Chapter 4
UVS Technology Issues

The aspirational capability objectives for UVS in the air, land and maritime environments articulated by several national roadmap documents (e.g. [202] [205] [206] [267] [268] [278] [281]) may effectively be expressed as: 2008-10 conduct of ISR missions; 2015-2020 autonomous patrol; and, 2025-2030 strike capability and combat missions. This implies a need for persistent UVS autonomy[40] in complex dynamic military environments and the automation of a range of higher order or 'intelligent' functions. As a result, the challenges discussed in this chapter pertain mainly to next-generation UVS and pre-empt any 'validated' current military requirements. The chapter focuses on the complexities of contextual decision-making, planning in dynamic environments, verification and validation, sensory deprivation and trust and reliability for autonomous military UVS. It is recognised that there are a large number of other technological challenges[41] that may result in improved UVS, however, these have been covered by a range of other teams in studies cataloguing the state-of-the-art and projected requirements in each of these fields against various likely missions and applications (e.g. [44] [141] [202] [204] [205] [206] [230] [234] [237] [244] [245] [268] [276] [277] [278] [279] [280] [281] [282]).

4.1 Technology Challenges

UVS are subject to the laws of physics: they have mass and inertia, their moving parts wear, their electrical components emit heat, their sensors are corrupted by noise, no two systems are exactly alike, they fail and the environments into which militaries place them are complex, dynamic and unstructured. As a result, we cannot accurately predict their behaviour in advance and there are several factors that currently limit them from achieving their full potential. For example:

---

[40] It should be noted that requirements such as 'persistent surveillance' specified in several of the same planning documents differs from 'persistent autonomy' as the former can be carried out by UVS in combination with humans, whereas the latter requires complete independence.

[41] Including communications, sensing, signal processing, data and information fusion, systems engineering and integration, launch and recovery, human factors, platform, aero/hydro-dynamics, mobility, collision avoidance, mission planning/re-planning, propulsion, size, and energy storage.

A. Finn and S. Scheding: Develop., Challenge. for Autonom. Unmann. Vehicles, ISRL 3, pp. 55–103.
springerlink.com

The projected affordability of future autonomous and unmanned systems is higher than it needs to be. UVS do not require humans to be onboard and consequently do not need life support systems, space for the humans, special armour or protection, etc. As a result, UVS can theoretically be made smaller and lighter than their manned counterparts. As the procurement cost of vehicles is roughly proportional their mass (about US$3,300/kg) [141] [268] [278] a reduction in mass can be expected to translate into cost savings and a commensurate drop in the support required for the vehicle. However, current experience indicates that UVS and automation do yet exhibit a level of savings (say) enjoyed by computers. There is, of course, every reason to believe that as the technology matures, the costs will start to fall in line with the trends shown by manned vehicles, although further research into the hidden costs of operating UVS is also needed. For instance, at present, multiple operators are required for even single systems: a PackBot UGV may only require one operator, but it requires two people to transport it into the field and several more to protect the user. Similarly, Predator, Global Hawk and many other operational UAVs require a small number of people to operate the sensors and fly the vehicle, but significantly larger numbers to support it (planning, maintenance, image analysis, and so on).

Related to cost are issues of UVS survivability. That is, to survive, a UVS must detect, identify, classify, plan and respond to a threat. If a high degree of automation is employed and the sensing modality mixed, the cost of designing and producing such a UVS will likely be high (and the UVS may also be physically large). It is then more difficult to justify the sacrificial use of such systems. Alternatively, if the system is inexpensive a human will likely be required to interpret some or all of the sensed data. As a result, response options will be delayed and the likelihood that the vehicle will be lost increases. The probability of losing UVS is also linked to the reliability of sub-systems, which tends to decrease with an increase in system complexity. A second order effect is that as system complexity increases so does the level of mission complexity in which the UVS is employed, which increases the likelihood of losing the asset. Another impact of increased autonomy is a reduction in the communications signature, which allows more covert operations to be undertaken, although this is often offset by an increase in the sensing modality required, which may lead to an increase in other signatures such as radar cross section or an emission signature in another band.

Linked to cost and survivability are issues of affordability: the more survivable the UVS the fewer that need to be acquired. Although this needs to be balanced against the likely attrition rate that will come with the increased mission complexity and threat exposure that they will experience. Similarly, although an increase in the level of UVS autonomy theoretically leads to a reduction in the number of human supervisors required to operate it, there may be an increase in the maintenance and training requirements.

Another key technology issue at present is that many EO sensors are able to detect almost all of the light entering the camera aperture, with sensor noise near the lower limits set by the laws of physics. Thus, the challenges for these cameras lie not in improving the sensitivity of the sensors to light, but in increasing the size of their imaging array and hence the capacity of sensors to have sufficiently broad fields of view and resolution to allow detection of entities at long ranges so that sophisticated image interpretation techniques can perceive and 'understand' the key elements in their environments.

These techniques and sensors then need to be combined with all-source data fusion and advanced machine learning or adaptation techniques to make the perception more robust and insensitive to environmental variations. This would allow mission and path planning beyond a platform's organic sensor range and greater persistence in the battlespace through the provision of continuous, all-weather, 3-D terrain and target classification, mapping and localisation. Such improvements would also allow detection, recognition and interpretation of human, vehicle and other threat activity such that the UVS could distinguish friend from foe and anticipated versus unanticipated movement, thereby improving survivability and their capacity to operate in shared environments.

