviour and Rational Planning(InteRRaP) Architecture

Description [73]

Interrap, as its name implies, was designed from the outset as a hybrid agent architecture, combining fast reactive behaviour with slower deliberative plan based behaviour. In addition, it was developed for co-operative behaviour between multiple agents. It is somewhat older than other architectures presented here having been developed circa 1993 and therefore somewhat before the development of the TLA with which it shares many common features. A schematic of the Interrap agent model is shown below.

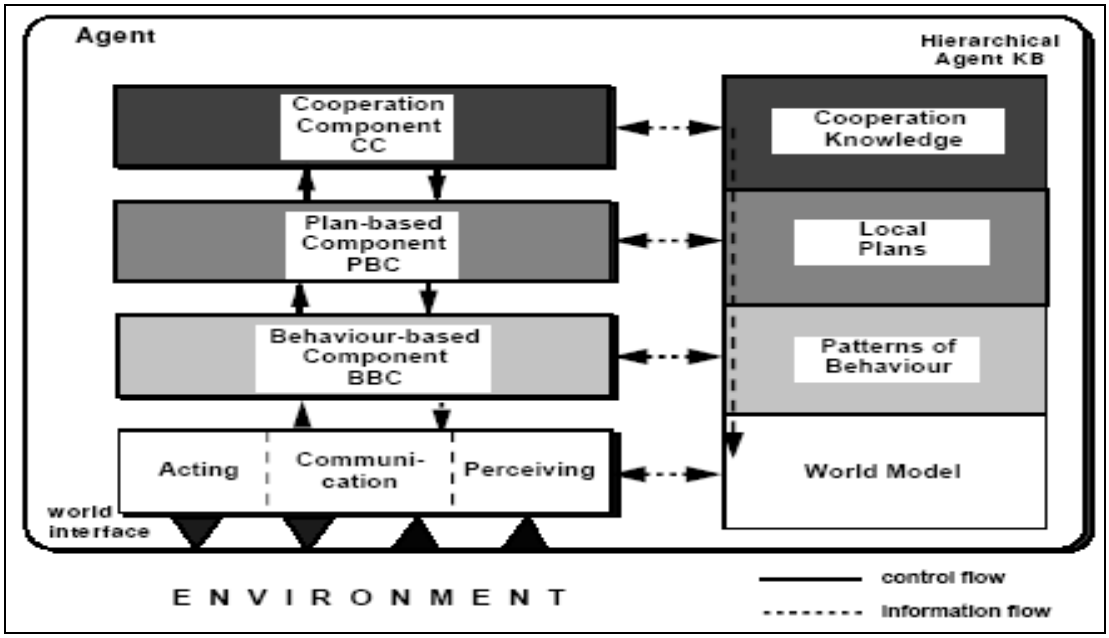


Figure 57: The InteRRaP Agent Model

Interrap is a layered architecture with each successive layer representing a higher level of abstraction. However, interestingly, there is also a division of these layers into those relating to knowledge and control. The lowest layer, the one which relates to the environment, comprises the world interface to the agent which contains functionality for acting, communicating and perceiving. Acting relates to that part of the agent which physically interacts with the environment. Communication consists of ingoing and outgoing messages, and Perceiving covers sensing and information processing. The knowledge portion of this layer is the world model.

Above the world interface layer is the behaviour based component which implements and controls the reactive behaviour of the agent which is defined by the Patterns of Behaviour (PoB) database in the knowledge hierarchy. This database is two dimensional and the position of a PoB determines its priority. A PoB represents procedural knowledge i.e. mechanisms which are not represented in a declarative manner but which describe procedures to be followed. These are obviously pre-compiled and are there to activate routine behaviours which do not require deep reflection of complex planning[126]. Each PoB has an activation pre-condition, success/failure metrics, a post condition and an executable body that defines the actions to be performed. These actions may be low level primitives, other PoBs or calls for higher level planning.

Above the behaviour based components is the plan based component consisting of a planner and a plan library. However, in planning, the planner will always check to see if a PoB exists which will satisfy its goals, or part of them, and if there is it will use it.

[126] Such as opening a door, starting the car engine, putting a hat on etc.

At the highest level is the co-operation based component which can generate joint plans that satisfy the goals of a number of co-operating agents.

Control is both data and goal driven. Perceptual input from the sensors will result in changes in the world model which will in turn trigger PoBs. The triggering of these PoBs may require the planner to be asked to develop new plans to achieve the goals of the agent. These again will result in primitives and messages at the world interface.

Suitability

Interrap, for all its age (which in the field of robotic architectures is equivalent to being pre-historic) was clearly well ahead of its time in its thinking and design. It should easily be adapted to incorporate lessons learned from the development of the TLA.

It is particularly relevant to a UAS application for many reasons:

- It is highly likely that a UAS would have fairly substantial sensors and processing in order for it to achieve complex tasks. This corresponds to the world interface, which for a UAS, would be the flight control and communication system normally called (by UAS designers) the sensor and effector suite of the system. The world model can regarded as an information processing system which models the state and status of all external and internal objects.
- Our decision system, ideally, would have multiple decision processes. These are supported, in principle at least, by Interrap.
- It is, explicitly, a control architecture by design but also implements planning and re-active components.
- Control in the air domain is extremely procedural. Many aspects of

control are repetitive and common. The concept of PoBs[127] in Interrap which specify routine behaviour would be highly beneficial to adopt.

- The concept of achieving mission success naturally leads to a goal driven approach. The re-action to uncertain events occurring in the world (internal and external) is also a fundamental requirement. Interrap can intrinsically handle both.

C.3 Other Robotic Architectures

There are many other robotic architectures, some of which are worthy of greater interest and inspection. These are mentioned here but may be researched in greater detail in follow-on studies. Some examples are:

C.3.1 JAUS[128]

The Joint Architecture for Autonomous Systems has a primary, and some would say overriding, aim of providing a standardised architecture across all autonomous platforms for reasons of interoperability. In that sense the designers share a similar desire to those of CLARAty. The main feature of JAUS is the standardised message passing interface independent of functional design and hardware. It is not known how mature or developed the architecture is but a small search has failed to reveal whether any design team has actually used it.

[127] The concept and design of the PoB structure could be improved and made more sophisticated. The pre-condition as it stands is essentially a rule based trigger. It could be modified to be the entry conditions for a case based system which will search though the case base and return the PoB with the best goodness of fit.

[128] Joint Architecture for Unmanned Systems home page is at http://www.jauswg.org/

C.3.2 Player[129]

Player is a not an architecture in the accepted sense but a device server, widely used in the robotic community, based on the TCP socket client/server model. It is therefore software language independent.

C.3.3 OSCAR[130]

OSCAR (Operational Software Components for Advanced Robotics) is an object orientated framework for the development of robotic control programs. It is similar to CLARAty in its use of object oriented decomposition but much less of an overall architecture.

C.4 Example Agent Architectures

C.4.1 Jack

Description [74]

JACK is commercial agent programming language and development environment that exists as a superset of the JAVA programming language. It follows the same principles as Object Orientated programming in that, by encapsulating desired behaviours in standardised modular classes, reliable, lower cost and scalable development can be more easily realised. Using an agent based approach extends that encapsulation. The agents in JACK follow the theoretical Belief-Desire-Intention (BDI) model of Georgeff and Rao***75***. This model describes deliberative behaviour though the agents acting according to its desires (goals or objectives), which force the generation or instantiation of

129 *"Overview of Player, Stage and Gazebo"*: Robotics Research Laboratory, Center for Robotics and Embedded Systems (CRES, University of Southern California available at http://robotics.usc.edu/?l=Projects:PlayerStageGazebo

130 *"OSCAR Overview"*, Robotics Research Group, University of Texas, Austin, USA available at http://www.robotics.utexas.edu/rrg/research/oscarv.2/.

suitable intentions (plans) according to what beliefs (world model) it has about its environment. In addition, JACK agents can also act in response to events and therefore exhibit reactive behaviour.

Suitability

A fundamental problem with JACK is the JAVA basis. Although JACK has been used as the autonomous controller for a tactical UAS, the Avatar UAS built by Condorra Ltd., this vehicle is a small (5 ft. wingspan), lightweight (7.7lbs) UAS and was demonstrated on a test range[131] in Australia in 2005. In that demonstration, JACK was run on a HP Ipaq PDA and provided the control advice to the vehicle system.

However, at the moment, JAVA is considered unsuitable for airborne systems operating under CAA rules in UK airspace and would be unlikely to gain full certification[132] for any autonomous control actions specified by a JAVA based system, particularly those involving safety at any level. This is unfortunate because in many ways JACK fulfils most of the requirements. In particular, JACK's capacity to specify plans in a scalar and modular manner, would be highly advantageous compared to other ways of writing software based plans. This latter point is discussed further in Appendix E and, whilst not fully pertinent to this thesis, a Mixed Language Architecture is proposed for study elsewhere.

[131] The certification requirements for operation on a test range under carefully controlled conditions are simple in comparison to that required for unrestricted operation in UK airspace (which is quite crowded when compared to Australia).

[132] As advised by Andrew Miller, Senior Certification Engineer and Tony Hopwood, Group Leader, System Safety Software Group, BAE SYSTEMS, Warton.

C.4.2 The JAM Architecture[133]

Description

JAM is an intelligent agent architecture developed in academic research by Dr Marcus J Huber. Like JACK it is JAVA based and based on the Belief-Desire-Intention (BDI) paradigm. Every JAM agent is composed of five primary components: a world model, a plan library, an interpreter, an intention structure and an observer. This is depicted in the figure below:

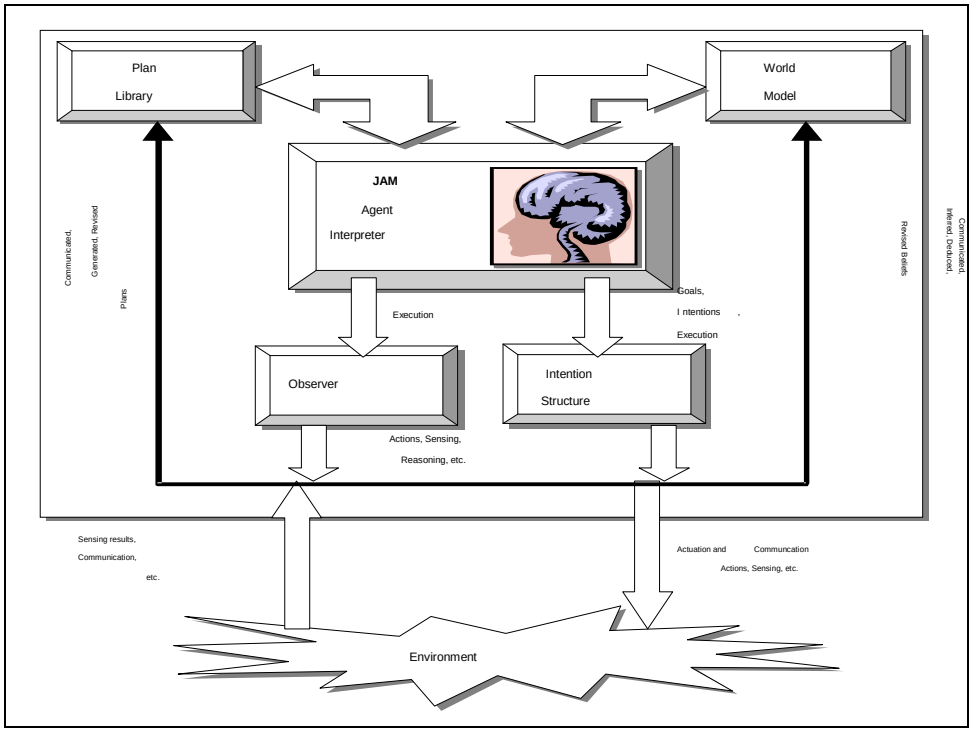


Figure 58: The Jam Architecture

The observer is a procedure external to the interpreter's main body which is primarily used to update the world model. Changes to the world model, or

133 Marcus.J.Huber. Extracts reproduced, with permission, from the *"JAM User Manual"* available at http://members.home.net/marcush/IRS/Jam/Jam-man.html

posting of new goals, triggers reasoning in the JAM interpreter to search for plans that may be applied to the current situation. This is done by filtering out plans that do not have acceptable pre-conditions or context fields. Once the Applicable Plan List (APL) has been assembled, the interpreter selects the plan with the highest utility and intends it to the goal (i.e. commits itself to execute the plan) by placing it in the intention structure. If the intended goal has the highest utility among all goals with intentions, then that plan is executed. However, if a previous intention has a higher utility then that plan is executed instead. As the utilities and contexts of the various goals and plans change, the agent will switch between goals in order to pursue the highest priority goal.

The JAM agent consists of a text file specifying the plans, an initial world model, initial top level goals and an Observer procedure. JAM low level functionality is extended by writing appropriate primitive functions in Java. When a JAM agent has completed all its goals, it stops.

Suitability

JAM has the same JAVA limitation that JACK has. However, unlike JACK, it is non-commercial and support for it from the originator has now ceased. It has been used in previous work [17] and suffers from several limitations. These are:

- Speed – the JAM interpreter is very slow. Tests, undertaken in previous work, show that compared, to a reactive mechanism, the JAM interpreter runs 30x slower and this was with a simple plan structure (no more than 5 concurrent plans).
- There is no reactive mechanism built in to the interpreter.

- The interpreter's interface with other parts of the system is (solely) by passing strings.
- The value model associated with plan assessment routine is a simple scalar model.
- It operates with a fixed and pre-defined plan set.

However, in terms of its ability to select appropriate plans, choose the best one and implement it, it would suffice but at a performance level well below that which is required.

C.5 Example Avionic Architectures

C.5.1 Eurofighter Typhoon Avionic System

In Para 3.2.7.1, the nature of the Eurofighter avionic system architecture was commented on. It is useful to examine this as an example the avionic architecture somewhat further and see if there any parallels with those from other areas such as robotics. In addition, it is an example of a complex multi-functional system which has been designed, implemented and certified.

The Eurofighter avionic system functional partition is composed of seven major sub-systems:

- Attack and Identification
- Navigation
- Communications
- Integrated Monitoring and Recording System
- Displays and Controls
- Armament

- Defensive Aids

The software in these sub-system partitions is written to a minimum of SIL 3 standard and is distributed among several computing centres which communicate via 1553B and other databuses. The layout of the Typhoon avionic system is shown below:

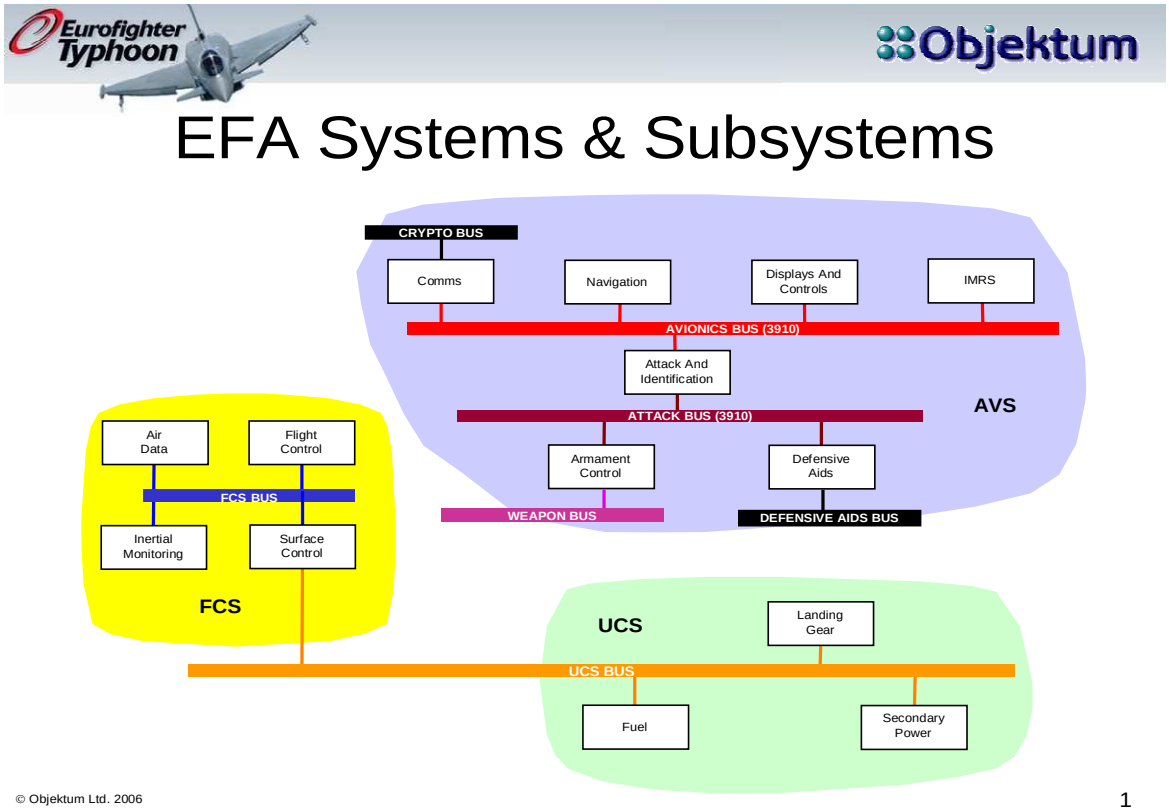


Figure 59: Schematic of the Eurofighter Typhoon Avionic System (AVS)

C.5.2 The J-UCAS Architecture and Common Operating System [*76*]

Description

The Joint Unmanned Air Combat System (J-UCAS) project started in October 2003 and was a joint programme between the US Defence Applied Research

and Procurement Agency (DARPA), the US Air Force and US Navy. It required the development of a family of UCAVs to demonstrate concept feasibility and operational assessment.

The programme specifically focused on achieving the following:

- Operator to Vehicle ratios significantly better than 1:1.
- Operation in challenging and dangerous environments
- Mission planning times reduced from days for single ship, to hours for multi ship operation.
- Improved communication management
- Multi ship co-operative targeting and attack compared to non-co-operative single ship reconnaissance missions.

The family of development vehicles was designed to use a common core system, known as the Common Operating System (COS), which was intended to form the basis of a standardised autonomous system. The COS represented the architecture, algorithms and software to:

- Control and manage system resources
- Facilitate information exchange
- Provide battle space awareness
- Enable inter-platform functionality
- Enable autonomous operation
- Maintain quality of service

The J-UCAS notional decision architecture is described at Reference [77] and represented in the schematic below:

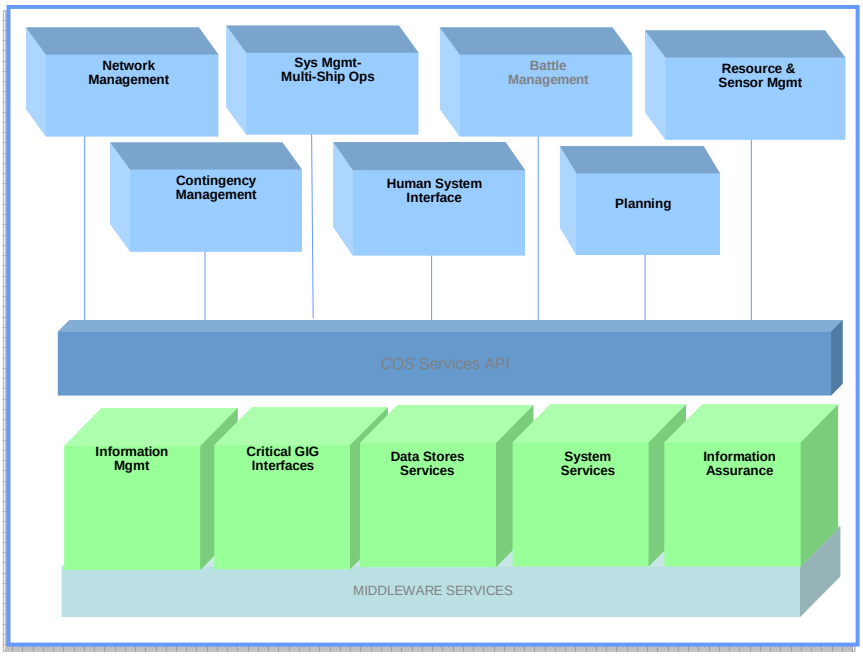


Figure 60: J-UCAS Notional Decision Architecture

Again the elements of Execution, Planning and, in particular, a strong emphasis on Information Processing are to be seen. Note that is only a decision architecture and the control element, though implicit for plan execution is not shown.

The *Common Operating System* is also described at Reference [76] and provides the autonomous system 'intelligence' for the overall J-UCAS. The *Common Operating Systems (COS)* enables interoperability among multiple air vehicles and control stations, facilitating the integration of other system components such as sensors, weapons, and communications. The *COS* encompasses the software architecture, algorithms, applications and services

that provide command and control, communications management, mission planning, much of the interactive autonomy, the human systems interface and the many other qualities associated with the J-UCAS system [76].

The COS provides the central autonomous core to the rest of the platform as described below:

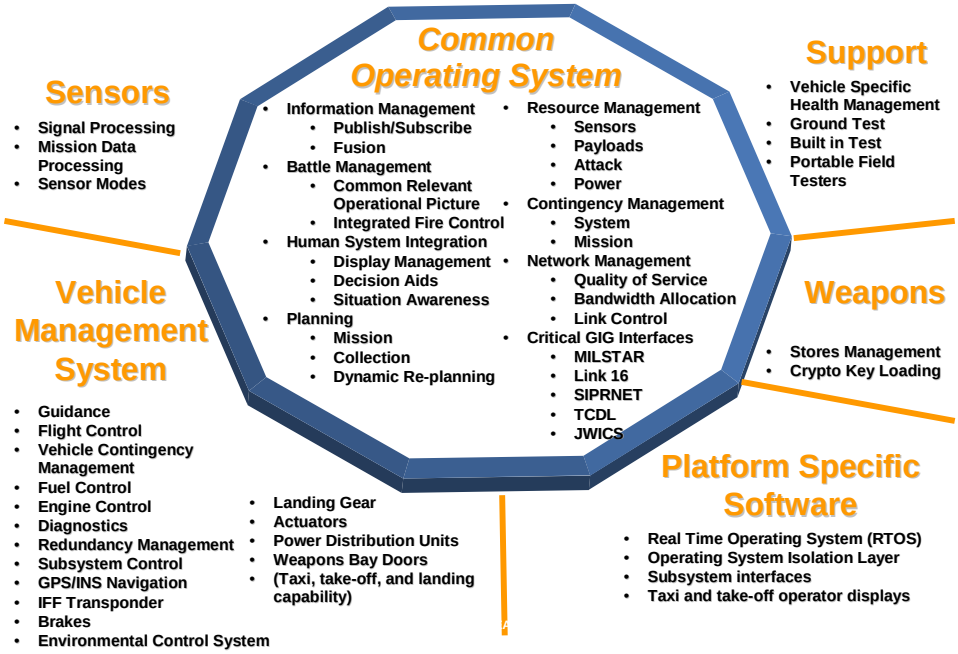


Figure 61: COS Interfaces within the J-UCAS Platform

Compliance and Suitability

The notional architecture closely follows the TLA but includes an extra layer for information distribution and management. Planning appears to be confined to routing and contingencies, but could easily be modified to include other areas. It appears that there is complete connectivity between all modules. However, as

this is a notional architecture and it is not clear whether it was ever built or developed further, it is hard to say whether this architecture, including this complete connectivity, is valid.

The COS, however, appears to be much more thought out in terms of functionality and represents a complete view of its integration with the rest of the vehicle systems.

Appendix D The Synthetic Environment Based Acquisition (SEBA) Process for System Implementation

D.1 Features of SEBA

D.1.1 The Spiral Development Process and the SEBA Wheel[78]

SEBA is predominantly concerned with the construction and management of an evolving set of Synthetic Environments (SEs), Models and Simulations (SEMS), of increasing complexity. The aim is to mitigate risk by simulating, modelling and emulating the necessary variables to ensure the equipment target remains within the specified boundaries (ordinarily of time, cost and performance).

All the information used and derived through the SEBA approach is held within a central Knowledge Repository, which may take the form of a Shared Data Environment (SDE) or Advanced Collaborative Environment (ACE) and is accessible by all stakeholders but partitioned to protect national security and commercial sensitivities. With the use of suitable configuration control, a data audit trail can be achieved and the components of decision-making traced. A knowledge repository also facilitates the management of resources and tools, such as the potentially large numbers of integrated models and simulations and their associated volumes of data.