UVS also need IDT capable of making plans relative to a leader, manned vehicle or environmental changes so that they can adjust their resource usage or properties, join or leave teams (for example relative to communications, sensor scheduling, surveillance points, target kill, etc). Similarly, we need IDT that allow UVS to independently identify and make intelligent, complex operational and tactical decisions (e.g. self-concealment, lethal or non-lethal self-protection, avoidance of threats, and mimic leader action).

Linked to this is the need for communications and image compression technologies to be developed that allow beyond line of sight (BLOS) transfer of high resolution imagery and sonar data between UVS and their manned counterparts. For example, the most capable underwater systems currently achieve around 10kbps, but have a high signature for detection. Alternatively, other, LPI systems are typically able to communicate at rates of only 2-3kbps over ranges of around 10km, although using larger arrays or techniques that predict propagation conditions longer ranges and larger bandwidths are possible. Similarly, while RF propagates freely in the earth's atmosphere up to about 100GHz and it is possible for small directional antennas (~ 20cm) to be combined with low power (1W) amplifiers and then used to exchange data at rates approaching 10Gbits/sec between a UAV[42] and its GCS more than 100km away, these systems are still far too heavy and large to be of use to UAVs in the small-medium sized class.

Launch and recovery of UAVs and UUVs from ships is also a major issue. Fortunately, many longer endurance UAVs fly slowly so they can take off and land at speeds similar to those of ships at sea. As a result, it is necessary only to contribute to or absorb a small amount of the UAV's energy if the vehicles are appropriately aligned. The same is true for UUVs, although many do not travel fast enough to keep up with operations at sea. Furthermore, most ships do not want to wait for sea state zero (or stop) before launching or recovering a UVS, and recovery of any UVS at sea is a hazardous undertaking – even if it were only damage to the UVS that were being considered.

[42] This theoretical data rate will be reduced by a factor of up to 100 if anti-jam protection is afforded.

There is also a need to establish a clear product certification process for UVS that includes safety cases and regulatory regimes that address the very real dangers and the issues of public perception. Similarly, the application of autonomy to weaponisation and automatic target detection and recognition also needs to be addressed (this includes the related architectural designs). Related to both these issues are the use and safe manoeuvre of such UVS in the presence of people and other vehicles; the use and application of UVS within a human command and control network that changes; the level and modality of interoperability between different UVS and their control stations; developing flexibility in the levels of automation and adaptive interfaces; optimisation of vehicle-to-operator ratio for manned-unmanned collaboration; and, development of adaptive knowledge management systems for UVS.

All these and many other deficiencies relating to component technologies[43] of UVS are largely responsible for UVS not yet providing a persistent presence on our battlefields. However, as a number of highly qualified teams have published studies cataloguing the state-of-the-art and likely requirements in each of these fields against various capability projections, likely missions, and potential applications, and a full catalogue of the spectrum of technological challenges currently faced by UVS developers and programmers is simply beyond the scope of this book, this chapter focuses on the higher-order functions required to instantiate persistence rather than the physical ones.

- **Human-UVS Interaction:** UVS currently lack the ability to interact with humans and other UVS in an efficient and naturalistic manner that enables the human-vehicle system to perform a full range of complex tasks in unstructured environments. This is largely covered in the previous chapter, but discussed throughout this section.
- **Contextual Decision-Making:** Metrics for good decision-making, particularly for a context unspecified at mission commencement, are usually poorly defined. Understanding the basic patterns of stability and predictability for the decision-making paradigms is a pre-requisite for robust autonomy.
- **Verification & Validation:** The integrated and polymorphic nature of the sub-systems that make up a UVS combined with the requirements for stand-alone operation in a broad spectrum of unpredictable environments, which may be critical to mission success means that V&V, poses a significant challenge.
- **Trust & Reliability:** Trust and reliability are key issues that drive the levels of confidence and autonomy that we place in UVS. Currently UVS lack the capacity to understand their state such that they can predict their performance or detect functional or component failures autonomously, which affects our trust in them.

[43] For example, communications, sensing, signal processing, data/information fusion, systems integration, launch and recovery, human factors, platform, aero/hydro-dynamics, mobility, collision avoidance, mission planning/re-planning, propulsion, size, and energy storage.

- **Persistence:** UVS need to achieve improved performance over time, particularly in regard to repeated operations in the same environment, while learning from their experiences.
- **Dynamic Environments:** UVS frequently lack the ability to detect, locate and track moving objects while simultaneously accounting for longer-term changes in their environments.
- **Sensory Deprivation:** UVS perceive their environment through a limited sensory perspective, which may "blind" their supervisors; or force them to attend to the demands of a particularly burdensome task.
- **Robustness:** UVS lack robustness in the systems integration of their functional components and in the reliability of the system in dynamic environments. As a result, at some level UVS will malfunction and we are unlikely to be able to predict the specific nature or timing of these failures. Furthermore, UVS frequently fail, not through a manufacturing or design flaw but through routine dynamic loading, collision with an obstacle, or operators using them beyond their design limits. The operational need for cannibalisation of parts and specialist support has potential force implications.

4.2 Contextual Decision-Making

Reflecting on a widely used definition of intelligence [4], "the ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioural sub-goals that supports the system's ultimate aim," we can see that intelligent autonomy is conceived within the context of a UVS within its environment rather than independent of it. As a result there are three aspects associated with testing such autonomous behaviour [6]: novelty in the environment or in the problem to be solved, uncertainty regarding what is to happen, and dealing with difficult situations.