Spiral development delivers an increasingly more detailed concept design, which requires more detailed analysis to support acquisition decisions. Within the 'SEBA wheel', Figure 1 below, the inner wheel is 'spun' quickly; possibly many times for each step progression around the outer wheel, which represents the phases of the acquisition process. Each revolution of the inner wheel evolves and increases the fidelity of the knowledge appertaining to a solution; this knowledge is captured in the expanding central Knowledge Repository.

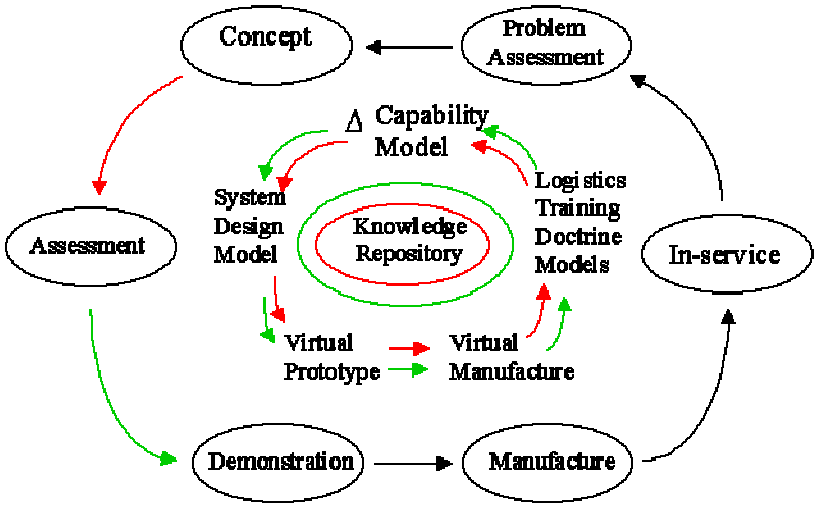


Figure 62: The 'SEBA Wheel'

D.1.2 Simultaneous Testing[79]

SEBA enables early integration of the system design. This not only reduces risk from the outset but will allow equipment testing and capability testing to take place throughout the concept, assessment and design stages of the systems engineering process. This is in contrast to traditional practices, where testing tends to take place as a final stage of the development process. Within SEBA, testing is an integral part of the design process rather than a final quality assurance check mechanism.

D.1.3 End User Involvement[79]

Synthetic environments allow the end user to be involved in the acquisition process in a full and meaningful way from the beginning of the acquisition lifecycle. This is in contrast to traditional programmes where users may be consulted but would rarely be a part of the design team and would probably not get to influence the development of capability. In SEBA the user can be involved from the outset by using models of the proposed equipment in a virtual

world which therefore allows equipment development to occur at the same time as doctrine development. In the past, the separation of the two has led to a capability gap. With SEBA, the equipment becomes an effective component of defence capability much sooner than previously would have been possible.

D.1.4 Iterative Modelling[79]

SEBA allows a process of iterative model development and integration, leading to greater understanding and insights earlier in the CADMID process. The evaluators can use simulations to explore the design space, to validate designs taking into account any restrictions/limitation of the target environment, and to converge upon a capability that meets the need. This is done in an iterative fashion, rather that at one particular stage, which means that the capability is developed in a much quicker and more efficient way.

D.1.5 Experimentation[79]

An acquisition process supported by models, simulations and synthetic environments allows design teams to concurrently explore greater numbers of potential solutions than is currently possible using traditional engineering means within a reasonable timeframe. On the other hand, a smaller number of designs may be pursued to a greater depth, equivalent to what is currently known as Rapid Prototyping. It also allows the investigation of several different, widely varying scenarios, giving confidence that the developed capability will be useful and relevant in all envisaged scenarios. Equally, SEBA allows the exploration of the design space through time, allowing the full lifecycle equipment costs to be considered at the beginning. Finally, it allows the proposed capability to be integrated into existing (and/or proposed) defence capability as a part of the design process. This further strengthens confidence that the capability requirement will be met.

D.2 Evolutionary Acquisition & Spiral Development[80]

The publication of the US Department of Defense (DoD) Directive 5000.1 and DoD 5000.2 established a preference for the use of Evolutionary Acquisition[134] strategies relying on a spiral development process.

Evolutionary Acquisition and spiral development are methods that will allow a reduction in the time for the delivery of effective capability. These approaches are incremental in nature and are designed to develop and field demonstrated technologies for both hardware and software in manageable pieces. Evolutionary Acquisition and spiral development also allow insertion of new technologies and capabilities over time.

These approaches provide the best means of getting advanced technologies into Service quickly while providing continual improvements in capability. Evolutionary Acquisition and spiral development are similar to pre-planned product improvement but are focused on providing an initial capability that may be less than the full requirement as a trade-off for earlier delivery, agility, affordability, and risk reduction.

For Systems Engineers, spiral development is a means whereby a baseline system can be developed very quickly with a minimal set of requirements. The development process is then one of increasingly refining the quality, scope and capability of the situated system, allied with more detailed levels of testing. With each spiral, new capability requirements can be developed and included within the design. The performance of this new design can then be measured and

[134] Within the Rapid Engineering environment, this is referred to as Incremental Development.

costed, thus enabling capability driven specification and acquisition within an appropriate contracting environment. Further amplification is covered below:

D.3 Evolutionary Acquisition[80]

Evolutionary Acquisition is an acquisition strategy that defines, develops, produces or acquires, and fields an initial hardware or software increment (or block) of operational capability. It is based on technologies demonstrated in relevant environments, time-phased requirements, and demonstrated manufacturing or software deployment capabilities.

These capabilities can be provided in a shorter period of time, followed by subsequent increments of capability over time that accommodate improved technology and allow for complete and adaptable systems over time. Each increment will meet a militarily useful capability specified by the user.

There are two basic approaches to Evolutionary Acquisition. In one approach, the ultimate functionality can be defined at the beginning of the program, with the content of each deployable increment determined by the maturation of key technologies.

In the second approach, the ultimate functionality cannot be defined at the beginning of the program, and each increment of capability is defined by the maturation of the technologies matched with the evolving needs of the user.

D.4 Spiral Development[79]

The spiral development method is an iterative process for developing a defined set of capabilities within one increment. This process provides the opportunity for interaction between the user, tester, and developer. In this process, the requirements are refined through experimentation, evaluation of design options, trade off analyses and risk management. Thus there is continuous feedback,

and the user is provided the best possible capability within the increment. Each increment may include a number of spirals. Spiral development implements evolutionary acquisition.

D.5 Anticipated Benefits and Concerns of SEBA

D.5.1 Anticipated Benefits

Currently, SEBA remains a largely unproven process. However, there are likely to be significant benefits associated with it if it is implemented as currently envisaged. Amongst these are the following:

Reduced Risk - A reduced overall programme risk, through informed and timely decision making. Additionally, this risk is driven out far earlier in the programme than conventionally.

Reduced Cycle Time - A reduced cycle time within programmes. In essence, the team will be able to iterate through concepts to assessment and design more quickly than traditional methods.

Reduced Ground and Flight Testing - A reduced requirement for exercises and trials during development time, hence reducing costs for time, space and prototypes.

Reduced Development Time and Cost - As a consequence of the above, and because of reduced re-work, a reduction in overall programme time and cost. This should ensure that equipments are brought into service sooner and are hence current for a longer time.

Better Visualisation - The ability to effectively visualise requirements and cost drivers, and their implications.

Improved Flexibility - An ability to adapt quickly and flexibly to changing requirements, scenarios and environments.

Improved Human Machine Interface Design - Support for the inclusion and evaluation of human factors. In essence, SEs support the inclusion of the human at the outset of and right through the development process.

Better Interoperability - Representation of the whole system, user and environment within the wider defence system of systems.

Better Teamworking - SEs involve linking multidisciplinary and multi-organisational teams together. So an SE based approach positively encourages IPTs to work in an integrated manner.

Better System Employment – As development proceeds, it will be possible to continually develop the operational doctrine for its subsequent employment. In parallel with this is the opportunity for pre-service training in terms of operational use. This latter aspect not only covers employment but extends into support and logistics.

All of the above factors, including the fact of early integration of the system, will inspire greater confidence that the system will deliver the desired capability.

D.5.2 Concerns[79]

There are concerns about the SEBA process and its implementation which are expressed in Ref 79. Amongst them are:

Verification & Validation - The user of any model, simulation or SE must have confidence that it is a credible representation of reality. That is, the model must be verified and validated.

False credibility - It appears to be an unfortunate fact that individuals will often attach greater credibility to a SE than the underlying component models themselves warrant.

Configuration Management - The concept of SEBA envisages an iterative, integrated approach to the use of models, simulations and synthetic

environments throughout the lifetime of a capability or equipment. This necessitates advanced configuration management mechanisms, as would be the case in any equipment acquisition programme.

Interoperability and Scalability - Taking a systems of systems view, it is possible that there will be a need to integrate whole SEs to investigate wider defence capability, doctrine and concepts. Hence SEs must be designed with interoperability and scalability in mind.

Cost - The construction of an SE may be a very expensive and time consuming process. The benefit is that going through that process may give you a better capability earlier. The downside is that in the short term it may not seem to be the most affordable solution.

Data Capture and Recording - SEs are used primarily to conduct experiments. Hence these experiments need to be designed using the principles of experimental design in line with the purpose and context of the problem i.e. each experiment has a defined purpose and results. The data need to be captured and recorded as would any other lab experiment.

Appendix E The Use of Jack in a Certifiable UAS Architecture

E.1 Architecture Concept

In developing the proposed solution architecture, a smaller scale model was constructed early in 2006, coded primarily in JAVA. A reference view of this is shown below:

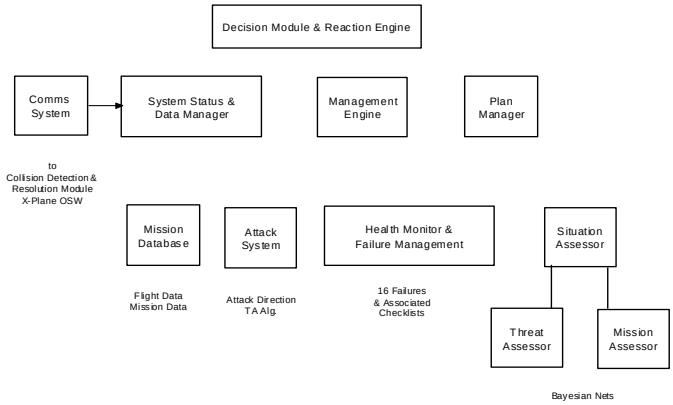


Figure 63: Concept Model Reference Architecture

This model included simple planning functionality to enable the aircraft to get airborne, fly a route, deal with a few emergencies (not fully tested or demonstrated) and land safely on the runway. It was noticed that considerable programming was required even for these simple plans and the conclusion was formed that programming plans was neither a simple nor scalable exercise. This would be made worse if plans are required to be generated by the autonomous system in response to unusual situations and events[135]. The preferred choice of programming language for a certifiable system is ADA and this language is

[135] Which is not envisaged at this stage

characterised by being robust, reliable and heavyweight. These characteristics make programming of plans likely to be even more difficult – a fact that was established during software development of the proposed solution.

Noting that JACK had been developed with the specific intention of producing scalable plans but could not be certified for control use, some thought was given as to how JACK could be used. The result was the development of the Mixed Language Architecture Concept[136] as described below.

The prime requirement for certification is that the generation of the control signal from data must come solely from certified software. However, it was considered that the planning function, **in itself,** is not necessarily required to be certified. In other words, the generation of a proposed plan cannot ever result in a control signal. What can, and does do though, is the assessment, selection and implementation of a plan. Therefore, it may be possible to gain certification by implementing a separation, and preferably a physical separation, between the planning functions and the executive/control functions.

The concept is therefore to use JACK, operating on its own hardware, to generate alternative plans and pass these to the Executive for assessment, selection and implementation. The Executive and Controllers would operate in their own environment, be programmed in ADA (i.e. applications for which ADA is most suited) and would be necessarily certified.