In this regard, the fundamental building block of good decision-making for automation is a high degree of Situational Awareness (SA),[44] where SA is defined as having three levels [96]: perception of elements in the environment; comprehension of the current situation; and projection of the future status. Issues for each component of SA include:

> ***Perception*** - Humans rely upon their five senses (or combinations thereof) to perceive their environment across the application domains. Their degree of success is often linked to their capacity to "notice things" while other events are unfolding. Many UVS ignore certain events as they are programmed to detect or interpret only particular ones.

[44] Here, for convenience, we include self awareness within the definition of situational awareness, although we shall return to discuss self-awareness in more detail in a later section (where we separate the two).

> ***Comprehension*** - Humans comprehend a situation by fusing their environmental perceptions with relevant contextual information and mission goals. Most UVS rely upon their supervisors to prioritise the importance and meaning of information, except possibly in regard to aspects of their navigational aspirations. For instance, some autonomous UGV navigation systems have the capacity to compute solutions for almost every environmental situation [243].[45]

> ***Projection*** - Humans make predictions on the basis of their perception and comprehension of a situation. Projection is frequently the most highly demanding cognitive activity and various stressors (cognitive workload, fatigue, stress, etc) can affect a human's capacity to fulfil this high-level task. Appropriate automation might ease this burden.

Unfortunately, novelty and difficult situations may be indistinguishable to an autonomous UVS. As a result, humans and the UVS may need to share their individual perceptions of the environment by developing and maintaining a common situational awareness picture. Consequently, the UVS information must be filtered, manipulated, and then presented in such a way that a user can quickly assess the status of the UVS (or the cooperative) and the battlespace it observes. If the mental resources required to accomplish this exceed the task demand, system performance will remain above the required threshold. In a high workload environment, when the demand imposed by competing tasks exceeds a user's capacity to process information, performance can be expected to suffer[46] [110].

To this end, automation needs to be introduced primarily where it replaces the difficult or complex UVS task responsibilities and presents the residual cognitive or physical tasks to operators appropriately. The problem, of course, is identifying the difficult high priority tasks for what is a dynamic decision-making environment. Furthermore, this information must be collected, processed, stored, and disseminated appropriately to those who need it, whatever their geographic location. Additionally, the selection of these responsibilities is dependent upon a number of factors that include the nature and complexity of the task, operational tempo, levels of operator training, experience, and so on.

In this regard, it is well-known that situational awareness has an effect on humans' abilities to successfully complete missions [95]. However, at present most attempts at improvements in human situation awareness focus on providing better interfaces between the UVS and its supervisor, allowing the human to carry

---

[45] If a UGV needs to traverse complex terrain a solution may or may not exist depending upon the width and mobility characteristics of the UGV. Alternatively, even if the terrain is traversable, the ease with which the UGV is able to execute its solution may vary.

[46] There is also a predicted drop-off in performance for low workload environments [295].

out the processes of determining his situation awareness better rather than capturing the machine's ability to observe, comprehend or make predictions (i.e. enhancing the UVS' ability to develop its own self and situation awareness and indirectly and simultaneously enhancing that of the user).

For decision-making to be distributed between the UVS and the human, a high degree of shared situational awareness is required. In a manned environment the devices that deliver shared situational awareness include spoken and non-verbal communications, visual and audio shared displays, and a shared environment [166]. Unfortunately, the bulk of these delivery mechanisms are not yet viable for a UVS and the sensed data must be pre-processed to convert it to a common reference frame, fused with state predictions based on historical observations, transmitted through communications interfaces, assimilated with other sensed data that have passed through a similar process to that described here, and then represented visually for interpretation and use by the cooperative's supervisor.

The degree of system automation required is fundamentally defined by the relationship between the human resource supplied and the situational awareness task demanded [251]. In this regard we must take account of several factors pertaining to a human's capacity to appraise his situation [95], including the limited cognitive processing capabilities of the supervisors. Humans are able to divide, direct and select their attention capabilities, but their perception is limited by their capacity to parallel process sensory events, sensor modality and working memory constraints and by their sensory channels. Consequently, complex or dynamic environments can quickly overload a human's attentive abilities such that they selectively sample their sensory channels. As a result, they typically manage their attention focus based on events, sensory updates, environmental conditions or task dynamics. Given that UVS frequently "ignore" information that would cause their supervisors to re-direct their attention, managing the attention requirements of supervisors such that they optimally sense and understand their environment is critical for environmental perception.

Although significant advances have been made in this area [90] [160] [197] [266], most solutions treat the task allocation, decentralised data fusion, and sensor scheduling problems independently. For instance, the effective allocation of a particular UAV within a team at any instant may depend upon sensor scheduling constraints imposed upon the payload. Moreover, as the number of UVS in the cooperative increases determining the required behaviour becomes more computationally intensive and complex. Similarly, emergent behaviour and unforeseen circumstances also become more common [166].

This means that due to the difficulty of forecasting the (probably emergent) behaviour of UVS, particularly within a networked or NCW environment, it may be very difficult to detect that something is going wrong. Consequently, another issue is how to provide diagnostic and feedback support to the UVS supervisors and their commanders, who may themselves be distributed over a wide geographic area, particularly as many supervisory functions are cognitive, hard to monitor, and embedded as components of other operations. As a result, rather than being able to monitor the individual tasks directly, the commander or supervisor may only be able to assess the systems outcome (i.e. the result of the autonomous cooperative's action).

In the context of defined tasks such as detecting and identifying targets, controlling and aiming a weapon, landing an aircraft and so on it is easy to understand what we mean by the phrase 'good' relative to an autonomous UVS; it is measured against the specific purposes of the designers and users. When we consider how to quantify the effectiveness an autonomous UVS making decisions between fulfilling a mission objective set by its commander and delaying achievement of this goal to satisfy other objectives (e.g. attacking an adversary en route to this objective) it is much more difficult to understand what we mean by 'good' as the metrics for such decision-making – particularly those pertinent to a context undefined at mission commencement – are usually poorly defined. The challenge for autonomous UVS is therefore as much based around the theory of work organisation as it is technical in nature; only after the basic patterns of stability and predictability have been thought through can UVS be productively applied.

For instance, a UVS control system needs to perform three basic tasks: avoid obstacles; avoid other UVS; and, operate the UVS within its performance envelope. Once these priority tasks have been accommodated, higher order tasks such as mission planning, surveillance, reconnaissance, target location, sensor scheduling, coordination, communication, etc. may then be undertaken. As stand-alone actions, the priority tasks are accommodated relatively easily as their goals are both decomposable and quantifiable in terms of physical quantities and closed loop control laws that relate to physical parameters such as lift, drag, thrust and so on.

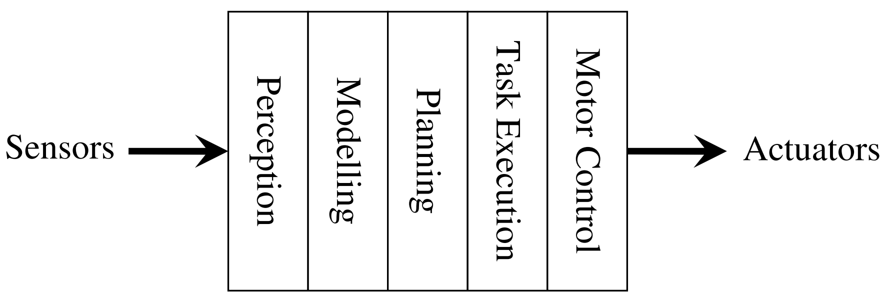

Fig. 4.1 Approach of linking perception to action through cognition (Adapted from [58])

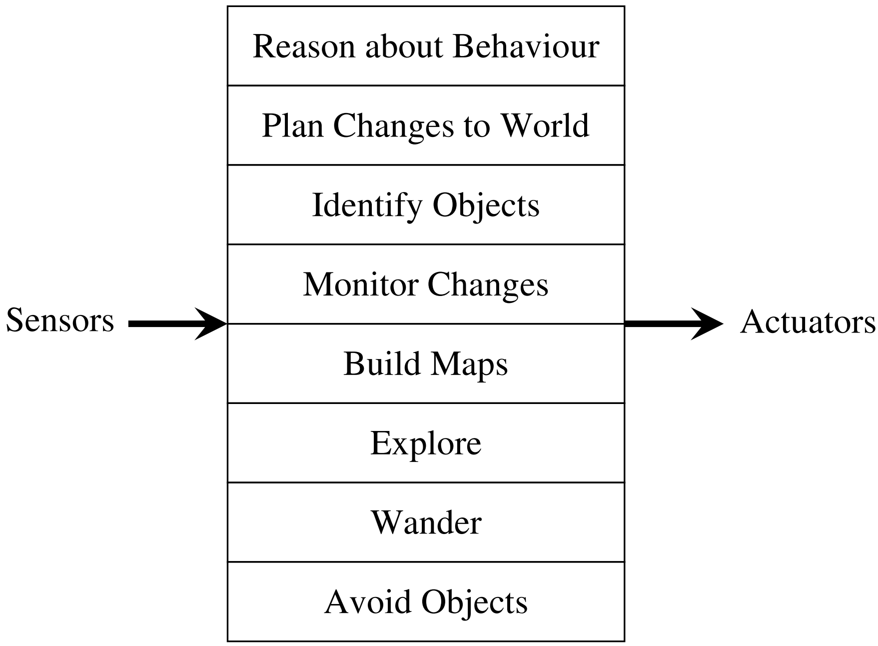

Fig. 4.2 Alternative decomposition of the perception-actuation problem (Adapted from [58])

Unfortunately, autonomous UVS also require the priority tasks to be closely integrated with the higher order tasks and there are at least two structural concepts for accommodating this integration. One in which perception is linked to action through cognition, where higher-level reasoning operates on the output of sensor-based perception to provide the necessary motion planning for actuation (see Figure 4.1); and another, bottom-up behaviour-based strategy where perception and actuation are more directly linked, without the need for detailed intermediate world models that relate one to the other (see Figure 4.2) [58].

There are several reasons why these higher order tasks are hard to instantiate:

- Higher order tasks are frequently more difficult for both humans and UVS to deal with as the decisions that they involve rely upon reasoning and judgement that are linked to the execution of higher order military command and control obligations.
- The problems are usually complex (for humans), which means that the problem is less well-understood and less structured and therefore harder to analyse or decompose into definable components. This means there is often a high likelihood of ambiguity, multiple possible courses of action, and/or the likelihood that one decision will impact a subsequent one.
- There are usually significant amounts of uncertainty, possibly conflict, both in terms of what is known a priori to any mission and what is observed during it. This means that determining optimality in terms of

any decision-making – that is selecting a course of action that has the highest probability of meeting any defined objectives – is much harder to compute with a high degree of reliability.