In order to test this concept, Agent Oriented Software (AOS), the JACK developers, were contacted and they agreed to help. They had already demonstrated JACK operating under VxWorks using Perc software sourced from Aeonics. Meanwhile BAE SYSTEMS had already demonstrated their own

[136] Whilst I can and do claim ownership of the concept and the implementation design, the system integration effort and final demonstration was done by others.

routing software running VxWorks, and the proprietary 3 layer stack operating system, on a VME card. The JACK environment, application and Perc, were installed on a RIO4 VME card. Using an existing avionic application running on a linux based network and connecting the two via Ethernet, successful two-way communications were established between the JACK application programme and the BAE SYSTEMS application. This was internally demonstrated to senior BAE SYSTEMS personnel on 13 December 2006. The implementation of the demonstration is shown below:

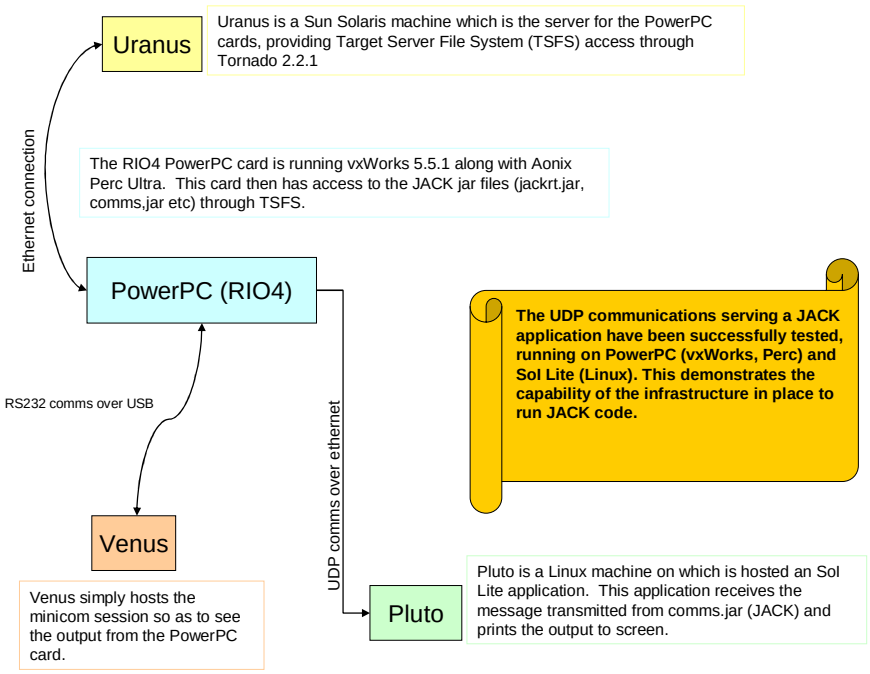


Figure 64: Mixed Language Architecture Demonstration Implementation

This demonstration is important because it not only demonstrates the feasibility of the Mixed Language Architecture in terms of communications and physical separation, but also because it utilised hardware and software typical of that used in state of the art commercial avionic systems flying today.

It is intended that this concept will be further extended during 2007 by a collaboration between BAE SYSTEMS and AOS under the ASTRAEA Programme. Particular emphasis will be on certifiability and BAE SYSTEMS certification engineers (not me) will study the functional system, develop use cases for safety functions and attempt to provide a process for certification.

Appendix F Description of Example UASs in Current Service [81]

F.1 MQ-1 Predator:

The Predator is an armed, multi-role, long endurance UAS (Group 4) that carries an EO/IR payload, laser target marker, laser illuminator and signal intelligence (SIGINT) payloads. Rated USAF pilots fly these aircraft by one of three methods. These methods are: manual flying, semi-autonomous monitored flight and pre-programmed flight. With two data link options, Predators can be flown LOS within approximately 100 miles of the launch and recovery base or flown BLOS via satellite datalinks. Missions can be controlled from the launch base or through Remote Split Operations (RSO) from worldwide-based mission control elements. The crew and aircraft can re-role to any component of the kill chain during one mission while performing the following missions and tasks: Intelligence, Surveillance, Reconnaissance (ISR), Close Air Support (CAS), Combat Search and Rescue (CSAR) support, precision strike, buddy laze, convoy over-watch, raid over-watch, target development, and terminal air control. Predators are used primarily for persistent ISR functions. The Predator force objective is 185 aircraft, funded through the Military Intelligence Program (MIP).

The Predator has the following performance:

- Max Altitude: 25,000 ft. ; Employment altitude: 10,000-20,000 ft.
- Max speed: 120 KIAS; Loiter speed: 80 KIAS
- Operational Endurance: 22 hrs.
- Max payload: 300 lbs. externally

F.2 MQ-9 Reaper:

The Reaper is an armed, multi-role, long endurance UAS that carries an EO/IR payload, laser target marker, laser illuminator and synthetic aperture radar (SAR). Seven external hard points allow an open architecture variety of weapon and SIGINT payloads to be carried. Rated USAF pilots fly these aircraft by one of three methods. These methods are: manual flying, semi-autonomous monitored flight and pre-programmed flight. With two data link options, Reapers can be flown LOS within approximately 100 miles of the launch and recovery base or flown BLOS via satellite datalinks. Missions can be controlled from the launch base or through remote split operations (RSO) from worldwide-based mission control elements. The crew and aircraft can re-role to any component of the kill chain during one mission while performing the following missions and tasks: ISR, CAS, CSAR support, precision strike, buddy laze, convoy over-watch, raid over-watch, target development, and terminal air control. Reapers are used primarily for persistent strike functions while possessing loiter time for ISR functions as well. The Reaper FY10 force objective is 319 aircraft. This will enable a transition plan for growth to 50 Reaper and Predator combined combat air patrols (CAP) by 4QFY11 and all Reaper by FY16.

The Reaper has the following performance:

- Max Altitude: 50,000 ft.; Employment altitude: 25,000-30,000 ft.
- Max speed: 240 KIAS ; Loiter speed: 100 KIAS
- Operational endurance: 18 hrs.
- Max payload: 3000 lbs. externally

F.3 RQ-4 Global Hawk:

The Global Hawk can be operated LOS or BLOS and transmit its data to the USAF Distributed Common Ground System (DCGS) or other nodes including the Army Tactical Exploitation system (TES) for exploitation and dissemination. The Global Hawk force structure contains two baseline models, RQ-4A and RQ-4B, in 4 production blocks, funded by the Military Intelligence Program (MIP). Seven RQ-4A Block 10 aircraft are equipped with EO, IR, and SAR sensors. Six RQ-4B Block 20 aircraft will be equipped with the Battlefield Airborne Communications Node (BACN). BACN provides a Tactical Data Link gateway between Link 16, the Situation Airborne Data Link (SADL) and the Integrated Broadcast System (IBS). Through BACN, users of these three systems can share information and form a common tactical picture. Further, BACN provides an Internet Protocol based networking capability so military networks can interface and share content across both secure and open internet connections. BACN provides the capability to "cross-band" military, civilian and commercial communications systems. Further, BACN allows soldiers on foot, or platforms without advanced communications systems to connect via cellular phones, existing narrow band radios, or even an airborne 802.11 to the battle field network. Forty-two RQ-4B Block 30 aircraft will have the Enhanced Integrated Sensor Suite (EISS) with EO, IR, and SAR and the Airborne Signals Intelligence Payload (ASIP) for SIGINT collection. Twenty-two RQ-4B Block 40 aircraft will have the Multi-Platform Radar Technology Insertion Program (MP-RTIP) payload; planned capability includes Active Electronically Scanned Array (AESA) radar with concurrent high-resolution SAR imagery, high-range-resolution (HRR) imagery, and robust Ground Moving Target Indicator (GMTI) data. The ground stations (10 for the multi-INT systems; 3 for the Block 40) consist of a Launch and Recovery Element (LRE) and the Mission Control Element (MCE). The crew is two pilots (1 for MCE, 1 for LRE), one sensor

operator, and additional support that include one Quality Control (QC) manager, and one communications technician.

The Global Hawk has the following performance:

- Max Altitude: 65,000 ft. (Block 10), 60,000 ft. (Blocks 20/30/40)
- Max speed: 340 KTAS (Block 10), 320 KTAS (Blocks 20/30/40)
- Max endurance: 28 hrs.
- Max payload: 2,000 lbs. (Block 10), 3,000 lbs. (Blocks 20/30/40)

Appendix G Sense and Avoid Experimental Data

G.1 Sense and Avoid Experimental Setup

The Sense and Avoid sub system of the AIMS architecture was implemented as in Figure 15. The experimental setup was as described below:

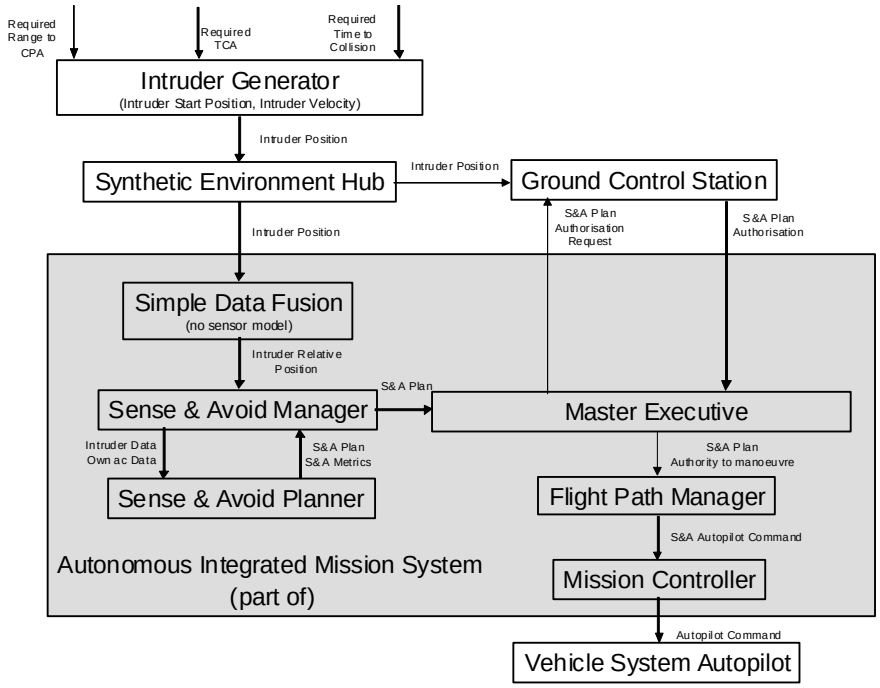


Figure 65: Schematic of Sense & Avoid Experimental Setup

The Intruder Generator generates the intruder's start position and velocity based the on input parameters above. An updated position for all hub objects is calculated at each cycle of the Synthetic Environment Hub. Note that an application named Simple Data Fusion, which tracks the intruder position, is used instead of a millimetric wave radar model (which was unavailable at the time of the experiment).