- It is difficult to accurately define suitable cost functions and metrics by which we quantify the benefit of applying one strategy or course of action over another. This adds to the complexity of the problem because we evaluate both the decision trade-offs and the quality of our decisions based on these.
- In addition to any 'decision cost' (the determination of one course of action over another), there is also an opportunity cost (the determination of a known course of action over one not considered), which is usually unknown or in-calculable.
- The prioritisation of goals is usually a subjective, contextual or interpretative task, for which it is not possible to anticipate all possible decisions and circumstances. Moreover, these tasks are often derived from direct or implied command and control strategies.
- From time-to-time it is necessary to re-frame an existing problem rather than interpreting the situation within the existing problem frame. Recognising this and dealing effectively with complex novelty and maintaining information regarding any decisions we may wish to make in this space (and in a form that is readily usable when we need it) is extremely challenging.
- Evolution has played a significant part in the development of human intelligence and its adaptation to the tasks to which it is suited.
- Other challenges include constraints imposed upon the problem space, such as time and mission constraints/obligations, mission complexity, and/or supervisory, environmental or adversarial interventions.

As [58] indicates, good decision-making agents reduce the complexity of the executive decision-making by breaking it down into component-decisions that are simpler to make. This also enables us to put greater structure into the verification problem (see Validation & Verification later) and to act on the sub-decisions and the information pertinent to each decision. It also aids in reducing the time it takes to make decisions by saving the UVS controller or the human supervisor from having to determine which information sources or sensors are relevant to any particular decision.

It may also result in information being presented in a more organised fashion, thus saving the operator valuable time when interpreting what might otherwise be a complex display. For example, if a controller or supervisor is only presented with sub-decisions that are possible at any instant, in-feasible decisions are automatically eliminated. If we then order the sub-decisions according to the hierarchy in which they have been determined we also have a mechanism for achieving traceability. In this way we can 'walk' a UVS operator through any decisions he needs to make while simultaneously recording any decisions made. This also allows interruptions (e.g. communications outages) to occur by providing continuity for algorithms and supervisors that need to bridge any such outages as they can return to the last relevant decision at the end of any outage.

4.2.1 Planning in Dynamic Environments

Military environments are inherently dynamic and if UVS are to maintain a persistent presence on the battlefield they must be able to adapt to their changing and often adversarial nature. In this autonomous UVS are no different from any other entities in the battlespace: they need to know what is going to happen next and what the best decision is now. Consequently, UVS require strategies not only for decomposing their missions into meaningful sub-tasks, but also for tracking progress towards mission goals and the changing nature of these tasks relative to the capabilities of the UVS. To do this they need to make plans and to establish a trade-off between the cost of any new plan (ideally compared to some global optimum) and the reaction time required to modify or repair any original plan given new information.

In this regard, there are two major steps involved in translating the problem-context into a solution framework, specification of the planning model and its evaluation function. In other words, the process of dynamic re-planning requires the creation of a model of the problem and use of that model to compute a solution. Consequently, when we solve a planning problem (or repair or modify a plan) we are actually only finding a solution to an approximation of the planning problem, which is a model of the real world. That is, we must find an incomplete solution to a problem that accurately represents physical processes, or a complete solution to a reduced (i.e. a simpler) problem [187].

The (processor-hungry) solution to this problem is to treat it as one of dynamic, constrained optimisation set in a time-varying environment and continuously re-compute and execute plans over some multi-objective cost function. Unfortunately, there are limits to the processing capability that most UVS can carry, tasks are time-constrained, the constraints and the solutions to the cost functions are typically only those that provide good approximations to the parameters under consideration, and the optimisation is often application-specific and depends upon the real world variables being optimised. As a result, it is usually preferable to find an approximate solution to a precise model rather than a precise solution to an approximate model. This is because if our model has a high degree of fidelity we can have confidence that the solution will be meaningful [188].

A decision must also be made to structure the planning algorithms either as complete or partial solutions.[47] The computational advantages of using partial solutions are attractive, but they are not without difficulties. For example, the problem must be organised so that the component problems can optimised efficiently and another evaluation function is required by which the relative value of the partial solutions can be determined. Furthermore, if the process is interrupted partial solution algorithms may not provide feasible planning strategies, whereas complete solution approaches should always be able to provide

[47] In complete solutions, all decision variables are specified and evaluation takes place by comparing two complete solutions; better ones replacing previous ones. In incomplete solutions, a complex problem is simplified by decomposing it into smaller, discrete problems that are easier to solve. When the partial solutions are all solved, they may then be combined and used as building bocks for the solution of the original problem.

at least one feasible plan. Unfortunately, while usually relatively simple to implement, complete solutions tend to be computationally expensive, as their main requirement is that the problem space must be evaluated exhaustively [9].

In this regard, there are many traditional approaches that can be applied to the problem, mainly because none is particularly robust to the broader problem space. In other words, if the problem space changes, so must the technique. As a result, many practical planning techniques tend to combine the benefits of reactive (local) and deliberative (exhaustive)[48] techniques, creating hierarchal systems that engage the low-level reactive planners under higher-level deliberative ones (or parallelised versions of the same thing). Depending upon circumstance, however, some scenarios are better serviced by deliberative strategies that execute closer to global optimality, whereas others achieve mission success faster using the reactive ones [9]. As a result, reliance on either choice can be poor in certain circumstances.

At a most basic level, a planner (e.g. for navigation) should prescribe a solution that is longer than the reaction distance of the UVS so that when new trajectories are computed it can avoid obstacles and other users of its environment. If this is achievable and the sensors onboard the UVS have sufficient range and resolution to perceive this environment then the UVS will at least operate safely within its environment while the higher level plans are computed. There are analogous safe planning solutions for other UVS behaviours such as weapons control, sensor scheduling and communications. For example, in regard to autonomous UVS weapons, we can postulate an acceptable generic architecture based on the major Principles of the Law of Armed Conflict (i.e. responsibility, military necessity, target discrimination and proportionality). This is discussed in greater depth in the section of Legal Issues, and in particular in the section on the Ethical Control of UVS.

A fundamental operating condition for most military UVS is that, once operating, most UVS systems cannot simply stop to compute a new plan every time the environment or circumstances change. Consequently, planning must be performed concurrently with normal system operation. There are several requirements:

- Robust plans are required to minimise the frequency with which successive calls are made to the planner
- When a call must be made to the planner, the repaired plan should only differ from the original plan by a limited amount
- In order to accommodate any limited deviation, the original plan should be readily adaptable to likely changes in the environment or mission
- When a call must be made to the planner, adapting previous plans and/or making completely new ones should take as little time as possible
- Despite the implied time pressures, the plans should be of a consistent quality

[48] Reactive techniques consider only recent and/or current information and produce local or one step-ahead strategies based on conditional responses. Deliberative techniques derive their recommendations based on all available information and strive for global optimality or a complete mission plan.

It is, of course, difficult to know a priori what any update rate pertinent to the dynamic decision-making timescale should be. Additionally, therefore, we will need to incorporate a degree of adaptive or reinforced learning into the prediction component of the planning algorithms to allow them to determine their own update requirements. That is, we will want them to have the capacity to learn from task and environmental changes in order to accommodate a better sampling frequency of sensor inputs and prediction outputs.

In this regard, evolutionary algorithms, which are essentially an adaptive combination of many techniques, show considerable promise. Regardless of the technique however most will compare newly generated solutions to existing ones and make some determination as to which solutions are feasible and/or preferred, and which need to be pruned or retained for further processing. This pruning process, however, is largely based on the evaluation function. Furthermore, it is usually assumed that the evaluation function is well-defined, whereas in reality problems are often set in noisy or uncertain environments. A key challenge in this regard, however, is whether to use an existing plan that is known to be sub-optimal or to wait for a better solution to be computed (the obvious solution – 'wait and see' – can lead to planning discontinuities).

Furthermore, each element of a planner must also detect that it has failed and inform any other components. These requirements are strongly linked to the desire to achieve persistent autonomy; that is, for the UVS to be able to determine conditions under which the prescribed mission tasks are unachievable, either within a required time frame or the broader capability framework of the UVS. By having this level of self-awareness, and notifying users of such limitations, the human-UVS system can then adapt accordingly.

For persistent autonomy, it is a fundamental requirement that the UVS be able to provide feasible solutions, and hence recognise those that are infeasible relative to mission time constraints, its own capabilities, etc. If the UVS determines to prosecute an infeasible plan, it has not really found a solution to its problems. That said it is acceptable for the UVS planner to work in infeasible space, defining solutions that it cannot achieve in order to determine those that it can perform. In this regard, however, there are some challenges for developers [187]:

- How do we compare infeasible plans
- Should we use an evaluation function for the feasible or the infeasible plans
- Are (or should) these two evaluation functions be related to one another
- Should we simply eliminate infeasible plans or attempt to repair them
- If we attempt to repair the plans, should we "move" them by the least amount
- Alternatively, is more radical "surgery" appropriate (i.e. more feasible solution)
- Do we need to find the delineation between feasible and infeasible plans
- Should we extract a set of constraints that define the feasible/infeasible boundary
- Having determined the infeasible plans, how do we translate them to feasible ones

As implied above, most planners cast their predictions as a binary problem for which the solution is either feasible or infeasible. Relative to the capabilities of the UVS, however, there may be areas of grey where solutions are simply difficult rather than impossible, particularly in relation to UUV operations near the sea floor or UGVs attempting off-road navigation. For instance, a bridge that is too narrow for a UGV might be considered a hard obstacle, whereas a steep slope might be considered a soft one; where a soft obstacle is one that can be negotiated by adapting UVS behaviour (e.g. velocity or heading).

One solution to such challenges is to explicitly compute cost functions that are defined in behaviour space (e.g. mobility maps)[49] [144]. These plans may then be treated as input to reinforced learning techniques that then learn by physically interacting with the environment. At present, however, even though the behaviour bounds of the UVS are relatively straightforward and well-understood, complex UVS-environmental interactions still lead to unknown and un-modelled factors. This means that the evaluation function is not crisp and application of reinforced learning strategies can therefore be complex.

4.3 Verification and Validation

“We're sitting on four million pounds of fuel, one nuclear weapon and a thing that has 270,000 moving parts built by the lowest bidder. Makes you feel good, doesn't it?”