G.2 Sense and Avoid Performance Model

A copy of the Matlab code for the Sense and Avoid Performance Model for the 10k Detection Range parameter is given below :

```
% File: Performance Margin Test file.m
% A timeline model for sense and avoid variables
% A positive mean distribution denotes a human can be in the loop
% a negative one denotes that the autonomous system will perform better
% Let us simulate ...

PM = 0;
nSim = 40000;
result = zeros(nSim, 1);
sortedres = zeros(nSim, 1);
meandet = 10000;
plus = 0;
neg  = 0;

for k = 1:nSim,
DetectRange = meandet + 1363 * randn();
% Track Crossing Angle +- 30 degs
TCA = 0.5 *(rand());
% closing velocity 100m/s - 200m/s
CloseVelocity = (261.4 + 3.3 * randn()); %* cos(TCA);
% initial time to collision
TimeToCollision = DetectRange / CloseVelocity;
% downlink latency, poisson with a mean of 1.2
downlink = poissrnd(1.2);
% uplink latency, poisson with a mean of 1.2
uplink = downlink;
% pilot decision process, gaussian with a mean of 7, SD of 2
PilotDecisionProcess = 7.0 + 2 * randn();
% safety buffer
SafetyBuffer = 7;% + 1 * rand();
% safe separation generation
SafeSepGen = 11;% + 3 * rand();

PM = fix((TimeToCollision -  downlink - ...
        PilotDecisionProcess - uplink - ...
        SafetyBuffer - SafeSepGen));
```

```
result(k) = PM;

%fprintf('  PerformanceMargin: %.1f\n', PM);

sortedres = sort(result);
sortmean = chop(mean(sortedres), 3);
sortdev = chop(std(sortedres), 3);
skew = chop(skewness(sortedres), 3);

hist(sortedres, 16);

xlabel('Performance Margin(secs)');
ylabel('Number of Occurrences');
text(10, 8000, ['Mean Det Range(km) = ' num2str(meandet/1000)]);
text(10, 7500, ['Skewness = ' num2str(sortdev)]);
text(10, 7000, ['Mean Closing Velocity = 261m/sec']);
text(10, 6500, ['Distribution Mean = ' num2str(sortmean)]);
text(10, 6000, ['Number of Samples = ' num2str(nSim)]);
text(12, 5500, ['Distribution SD = ' num2str(sortdev)]);
text(12, 5000, ['Pilot Decision = 5 sec']);
text(14, 4500, ['Safety Buffer = 7 sec']);
text(14, 4000, ['Safe Sep = 11 sec']);
text(14, 3500, ['TCA = 180 degs']);
```

G.3 Sense and Avoid Experimental Raw Data as Logged

Ref	Detection Range(km)	UAV speed m/s	UAV speed Knots	Intruder Speed	Knots	Closing Speed (m/s)	TCA (deg)	Downlink Latency (sec)	Uplink Latency (sec)	Pilot Decision Time (sec)	Separation Required m	Separation Required nm	Time at 'Intruder Detected'	TIME	Time S&A Message Sent	TIME	Time Auth. Received	TIME	Time Manoeuvre Initiated	Time Auto Manoeuvre	PACT Level for Manoeuvre	Performance Margin (sec)	Separation Achieved(m)
1	9.6	115.8	225	136.9	266	252.8	180	3	3	5	926	0.5	5.32135E+13	0.302	5.32138E+13	13.1	5.32269E+13	-2.2	5.3225E+13	5.32249E+13	5A	-2.0	939
2	9.8	115.8	225	144.7	281	260.5	180	3	3	5	926	0.5	5.35186E+13	0.299	5.35189E+13	12.502	5.35314E+13	-1.5	5.353E+13	5.35302E+13	5A	-1.2	936
3	10	115.8	225	137.4	267	253.3	180	3	3	5	926	0.5	5.36908E+13	0.3	5.36911E+13	12.905	5.3704E+13	0	5.3704E+13	5.37043E+13	4A	0.3	940
4	9.9	115.8	225	137.4	267	253.3	160	3	3	5	926	0.5	5.38645E+13	0.301	5.38648E+13	12.402	5.38772E+13	0	5.3877E+13	5.38778E+13	4A	0.6	915
5	8.5	115.8	225	128.7	250	244.5	180	3	3	5	926	0.5	5.4906E+13	0.301	5.49063E+13	11.9	5.49182E+13	-4	5.4914E+13	5.49144E+13	5A	-3.8	859
6	8.2	115.8	225	132.3	257	248.1	-160	3	3	5	926	0.5	5.50726E+13	0.3	5.50729E+13	12.6	5.50855E+13	-5.7	5.508E+13	5.508E+13	5A	-5.5	835
7	8	115.8	225	130.2	253	246.1	180	3	3	5	926	0.5	5.53838E+13	0.299	5.53841E+13	NA	0	NA	5.539E+13	5.53901E+13	5A	Auth	920
8	10.1	115.8	225	137.4	267	253.3	180	3	3	5	926	0.5	5.57686E+13	0.299	5.57689E+13	12.401	5.57813E+13	0	5.5781E+13	5.5782E+13	4A	0.6	910
9	11.2	115.8	225	120.5	234	236.3	180	3	3	5	926	0.5	5.61248E+13	0.302	5.61251E+13	12.3	5.61374E+13	0	5.6137E+13	5.61476E+13	4A	10.3	923
10	10.5	115.8	225	150.3	292	266.1	180	3	3	5	926	0.5	5.62382E+13	0.302	5.62385E+13	12.301	5.62508E+13	0	5.6251E+13	5.62511E+13	4A	0.3	
11	10.3	115.8	225	158.0	307	273.9	180	2	2	5	926	0.5	2.70861E+13	0.299	2.70864E+13	9.703	2.70961E+13	0	2.7096E+13	2.70972E+13	4A	1.1	914
12	11.1	115.8	225	133.8	260	249.7	180	2	2	5	926	0.5	2.72556E+13	0.3	2.72559E+13	10.401	2.72663E+13	0	2.7266E+13	2.7274E+13	4A	7.7	920
13	10.3	115.8	225	131.3	255	247.1	180	2	2	5	926	0.5	2.73769E+13	0.303	2.73772E+13	10.097	2.73873E+13	0	2.7387E+13	2.73926E+13	4A	5.3	918
14	9.4	115.8	225	128.7	250	244.5	180	2	2	5	926	0.5	2.75186E+13	0.298	2.75189E+13	10.202	2.75291E+13	0	2.7529E+13	2.75311E+13	4A	2.1	917
15	8.6	115.8	225	125.1	243	240.9	180	2	2	5	926	0.5	2.7682E+13	0.299	2.76823E+13	9.999	2.76923E+13	-1	2.7691E+13	2.76915E+13	5A	-0.8	941
16	7.6	115.8	225	125.1	243	240.9	180	2	2	5	926	0.5	2.78099E+13	0.301	2.78102E+13	NA	0	NA	2.7815E+13	2.78154E+13	5A	Auth	938
17	8.3	115.8	225	122.5	238	238.3	180	2	2	5	926	0.5	2.79636E+13	0.299	2.79639E+13	10.601	2.79745E+13	-2.6	2.7972E+13	2.79722E+13	5A	-2.4	938
18	9.1	115.8	225	132.3	257	248.1	180	2	2	5	926	0.5	2.82076E+13	0.299	2.82079E+13	10.401	2.82183E+13	-0.5	2.8218E+13	2.8218E+13	5A	-0.3	914
19	8.1	115.8	225	119.9	233	235.8	180	2	2	5	926	0.5	2.83258E+13	0.302	2.83261E+13	10.1	2.83362E+13	-2.1	2.8334E+13	2.83345E+13	5A	-1.6	914
20	7.8	115.8	225	128.2	249	244.0	180	2	2	5	926	0.5	2.84518E+13	0.302	2.84521E+13	NA	0	NA	2.8457E+13	2.84573E+13	5A	Auth	939
21	10	115.8	225	138.0	268	253.8	180	1	1	5	926	0.5	2.86391E+13	0.299	2.86394E+13	9	2.86484E+13	0	2.8648E+13	2.86524E+13	4A	4.0	934
22	9.6	115.8	225	138.0	268	253.8	180	1	1	5	926	0.5	2.89029E+13	0.297	2.89032E+13	8.303	2.89115E+13	0	2.8912E+13	2.89143E+13	4A	2.8	936
23	9	115.8	225	142.6	277	258.4	180	1	1	5	926	0.5	2.90043E+13	0.298	2.90046E+13	7.602	2.90122E+13	0	2.9012E+13	2.9013E+13	4A	0.8	940
24	8.1	115.8	225	140.5	273	256.4	180	1	1	5	926	0.5	2.90983E+13	0.298	2.90986E+13	NA	0	NA	2.9104E+13	2.9104E+13	5A	Auth	940
25	8.6	115.8	225	138.0	268	253.8	180	1	1	5	926	0.5	2.92026E+13	0.302	2.92029E+13	8.298	2.92112E+13	-1.4	2.921E+13	2.92102E+13	5A	-1.0	940
26	7.8	115.8	225	131.3	255	247.1	180	1	1	5	926	0.5	2.94699E+13	0.303	2.94702E+13	NA	0	NA	2.9475E+13	2.94753E+13	5A	Auth	918
27	10.1	115.8	225	137.4	267	253.3	180	1	1	5	926	0.5	2.9593E+13	0.298	2.95933E+13	8.4	2.96017E+13	0	2.9602E+13	2.96066E+13	4A	4.9	918
28	11	115.8	225	136.4	265	252.2	180	1	1	5	926	0.5	2.97473E+13	0.301	2.97476E+13	8.299	2.97559E+13	0	2.9756E+13	2.97644E+13	4A	8.6	917
29	11.5	115.8	225	144.1	280	260.0	180	3	3	5	463	0.25	3.05562E+13	0.298	3.05565E+13	11.801	3.05683E+13	0	3.0568E+13	3.05816E+13	4A	13.3	465
30	10.5	115.8	225	149.8	291	265.6	180	3	3	5	463	0.25	3.07089E+13	0.302	3.07092E+13	12.098	3.07213E+13	0	3.0721E+13	3.07297E+13	4A	8.3	469
31	9.7	115.8	225	148.8	289	264.6	180	3	3	5	463	0.25	3.08279E+13	0.302	3.08283E+13	12.299	3.08405E+13	0	3.0841E+13	3.08451E+13	4A	4.6	468
32	8.9	115.8	225	146.7	285	262.5	180	3	3	5	463	0.25	3.09434E+13	0.302	3.09437E+13	12.6	3.09563E+13	0	3.0956E+13	3.09576E+13	4A	1.3	464
33	8.3	115.8	225	146.7	285	262.5	180	3	3	5	463	0.25	3.10899E+13	0.3	3.10902E+13	12.202	3.11024E+13	-0.9	3.1101E+13	3.11018E+13	5A	-0.6	461
34	8.8	115.8	225	144.1	280	260.0	180	3	3	5	463	0.25	3.1187E+13	0.303	3.11873E+13	12.699	3.12E+13	0	3.12E+13	3.12012E+13	4A	1.2	462
35	8.1	115.8	225	140.5	273	256.4	180	3	3	5	463	0.25	3.12998E+13	0.302	3.13001E+13	12.901	3.1313E+13	-1.6	3.1311E+13	3.13119E+13	5A	-1.1	459
36	7.6	115.8	225	148.3	288	264.1	180	3	3	5	463	0.25	3.14542E+13	0.302	3.14545E+13	12.299	3.14668E+13	-3.9	3.1463E+13	3.1464E+13	5A	-2.8	462
37	7.4	115.8	225	141.0	274	256.9	180	3	3	5	463	0.25	3.15719E+13	0.298	3.15722E+13	12.601	3.15848E+13	-4.5	3.158E+13	3.15806E+13	5A	-4.3	454
38	7	115.8	225	144.1	280	260.0	180	3	3	5	463	0.25	3.16959E+13	0.299	3.16962E+13	NA	0	NA	3.1702E+13	3.17025E+13	5A	Auth	453
39		0		0.0		0.0	180				463	0.25		0		NA		NA			4A	Auth not se	
40		0		0.0		0.0	180				463	0.25		0		NA		NA			4A	Auth not se	
41		0		0.0		0.0	180				463	0.25		0		NA		NA			4A	Auth not se	
42		0		0.0		0.0	180				463	0.25		0		NA		NA					

G.4 Sense and Avoid Experimental Raw Data Subset for a Required Closest Point of Approach of 926m

Ref	Detection Range(km)	UAV speed m/s	UAV speed Knots	Intruder Speed knots	Closing Speed (m/s)	TCA (deg)	Downlink Latency (sec)	Uplink Latency (sec)	Pilot Decision Time (sec)	Separation Required m	Separation Required nm	Time at 'Intruder Detected'	TIME	Time S&A Message Sent	TIME	Time Auth. Received	TIME	Time Manoeuvre Initiated	Time Auto Manoeuvre		PACT Level for Manoeuvre	Performance Margin (sec)	Separation Achieved(m)
5	8.5	115.8	225	128.7	250	244.5	180	3	3	5	926	0.5	5.4906E+13	0.301	5.49063E+13	11.9	5.49182E+13	-4.002	5.49142E+13	5.49144E+13		5A	-3.8	859
15	8.3	115.8	225	122.5	238	238.3	180	2	2	5	926	0.5	2.79636E+13	0.299	2.79639E+13	10.601	2.79745E+13	-2.6	2.79719E+13	2.79722E+13		5A	-2.4	938
1	9.6	115.8	225	136.9	266	252.8	180	3	3	5	926	0.5	5.32135E+13	0.302	5.32138E+13	13.1	5.32269E+13	-2.199	5.32247E+13	5.32249E+13		5A	-2.0	939
17	8.1	115.8	225	119.9	233	235.8	180	2	2	5	926	0.5	2.83258E+13	0.302	2.83261E+13	10.1	2.83362E+13	-2.098	2.83341E+13	2.83345E+13		5A	-1.6	914
2	9.8	115.8	225	144.7	281	260.5	180	3	3	5	926	0.5	5.35186E+13	0.299	5.35189E+13	12.502	5.35314E+13	-1.501	5.35299E+13	5.35302E+13		5A	-1.2	936
21	8.6	115.8	225	138	268	253.8	180	1	1	5	926	0.5	2.92026E+13	0.302	2.92029E+13	8.298	2.92112E+13	-1.399	2.92098E+13	2.92102E+13		5A	-1.0	940
14	8.6	115.8	225	125.1	243	240.9	180	2	2	5	926	0.5	2.7682E+13	0.299	2.76823E+13	9.999	2.76923E+13	-1	2.76913E+13	2.76915E+13		5A	-0.8	941
16	9.1	115.8	225	132.3	257	248.1	180	2	2	5	926	0.5	2.82076E+13	0.299	2.82079E+13	10.401	2.82183E+13	-0.499	2.82178E+13	2.8218E+13		5A	-0.3	914
3	10	115.8	225	137.4	267	253.3	180	3	3	5	926	0.5	5.36908E+13	0.3	5.36911E+13	12.905	5.3704E+13	0	5.3704E+13	5.37043E+13	0.29	4A	0.3	940
9	10.5	115.8	225	150.3	292	266.1	180	3	3	5	926	0.5	5.62382E+13	0.302	5.62385E+13	12.301	5.62508E+13	0	5.62508E+13	5.62511E+13		4A	0.3	
7	10.1	115.8	225	137.4	267	253.3	180	3	3	5	926	0.5	5.57686E+13	0.299	5.57689E+13	12.401	5.57813E+13	0	5.57813E+13	5.5782E+13		4A	0.6	910
20	9	115.8	225	142.6	277	258.4	180	1	1	5	926	0.5	2.90043E+13	0.298	2.90046E+13	7.602	2.90122E+13	0	2.90122E+13	2.9013E+13		4A	0.8	940
10	10.3	115.8	225	158	307	273.9	180	2	2	5	926	0.5	2.70861E+13	0.299	2.70864E+13	9.703	2.70961E+13	0	2.70961E+13	2.70972E+13		4A	1.1	914
13	9.4	115.8	225	128.7	250	244.5	180	2	2	5	926	0.5	2.75186E+13	0.298	2.75189E+13	10.202	2.75291E+13	0	2.75291E+13	2.75311E+13		4A	2.1	917
19	9.6	115.8	225	138	268	253.8	180	1	1	5	926	0.5	2.89029E+13	0.297	2.89032E+13	8.303	2.89115E+13	0	2.89115E+13	2.89143E+13		4A	2.8	936
18	10	115.8	225	138	268	253.8	180	1	1	5	926	0.5	2.86391E+13	0.299	2.86394E+13	9	2.86484E+13	0	2.86484E+13	2.86524E+13		4A	4.0	934
22	10.1	115.8	225	137.4	267	253.3	180	1	1	5	926	0.5	2.9593E+13	0.298	2.95933E+13	8.4	2.96017E+13	0	2.96017E+13	2.96066E+13		4A	4.9	918
12	10.3	115.8	225	131.3	255	247.1	180	2	2	5	926	0.5	2.73769E+13	0.303	2.73772E+13	10.097	2.73873E+13	0	2.73873E+13	2.73926E+13		4A	5.3	918
11	11.1	115.8	225	133.8	260	249.7	180	2	2	5	926	0.5	2.72556E+13	0.3	2.72559E+13	10.401	2.72663E+13	0	2.72663E+13	2.7274E+13		4A	7.7	920
23	11	115.8	225	136.4	265	252.2	180	1	1	5	926	0.5	2.97473E+13	0.301	2.97476E+13	8.299	2.97559E+13	0	2.97559E+13	2.97644E+13		4A	8.6	917
8	11.2	115.8	225	120.5	234	236.3	180	3	3	5	926	0.5	5.61248E+13	0.302	5.61251E+13	12.3	5.61374E+13	0	5.61374E+13	5.61476E+13		4A	10.3	923
	Mean =				Mean =		Mean	Mean =											0.8				Mean =	Mean =
	9.676				251		2	2															2.0	926.8
	Std Dev =				Std Dev =		Std D	Std Dev =															Std Dev	Std Dev =
	0.925				9.38		0.8	0.8															3.7	11.55

G.5 Sense and Avoid Experimental Raw Data Subset for a Required Closest Point of Approach of 463m

Ref	Detection Range(km)	UAV speed m/s	UAV speed Knots	Intruder Speed	knots	Closing Speed (m/s)	TCA (deg)	Downlink Latency (sec)	Uplink Latency (sec)	Pilot Decision Time (sec)	Separation Required m	Separation Required nm	Time at 'Intruder Detected'	TIME	Time S&A Message Sent	TIME	Time Auth. Received	TIME	Time Manoeuvre Initiated	Time Auto Manoeuvre	PACT Level for Manoeuvre	Performance Margin (sec)	Separation Achieved(m)
37	7.4	115.8	225	141.0	274	256.9	180	3	3	5	463	0.25	3.1572E+13	0.3	3.1572E+13	12.6	3.1585E+13	-4.5	3.158E+13	3.1581E+13	5A	-4.3	454
36	7.6	115.8	225	148.3	288	264.1	180	3	3	5	463	0.25	3.1454E+13	0.3	3.1455E+13	12.3	3.1467E+13	-3.9	3.1463E+13	3.1464E+13	5A	-2.8	462
35	8.1	115.8	225	140.5	273	256.4	180	3	3	5	463	0.25	3.13E+13	0.3	3.13E+13	12.9	3.1313E+13	-1.6	3.1311E+13	3.1312E+13	5A	-1.1	459
33	8.3	115.8	225	146.7	285	262.5	180	3	3	5	463	0.25	3.109E+13	0.3	3.109E+13	12.2	3.1102E+13	-0.9	3.1101E+13	3.1102E+13	5A	-0.6	461
34	8.8	115.8	225	144.1	280	260.0	180	3	3	5	463	0.25	3.1187E+13	0.3	3.1187E+13	12.7	3.12E+13	0	3.12E+13	3.1201E+13	4A	1.2	462
32	8.9	115.8	225	146.7	285	262.5	180	3	3	5	463	0.25	3.0943E+13	0.3	3.0944E+13	12.6	3.0956E+13	0	3.0956E+13	3.0958E+13	4A	1.3	464
31	9.7	115.8	225	148.8	289	264.6	180	3	3	5	463	0.25	3.0828E+13	0.3	3.0828E+13	12.3	3.0841E+13	0	3.0841E+13	3.0845E+13	4A	4.6	468
30	10.5	115.8	225	149.8	291	265.6	180	3	3	5	463	0.25	3.0709E+13	0.3	3.0709E+13	12.1	3.0721E+13	0	3.0721E+13	3.073E+13	4A	8.3	469
29	11.5	115.8	225	144.1	280	260.0	180	3	3	5	463	0.25	3.0556E+13	0.3	3.0557E+13	11.8	3.0568E+13	0	3.0568E+13	3.0582E+13	4A	13.3	465
	Mean =					Mean =		Mean	Mean =														
	8.978					261.4		3	3													Mean =	Mean =
	Std Dev =					Std Dev =		Std D	Std Dev =													3.0	463.75
	1.363					3.3		0	0													Std Dev =	Std Dev =
																						5.4	3.4538

G.6 Data for the CPA 926m Plot

								Bank	Turn start																
Velocity Au	0	0	0	0	0	0	0	0	0	6.568163	13.13633	19.70449	26.27265	32.84081	39.40898	45.97714	52.5453	59.11346	65.68163	72.24979	78.81795	85.38611	91.95428	98.52244	105.0906
Velocity Av	125.5	125.5	125.5	125.5	125.5	125.5	125.5	125.5	125.5	124.5014	123.5027	122.5041	121.5055	120.5069	119.5082	118.5096	117.511	116.5123	115.5137	114.5151	113.5164	112.5178	111.5192	110.5206	109.5219
Velocity Bu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Velocity Bv	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5
Timescale	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Time	24.324	23.324	22.324	21.324	20.324	19.324	18.324	17.324	16.324	15.324	14.324	13.324	12.324	11.324	10.324	9.324	8.324	7.324	6.324	5.324	4.324	3.324	2.324	1.324	0.324
Ax	0	0	0	0	0	0	0	0	0	6.568163	19.70449	39.40898	65.68163	98.52244	137.9314	183.9086	236.4539	295.5673	361.2489	433.4987	512.3167	597.7028	689.6571	788.1795	893.2701
Ay	-3060	-2934.5	-2809	-2683.5	-2558	-2432.5	-2307	-2181.5	-2056	-1931.5	-1808	-1685.49	-1563.99	-1443.48	-1323.97	-1205.46	-1087.95	-971.438	-855.925	-741.41	-627.893	-515.375	-403.856	-293.336	-183.814
Bx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
By	3060	2934.5	2809	2683.5	2558	2432.5	2307	2181.5	2056	1930.5	1805	1679.5	1554	1428.5	1303	1177.5	1052	926.5	801	675.5	550	424.5	299	173.5	48
SepX	0	0	0	0	0	0	0	0	0	6.568163	19.70449	39.40898	65.68163	98.52244	137.9314	183.9086	236.4539	295.5673	361.2489	433.4987	512.3167	597.7028	689.6571	788.1795	893.2701
SepY	-6120	-5869	-5618	-5367	-5116	-4865	-4614	-4363	-4112	-3862	-3613	-3364.99	-3117.99	-2871.98	-2626.97	-2382.96	-2139.95	-1897.94	-1656.92	-1416.91	-1177.89	-939.875	-702.856	-466.836	-231.814
SepR	6120	5869	5618	5367	5116	4865	4614	4363	4112	3862	3613	3365	3119	2874	2631	2390	2153	1921	1696	1482	1284	1114	985	916	923

G.7 Data for the CPA 423m Plot

				Bank	Turn start											
Velocity Au	0	0	0	0	0	6.840309	13.68062	20.52093	27.36124	34.20155	41.04186	47.88217	54.72248	61.56279	68.40309	75.2434
Velocity Av	130.7	130.7	130.7	130.7	130.7	129.7014	128.7027	127.7041	126.7055	125.7069	124.7082	123.7096	122.711	121.7123	120.7137	119.7151
Velocity Bu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Velocity Bv	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5	-125.5
Timescale	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Time	16.5	15.5	14.5	13.5	12.5	11.5	10.5	9.5	8.5	7.5	6.5	5.5	4.5	3.5	2.5	1.5
Ax	0	0	0	0	0	6.840309	20.52093	41.04186	68.40309	102.6046	143.6465	191.5287	246.2511	307.8139	376.217	451.4604
Ay	-1886.3	-1755.6	-1624.9	-1494.2	-1363.5	-1233.8	-1105.1	-977.392	-850.686	-724.979	-600.271	-476.562	-353.851	-232.138	-111.425	8.290451
Bx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
By	1886.3	1760.8	1635.3	1509.8	1384.3	1258.8	1133.3	1007.8	882.3	756.8	631.3	505.8	380.3	254.8	129.3	3.8
SepX	0	0	0	0	0	6.840309	20.52093	41.04186	68.40309	102.6046	143.6465	191.5287	246.2511	307.8139	376.217	451.4604
SepY	-3772.6	-3516.4	-3260.2	-3004	-2747.8	-2492.6	-2238.4	-1985.19	-1732.99	-1481.78	-1231.57	-982.362	-734.151	-486.938	-240.725	4.490451
SepR	3773	3516	3260	3004	2748	2493	2238	1986	1734	1485	1240	1001	774	576	447	451