Alan Shepard (Astronaut)

A number of studies have indicated that military personnel believe that only humans are capable of operating in a “free flowing environment of an offensive combat mission” [24]. However, trust and reliability really only guides rather than determines the reliance that humans put in automation and recent research has produced several seemingly conflicting findings [163].What is clear is that many military personnel do not want UVS operating in the same environment as manned platforms, particularly in hazardous environments. This is illustrated by the current need for a number of highly qualified humans to observe certain UVS and take control of them if they feel uncertain as to what they are doing. On the other hand, several studies [199] [26] [61] [86] [193] have demonstrated the human tendency to rely on computer-based recommendations, even though there may be contradictory (and correct) information readily available. This is usually referred to as *decision bias* [68] and typically results from the use of heuristics that people routinely use to reduce cognitive workload involved in problem-solving. It can result from errors of omission (where operators fail to notice a problem) to errors of commission (where people follow an automated directive that is wrong).[50]

[49] Alternatively, the degree of terrain ruggedness might be monitored through feedback from onboard inertial sensors and the UGV behaviour then adaptively controlled, as appropriate.

[50] Paradoxically, for imperfect automation the greater its reliability the greater the chance of operator over-reliance; this is because of the rarity of incorrect automation advisories with the commensurate result that the operator uncritically follows unreliable advice [220].

4.3.1 Trust and Reliability in UVS

Ultimately, fully autonomous UVS will need to achieve higher levels of reliability due to the very nature of these systems (for example, there will be no-one to change the UGV's flat tyre). Furthermore, the more a system is capable of doing autonomously the less human intervention is required and the greater the endurance requirements become. As a result, while endurance is typically measured in hours today, in ten years this may become weeks or even months. As a minimum, therefore, systems reliability must keep pace with mission endurance.

Fundamentally, there is likely to be a minimum threshold for reliability and autonomous UVS will start to be adopted by defence forces when they are cost-effective and have a proven, reliable track record. That said few technologies gain instant acceptance when introduced onto the modern battlefield as warfighters often inherently dislike or even distrust a new system. As experience is gained, however, reliable technology tends to earn the trust of its user community and the value of the capability enhancement is appreciated. Trust and reliability are therefore key issues that drive the level of confidence – and hence degree of automation – that we place in UVS. Moreover, trust in automation, and technology more generally, is a multi-dimensional construct that changes with time. It is influenced by the types and format of information received by humans, their individual approaches to developing and determining trust, and influences such as system capability and reliability. Moreover, users of UVS frequently trust malfunctioning equipment and/or mistrust equipment that is operating correctly.

These imperfect relationships are described by [219] as the "disuse and misuse" of automation. Misuse refers to failures that occur as a result of humans inadvertently or inappropriately relying on automation, whereas disuse refers to failures that occur as a result of them rejecting the advice or capabilities of automation. The processes of disuse and misuse are often described as a binary process of engaging or disengaging in reliance, whereas the practice is often more gradual, complex, and the combination of multiple factors. Nevertheless, and even though many studies indicate that humans respond socially to technology (and computers in particular evoke similar reactions to human collaborators [201]), this simplification makes the topic easier to discuss and the modelling of key parameters more tractable.

It is widely acknowledged that while humans are very good at issuing high-level goals, managing uncertainty, and injecting a degree of creativity and flexibility into systems, they are also prone to disuse and misuse; where these biases are heavily influenced by experience, the framing of cues, and the presentation of information. To this end, UVS that provide inappropriately framed information may inadvertently reinforce the human tendency to use heuristics, and hence the potential for decision bias [201] [86]. Humans are also prone to physical and cognitive errors and it may be reasonably argued (see Legal Challenges) that any UVS sufficiently complex to take decisions on our behalf will likewise be prone to hardware, software and/or algorithmic errors, mistakes and failures. Humans often also become frustrated and confused when a machine does not do what they expect it to. Moreover, uncertainty in humans frequently manifests itself as hesitation or failure to act. Nevertheless, during this process, humans usually continue to gather additional information to improve their awareness of their environment or their confidence in a particular line of action.

UVS, on the other hand, rely upon their sensors, actuators and IDT to reduce uncertainty or improve their confidence levels. Unfortunately, their sensors can introduce or increase uncertainty as they often have narrow spectral or physical fields of view. Alternatively, the algorithms used by the UVS may employ heuristics to abstract data or events, which can introduce noise or erroneous data, thereby reducing confidence levels or introducing greater uncertainty [24]. As a result, algorithms are often fragile and variation in a sensor's data stream can result in poor classification or processing results.

Trust is essentially based upon a perception that is linked to organisational, sociological, interpersonal, psychological and neurological processes which should (but usually do not) influence the design, evaluation, and training approaches to UVS. Not surprisingly, therefore, the topic is complex and draws on a diverse range of research from a number of fields. Moreover, it has generated several definitions [69]. However, trust between humans and technology is essentially driven by a combination of the probability that humans can successfully predict the anticipated action of the technology *before* they can monitor such action and the reliance they have upon the technology [26]. There is a considerable body of work that shows that not only is trust important to mediating how people rely on each other in relation to task completion, but that this is also extended to the relationship between humans and automation. Furthermore, these studies also indicate that it can be observed and measured consistently [163].