Appendix H UAS Accident Data

							HFACS Model Categories			
Index	Date	UAV Type	Sub-type	Primary Cause	Notes	Error Type	Primary	Model Category 2	Model Category 3	Model Category4
1	04/10/2000	Predator	RQ-1L	Failure of the flight computer	watchdog timer disabled - Configuration control process failed	Supervisory Error	Unsafe Supervision	Supervisory Violation	Expediant disablement of a safety function	
2	23/10/2000	Predator	RQ-1K	Failure of the variable pitch propeller control	primary failure caused by maintenance error	Maintenance error	Maintenance error			
3	14/09/2000	Predator	RQ-1L	Activation of "erase memory" menu option	poor design of system and GCS - Failure to	Pilot Error	Unsafe Acts	Violation	Routine	Failed to follow procedure
4	30/03/2001	Predator	RQ-1L	Failure to follow checklists	Pitot static icing - Non use of pitot heating	Pilot Error	Unsafe Acts	Violation	Routine	Failed to follow procedure
5	22/01/2002	Predator	RQ-1L	Failure to follow checklists	Incorrect procedures during hand over of	Pilot Error	Unsafe Acts	Violation	Routine	Failed to follow procedure
6	25/01/2002	Predator	RQ-1B	Pilot loss of control during landing	Landing attempted on wind gust limits	Pilot Error	Unsafe Acts	Violation	Exceptional	Accepted an Uneccessary Hazard
7	17/05/2002	Predator	RQ-1L	Incorrect assembly of tailplane servo		Manufacturing error				
8	25/05/2002	Predator	RQ-1B	Engine failure	Failure of cylinder rocker arm	Mechanical failure	Mechanical failure			
9	25/10/2002	Predator	RQ-1L	CFIT	Failure to monitor altitude - Distraction - low SA	Pilot Error	Unsafe Acts	Error	Skill Based Error	Failed to Prioritise Attention
10	01/01/2003	Predator	RQ-1B	Engine failure caused by loss of oil	failure of seals	Maintenance error	Maintenance error			
11	11/12/2003	Predator	RQ-1	Pilot loss of control, abrupt pitch inputs	Incorrect diagnosis of icing, software anomaly, intermittant link connection	Pilot Error	Unsafe Acts	Error	Skill Based Error	Overcontrolled the Aircraft
12	14/06/2004	Predator	MQ-1L	Late executed go around	wind conditions, risk management contributed	Pilot Error	Unsafe Acts	Violation	Routine	Poor technique/airmanship
13	17/08/2004	Predator	MQ-1L	Engine fire	Incorrect routing of oil lines	Maintenance error	Maintenance error			
14	22/09/2004	Predator	MQ-1L	Failure of pilot to correct high flare	committed approach outisde pblished command criteria	Pilot Error	Unsafe Acts	Violation	Routine	committed approach outisde pblished command criteria
15	24/11/2004	Predator	MQ-1L	Crashed short of the runway	Pilot did not execute landing checklist	Pilot Error	Unsafe Acts	Violation	Routine	Failed to follow procedure
16	14/01/2005	Predator	MQ-1L	Lost communications, ran out of fuel	Inflexible lost link procedures - incorrect procedures following system crash	Design Error	Organisational Influences	Resource Management	Equipment/Facility/Resources	Failed to Correct a Known Design Problem
17	27/03/2005	Predator	MQ-1L	Engine fire	Fuel leak caused by failure of the fuel feed priming solenoid	Design Error	Organisational Influences	Resource Management	Equipment/Facility/Resources	Poor aircraft design
18	30/03/2005	Predator	MQ-1L	Propellor pilot bearing failure		Mechanical failure				
19	17/09/2005	Predator		Flew into hazardous weather	Flight control failure	Weather				
20	21/10/2005	Reaper	MQ-9A	Pilot failed to control aircraft glidepath	inadequate supervion, CRM issues	Pilot Error	Unsafe Acts	Error	Perceptual Error	Due to misjudged flightpath
21	20/03/2006	Predator	MQ-1	Pilot turned off Stability Augmentation	Operator error	Pilot Error	Unsafe Acts	Error	Decision Error	Inappropriate procedure
22	25/04/2006	Predator		Engine Fuel shut off signal inadvertantly sent	Operator error - checklist not followed	Pilot Error	Unsafe Acts	Violation	Routine	Failed to follow procedure
23	22/06/2006	Predator	MQ-1L	Engine failure due to loss of oil	Oil filter incorrectly fitted	Maintenance error	Maintenance error			
24	03/08/2006	Predator	MQ-1B	Inadvertant engine shut down in flight	Mistook engine kill for undercarriage raise	Pilot Error	Unsafe Acts	Error	Skill Based Error	Failed to Prioritise Attention
25	17/01/2007	Predator	MQ-1B	Engine failure	Carankshaft failed	Mechanical failure				
26	23/02/2007	Predator	MQ-1B	Variable pitch propeller servo		Mechanical failure				
27	26/03/2007	Predator	MQ-1B	Pilot Induced oscillation on landing	Pilot misjudged height initially	Pilot Error	Unsafe Acts	Error	Perceptual Error	Due to misjudged flightpath
28	30/07/2007	Predator	MQ-1B	Engine failure	Ignition box failure	Manufacturing error				
29	31/07/2007	Predator	MQ-1L	Engine Failure	Manifold Air Pressure Sensor failure	Design Error	Organisational Influences	Resource Management	Equipment/Facility/Resource	Poor aircraft design
30	29/11/2007	Predator	MQ-1L	Systems failure	Primary Control Module lost comms with tail computer board - preprgammed crash	Design Error	Organisational Influences	Resource Management	Equipment/Facility/Resources	Poor aircraft design
31	17/12/2007	Predator	MQ-1B	Electrical Failure	Alternator failure	Design Error	Organisational Influences	Resource Management	Equipment/Facility/Resource	Poor aircraft design
32	09/04/2008	Predator	MQ-1B	Engine failure	Engine throttle body assembly failure	Design Error	Organisational Influences	Resource Management	Equipment/Facility/Resource	Poor aircraft design
33	02/05/2008	Predator	MQ-1B	Engine failure	Ignition box failure	Design Error	Organisational Influences	Resource Management	Equipment/Facility/Resources	Poor aircraft design
34	02/06/2008	Predator	MQ-1B	Electrical system failure	Alternator failure	Design Error	Organisational Influences	Resource Management	Equipment/Facility/Resource	Poor aircraft design
35	12/06/2008	Predator	MQ-1B	Systems failure	faulty connection between the Primary Control	Design Error	Organisational Influences	Resource Management	Equipment/Facility/Resource	Poor aircraft design
36	21/07/2008	Predator	MQ-1B	Lost comms and flew into bad weather on route home	GCS power supplies failed temporarily	Weather				
37	19/10/2008	Predator	MQ-1B	Failure of Propeller qwill shaft bearing	Incorrect installation of bearing	Maintenance error	Maintenance error			
38	22/02/2009	Predator	MQ-1B	Systems failure	Primary Control Module failure	Design Error	Organisational Influences	Resource Management	Equipment/Facility/Resources	Poor aircraft design
39	20/03/2009	Reaper	MQ-9	Engine failure, failed to perform emergency landing	improperly assembled oil control valve, faulty design of height datum	Maintenance error	Maintenance error			
40	20/04/2009	Predator	MQ-1B	Catastrophic electrical failure	Short circuit most likely	Mechanical failure	No cause found			
41	28/04/2009	Predator	MQ-1B	Failure of Manifold Absolute Pressure vacuum line	Disconnected fitting - Line cut too short, poor technical orders	Maintenance error	Maintenance error			
42	08/05/2009	Predator	MQ-1B	Failure of right wing control module	Incorrect placement of control chip	Supervisory Error	Unsafe Supervision	Supervisory violation	Authorised and unnecessary hazard	
43	13/08/2009	Predator	MQ-1B	Failure of propeller qwill shaft		Manufacturing error				
44	04/09/2009	Predator	MQ-1B	Failure of Variable Pitch Propeller servo	No cause found	Mechanical failure	No cause found			
45	14/09/2009	Predator	MQ-1B	Failure of Left tail board servo	No cause found	Mechanical failure	No cause found	Single point failure		
46	03/09/2009	Predator	MQ-1B	Controlled Flight into Terrain	Channelised attention of flight crew	Pilot Error	Unsafe acts	Error	Skill Based Error	Channelised attention of flight crew
47	20/11/2009	Reaper	MQ-9	Crashed after lost link	Electrical failure	Mechanical failure				
48	20/04/2010	Predator	MQ-1B	Crashed on Touch and Go	Airspeed too low for conditions	Pilot Error	Unsafe acts	Error	Skill Based Error	Due to misjudged flightpath
49	28/07/2010	Predator	MQ-1B	Crashed during Taxi	Operator failed to follow checklist	Pilot Error	Unsafe Acts	Violation	Routine	Failed to follow procedure
50	16/08/2010	Predator	MQ-1B	Crashed during Take Off	Operator forgot to put Stabs on	Pilot Error	Unsafe Acts	Violation	Routine	Failed to follow procedure
51	31/08/2010	Reaper	MQ-9	Crashed during Hi AOA demo	Pilot stalled ac	Pilot Error	Unsafe acts	Error	Skill Based Error	Due to misjudged flightpath
52	19/09/2010	Predator	MQ-1B	Crashed in cruise	Majoroil leak caused engine failure	Mechanical failure				
53	09/12/2010	Predator	MQ-1B	Crashed in descent	Pilot stalled aircraft after switching off stabs	Pilot Error	Unsafe acts	Error	Skill Based Error	Failed to follow procedure

This data was collected from the web pages: United States Air Force Class A Aerospace Mishaps, at http://usaf.aib.law.af.mil/ last accessed on 5 July 2011

REFERENCES

[1] Civil Aviation Authority, Directorate of Airspace Policy: *"CAP 722 - Unmanned Aircraft System Operations in UK Airspace – Guidance"*, Section 2 Policy,

Chapter 1 UAS Operating Principles, Sub Chapter 5 Airspace Principles for UAS Operations in the UK, Sub Chapters 5.5 and 5.2 refer respectively, 6 April 2010

[2] Douglas A. Weigmann and Scott A. Shappell*, "A Human Error Approach to Aviation Accident Analysis: The Human Factors Analysis and Classification System",* Ashgate Publishing Ltd., Burlington, USA, 2003.

[3] James Reason, *"Human Error"*, Cambridge University Press, 1990.

[4] Butler A. Jumper: "*Air Force to bring "warfighting" focus to combat drone effort"*. Defense Daily Network 2005 Jan 25. Retrieved February 10, 2005, from the World Wide Web: http://www.defensedaily.com

[5] Remarks made at a US Air Force Association Conference Feb 2008: General Hal Hornberg, Chief of Air Combat Command USAF, 2008.

[6] James T. Luxhøj , Kimberlee Kauffeld: *"Evaluating the Effect of Technology Insertion into the National Airspace System "*, Department of Industrial and Systems Engineering, Rutgers, The State University of New Jersey 2003.

[7] Many aspects of this section is referenced from: Lawrence R. Newcombe: *"Unmanned Aviation: A Brief History of Unmanned Aerial Vehicles",* American Institute of Aeronautics and Astronautics, Reston Virginia , 2004

[8] One such citation is: Elizabeth Bone and Christopher Bolkcom: *"Unmanned Aerial Vehicles: Background and Issues for Congress",* Congressional Research Service, The (US) Library of Congress, April 2003

[*9*] Sqn Ldr Rajesh Kumar, RAAF: *"Tactical Reconnaissance: UASs versus Manned Aircraft, Chapter 6 - The Likely Cost of UAS and Manned Aircraft Operations"*, The Research Department Air Command and Staff College, USAF, March 1997.

[10] Lt Cdr. F. Karen Coyle*: "Unmanned Aerial Vehicles: Operational Implications for the Joint Force Commander"*. Defence Technical Information Center, Ft Belvior, VA. p14 (believed unpublished but cited in [8]).

[11] UAS Roadmap: *"Unmanned Aircraft Systems Roadmap: 2005-2030",* Office of the Secretary of Defense, August 4, 2005.

[12] Major James C. Hoffman, USAF and Charles Tustin Kamps: *"At the Crossroads, Future "Manning" for Unmanned Aerial Vehicles"*, Al Udeid Air Base, Qatar Maxwell AFB, Alabama, March 2005

[13] Department of Defence (USA): *"Unmanned Aerial Vehicle Reliability Study*", February 2003

[14] General Audit Office (USA): *"Unmanned Aerial Vehicles - Improved Strategic and Acquisition Planning Can Help Address Emerging Challenges"*, March 2005.

[15] Ella M. Atkins: *"Certifiable Autonomous Flight Management for Unmanned Aircraft Systems"***,** The BRIDGE Volume 40, Number 4, Winter 2010

[16] M.L. Cummings (MIT, Cambridge, Mass), S. Bruni, S. Mercier and P.J. Mitchell: *"Automation Architecture for Single Operator, Multiple UAS Command and Control"*, The International C2 Journal | Vol 1, No 2 | 1-24, 2007.

[17] Patchett C.H.: *"The Performance of an Intelligent Agent in Air Combat"*: Masters Thesis, Cranfield University, 2002.

[18] Warwick. G*. "Trust: The Greatest Obstacle to UAS Autonomy"*, Aviation Week, 13 September 2010.

[19] Bartolomasi Paulo G., Operational Analysis Department, BAE Warton: *"The Theory of Manoeuvre Warfare"*, BAe Report BAe-WOA-RP-GEN-11175, June 1995. Unpublished work.

[20] Sheridan, T.B. and Verplank W.L., 1978*. "Human and Computer Control of Undersea Tele-operators"*; Technical Report. MIT Man-machine Systems Laboratory, Cambridge, MA.

[21] Parasuraman R., Sheridan T.B, and Wickens C.D. (2000). *"A Model for Types and Levels of Human Interaction with Automation*". IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and Humans, Vol 30, No 3, pp. 286-297. May 2000.

[22] M. L. Cummings: *"Human Supervisory Control of Swarming Networks",* Massachusetts Institute of Technology, Cambridge, MA 02139

[23] R.M.Taylor*: "Cognitive Cockpit Systems Engineering: Pilot Authorisation and Control of Tasks";* Human Sciences, DSTL, Farnborough, UK.2001.

[24] A.Hill, P.Wllkinson and F.Cayzer (nee Sturrock): *"Adaptive Human-System Interaction: Assessment Facility Development Requirements",* BAES-ASE-W-W7N3-

RP-000027; BAE SYSTEMS, 2006. Unpublished report.

[25] IEEE STD 610.12 as adapted by the U.S. DOD C4ISR:"*Architecture Framework"*, Version 2.0, December 1997

[26] Bass, Clements, Kazman: "*Software Architecture in Practice",* 2nd edition*,* Addison-Wesley 2003

[27] Erann Gat: *"On Three Layer Architectures"*, Jet Propulsion Laboratory, California Institute of Technology, published in AI and Mobile Robots. AAAI Press, 1998.

[28] Ralph Hartley and Frank Pitone: *"Experiments with the Subsumption Architecture",* Proceedings of the International Conference on Robotics and Automation (ICRA), 1991.

[29] English philosopher and Franciscan monk William of Ockham (ca. 1285-1349). *"Pluralitas non est ponenda sine neccesitate"* or *"plurality should not be posited without necessity."* This can be taken to mean, in modern life, that in any given model, there should be no concepts, variables or constructs that are not really needed to explain the phenomenon. By doing that, developing the model will become much easier, and there is less chance of introducing inconsistencies, ambiguities and redundancies.

[30] R Edwards:, *Integrated Modular Systems- Architecture Concept Summary*; April 2001; BAE Systems. Unpublished report

[30] Tom Erkkinen: "*High-Integrity Code Generation and Verification"*, Aerospace and Defense Digest, February 2005

[32] UK Ministry of Defence: *"Defence Technology Strategy – for the demands of the 21st Century"*, pp112-113, 2006, available at http://www.mod.uk.

[33] D.R.Haddon, C.J.Whittaker: *"Aircraft Airworthiness Certification Standards for Civil UASs"*, Civil Aviation Authority, August 2002

[34] Informal interview with BAE SYSTEMS UAS pilots held during a "Day in the Life of a UAS Pilot " workshop held at BAE SYSTEMS Warton, April 2009.

[35] Mica R. Endsley: "*Situation Awareness and Human Error: Designing to Support Human Performance"*, SA Technologies, Inc. 1999

[36] Denis Besnard, David Greathead & Gordon Baxter: "*When Mental Models Go Wrong: Co-Occurrences In Dynamic, Critical Systems",* Universities of York and Newcastle, 2004.

[37] Kevin Burns: "*Mental Models and Normal Errors*", The MITRE Corporation, Bedford, MA, 2004.

[38] *Nall Report 2007*: Aircraft Operator and Pilot's Association Aircraft Safety Foundation, USA, 2008

[39] "*Statistical Summary of Commercial Jet Airplane Accidents Worldwide Operations 1959 – 2010"*: Aviation Safety, Boeing Commercial Airplanes, Seattle, Washington, USA. June 2011.

[40] A.P.Tvaryanas, W.T.Thompson, S.H.Constable : *"US Military Unmanned Aerial Vehicle Mishaps: Assessment of the Role of Human Factors Using the HFACS Classification System"*, USAF, March 2005

[41] S.D.Manning, C.E.Rash, P.A.LeDuc: *"The Role of Human Causal Factors in US Army UAV Accidents",* US Army, Fort Rucker, Alabama, 2004

[42] K.W. Williams:"*A Summary of Unmanned Aircraft Accident/Incident Data: Human Factors Implications*", Civil Aerospace Medical Institute, Federal Aviation Administration, 2004.

[43] Colonel David L.Nichols, "Mishap Analysis: An Improved Approach to Accident Prevention ", Air University Review, Air War College (USAF), July - August 1973

[44] Barry McGuinness, Ben Dawson*:"QUASA: The Measurement of Team and Shared Situational Awareness"*, Advanced Technology Centre, BAE Systems, Filton, 2004 (unpublished work).

[45] Manningham D: **"***The Cockpit: A Brief History*" McGraw-Hill, Inc. 1997

[46] Capt Etienne Tarnowski**,** Experimental Test Pilot, Airbus: "Cockpit Automation Philosophy", NATO RTO HFM Symposium on "The Role of Humans in Intelligent and Automated Systems", held in Warsaw, Poland, 7-9 October 2002, and published in RTO-MP-088.

[47] Bainbridge L: *"The Ironies of Automation"*, in J. Rasmussen, K. Duncan and J Leplat (Editors*) "New Technology and Human Error"*, London: Wiley, 1987

[48] Donald A. Norman, University of California, San Diego: *"The Problem Of Automation: Inappropriate Feedback And Interaction, Not Over-Automation"*, Philosophical Transactions of the Royal Society of London, 1990.

[49] Ken Carpenter: *"Draft Specification of the European Encounter Model – a précis"*. ACAS PROGRAMME, Work Package 1 - Studies on the safety of ACAS II in

Europe, 26 February, 2000.

[50] ACAS PROGRAMME, Work Package 1*:"Final Report on the Studies of the Safety of ACAS II in Europe"*, ACAS/ACASA 02-014, March 2002

[51] As discussed with members of the Communications Research Team at Military Air Solutions, BAE SYSTEMS on 21 November 2010.

[52] EUROCONTROL/FAA Future Communications Study Operational Concepts and Requirements Team: "*Communications Operating Concept and Requirements (COCR) for the Future Radio System (FRS)*", Version 2.0, May 2007

[53] 2nd Lt Thomas B. Billingsby: *"Safety Analysis of TCAS on Global Hawk using Airspace Encounter Models"*, USAF Academy, June 2006

[54] Aircraft Accident Report NTSB/AAR-10/03, PB2010-910403: *"Loss of Thrust in Both Engines After Encountering a Flock of Birds and Subsequent Ditching on the Hudson River, US Airways Flight 1549, Airbus A320-214, N106US, Weehawken, New Jersey January 15, 2009"*: National Transportation Safety Board, May 4 2010

[55] Aircraft Accident Synopsis CHI06MA121: "*Predator B Accident, Nogales, Arizona, 25 April 2006*": National Transportation Safety Board, October 31 2007.

[56 Werner J.A. Dahm, Chief Scientist of the U.S. Air Force (AF/ST): "*Report on Technology Horizons: A Vision for Air Force Science & Technology During 2010-2030*", USAF, Washington, May 2010

[57] Hyacinth S.Nwana: *"Software Agents: An Overview",* BT Laboratories, Knowledge Engineering Review, Vol. 11, No 3, pp.1-40, Sept 1996. Also available at http://www.sce.carleton.ca/netmanage/docs/AgentsOverview/ao.html

[58] Stan Franklin and Art Graesser: *"Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents"*, Institute for Intelligent Systems, University of Memphis, Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer – Verlag,1996.

[59] Ralph L.Keeney and Howard Raiffa, *"Decisions with Multiple Objectives: Preferences and Value Tradeoffs",* Cambridge University Press, first issued 1976 , re-issued 1993, ISBN 0-521-43883-7.

[60] John G. Morris, Christine M. Mitchell, William Potter: *"A Designer's Associate: Support for the Design of Software for Complex Dynamic Control Systems"*, Technical Report GIT-COGSCI-94/30, Cognitive Science Program, College of Computing,

Georgia Institute of Technology, Atlanta, GA, 1994 .

[61] Endsley M.R: *"Towards a Theory of Situation Awareness in Dynamic Systems",* Special Issue: "Situation Awareness", published in "Human Factors" 37(1) pp 32-64, 1995.

[62] G. A. Klein, *"Recognition-Primed Decisions"* in William B. Rouse (ed.), Advances in Man-Machine System Research (Greenwich, CT: Jai Press, 1989).

[63] John R.Boyd*: "A Discourse on Winning and Losing"*, 1986, a collection of briefings on competitive strategy. Unpublished reports but available at http://www.d-n-i.net/second_level/boyd/military.htm and at www.au.af.mil/au/awc/awcgate/awc-thry.htm#boyd

[64] James.S.Albus: *"A Reference Model Architecture for Intelligent Systems Design"*; Intelligent Systems Division, Manufacturing Engineering Laboratory, National Institute for Standards and Technology, 1999.

[65] James.S.Albus and Anthony.J.Barbera: RCS: *"A Cognitive Architecture for Intelligent Multi-Agent Systems"*; Intelligent Systems Division, Manufacturing Engineering Laboratory, National Institute for Standards and Technology, 2003.

[66] Dickmanns E.D: *"A General Dynamic Vision Architecture for UGV and UAS",* Journal of Applied Intelligence, 2, pp251-270, 1992

[67] Albus J*: "Brains Behaviour and Robotics"*, BYTE/McGraw Hill, Peterborough, NH, 1981

[68] Fodor J*: "The Mind Doesn't Work that Way";* MIT Press, Cambridge, Mass, 2000.

[69] Searle J: *"The Rediscovery of the Mind"*, MIT Press, Cambridge, Mass., 1992

[70] Pylyshyn, Z: *"The Robot's Dilemma: The Frame Problem in Artificial Intelligence"*; Ablex, Norwood, N.J., 1987.

[71] Kamel S. Saidi, Robert Bunch, Alan M. Lytle, Fredrick Proctor*,: "Development of a Real Time Control System Architecture for Automated Steel Construction"*, National Institute of Standards and Technology, Gaithersburg, MD, ISARC, 2006.

[72] Volpe, Nesnas, Estlin, Mutz, Petras, Das: *"The CLARAty Architecture for Robotic Autonomy"*, 2001 *also "CLARAty: Coupled Layer Architecture for Robotic Autonomy"*, 2000; Jet Propulsion Laboratory, California Institute of Technology, CA.

[73] Muller J.P., Pischel M.: *"The Agent Architecture InteRRaP: Concept and*

Application"; German Research Center for Artificial Intelligence (DFKI), Saarbrucken 11, 1993.

[74] *"JACK™ Intelligent Agents: Agent Manual, Release 5.2",* dated 10-June-05, Agent Oriented Software Pty. Ltd., Victoria, Australia

[75] Rao.A and Georgeff.M: *"BDI Agents: From Theory to Practice"*; Technical Note 56, Australian Artificial Intelligence Institute, April 1995.

[76] DARPA: *"J-UCAS – Commonly Asked Questions"*; 2005. Available at www.darpa.mil/j-ucas

[77] Pitarys M, Deputy Director J-UCAS: A presentation entitled *"J-UCAS: Common Systems & Technologies"*, Industry Day for Common Operating System Development", June 2004. Available at www.darpa.mil/j-ucas

[78] Ministry of Defence WebSite, http://192.5.30.131/issues/simulation/seba.htm, 20 September 2005

[79] Michelle Bevan, Sean Price: *"Simulation Based Acquisition: A US/UK Perspective"*, Presented at the European Simulation Interoperability Workshop in June 2003;

[80] Crosstalk: The Journal of Defence Engineering, August 2002 Issue. Article re-produced at: http://www.stsc.hill.af.mil/crosstalk/2002/08/easd.html

[81] United States Air Force: "*Unmanned Aircraft Systems Flight Plan, 2009-2047*", Headquarters, United States Air Force, Washington DC, 18 May, 2009