The relevance of this is that, if we can measure trust, we can use it as a framework by which we measure the level of 'trust' a UVS might have in a human. That is, the extent to which it might be able to reliably anticipate any likely human behaviour. One of the complexities in this regard relates to the unpredictability of autonomous UVS in unfamiliar environments. Often, when working with humans we can anticipate their actions by vicariously placing ourselves in their situation, or we have trained with them and have gained knowledge of their likely actions through experience, etc. Autonomous UVS have a tendency to surprise even their developers, although this is also one of their greatest assets as they can provide unexpected solutions to problems that could not be pre-programmed into them. In hazardous environments (i.e. ones in which UVS are likely to be used), however, these unexpected actions can be very disconcerting for humans. Nevertheless, just as we would attempt to develop an understanding of a human colleague in such circumstances, say, based on past performance in more familiar surroundings we should be able to develop an assessment of the perceived capabilities of UVS.

Establishing reliable automation in UVS also brings with it the challenge of identifying tasks, task components, or periods for which leadership can (or perhaps should) be assumed by the UVS rather than the human. At the very least, given the high workload environment of the modern battlefield and the cognitive and processing limitations of humans, we will need to consider whether human supervision of all tasks and at all times is optimal. In other words, should the supervision of specific tasks be replaced by a more equal relationship that reflects true human-UVS teaming, or (in certain circumstances) is it more appropriate for the UVS to write the human out of the decision-making loop entirely?

To some the notion of shared leadership may seem a little far-fetched. However, let us consider a scenario in which a semi-autonomous UGV can vary the level of trust it places in its user based on the user's level of attention, health, workload, etc, all measured by psycho-physical sensors embedded in the HMI. If we were able to observe these human states reliably, this would provide us with a mechanism for adapting the level of autonomy assumed by the UGV and thereby provide a means for varying its behaviour. Appropriately implemented, this would build increased trust into the relationship. In this regard [59] indicates that it is unlikely to be sufficient for humans to simply understand the UVS decision-making; the UVS must also be given a means by which it can understand the (potentially dynamically changing) intentions of the human. Ideally, both elements of the team will then have the capacity to adapt to this mutual flow of information, thereby building greater levels of trust in each other.

However, it is imperative that appropriate levels of mutual trust be established as any distrust will result in a 'fight' for control. In this regard the concept of understanding the limitations of the members of a team has been shown repeatedly to be more important than the establishment of a trust relationship per se; studies indicate that teams will often preferentially develop techniques for achieving outcomes with 'faulty' members they do understand over 'high-performing' members they are not familiar with. Consequently, it is more important for humans and UVS to understand each other's goals and limitations than it is for each to know other's capabilities and enjoy mutual trust. We will return to the topic of measuring UVS performance in the chapter on Force Integration of UVS.

Measuring the level of trust in humans relative to their UVS is a major factor in training operators to develop advanced skills in collaborative activities as it allows initial biases to be reduced, provides knowledge about system capabilities, and applies a risk-assessment based on the behaviour of the automation. Moreover, human-UVS teams only become truly effective when humans know how to appropriately trust (and hence rely on) the automation as they can then use this trust to direct the UVS accordingly within the relevant context [111].

Unfortunately, different human roles (i.e. commander, user, team mate) require different types of interaction with the UVS and hence potentially different levels of trust (and hence ways of measuring it). Moreover, while many may wish to interact with UVS at a high level (e.g. "Are there any targets over there and, if so, engage them appropriately") there will be many occasions when interaction is required at a lower level (i.e. a user wishes to control a specific payload on one UAV within a heterogeneous team). When this occurs the outcome can be either synergistic or counter-productive, depending upon the team relationship, the familiarity of the human with the UVS and their mutual understanding of the context. To this end, successful outcomes frequently depend upon the UVS (or teams thereof) to act predictably and to support varying levels and/or frequencies of user-UVS interaction.

Trust is ultimately built on system reliability and predictability, and to a very large extent it is the system's architecture that combines and defines the interactions between the sub-systems – and in particular the system's tolerance to faults and 'erroneous' data arising from real-world interactions. To this end, it is the architecture that drives our ability to define and grow our trust in UVS. In the cases where we are considering UVS carrying and using weapons the system must not only be trusted and safe, but enact behaviours that are seen to be safe and trustworthy. That is, users and observers must feel confident that the UVS will only use weapons within the constraints of the Laws of Armed Conflict. To accomplish this, the UVS will require a level of status reporting, the capacity to explain any ongoing or planned behaviour, and the ability to 'ask for help' when necessary. We return to this issue in greater depth in the section on Legal Issues.

Technological reliability is also a key factor in the development of trust. For instance, if the systems reliability is relatively high users may come to rely on UVS so that the occasional failures do not substantially reduce the level of trust – unless the failures are sustained. Another factor might be the degree to which failures are detected or particular behaviours are more generally observed. Similarly, the ease with which manual over-ride can be enacted, the degree of user self-confidence, and the overall complexity of the task may also all prompt different task strategies from different users.

One final note; and as a number of 'controlled flight into terrain' UAV accidents have demonstrated, human error is not usually a function of just the human, but of the system inaccurately or ineffectively facilitating user understanding of how the system actually works [4]. In other words, one cannot remove human error by simply increasing the level of automation and removing the human operator as the extent to which the UVS is made less vulnerable to operator error through increased automation makes it more vulnerable to designer error during the design and manufacturing processes.

Clearly, training has the potential to minimise or mitigate some of this, although it has been shown that training alone cannot overcome issues of trust arising from many aspects of poor design. Providing the users with interaction paradigms that they are familiar with, for instance 'natural' (i.e. human-like) interaction through gestures and speech (or even UVS that have identifiable 'personality traits') has been shown to improve trust between users and their technology [53]. Over-reliance on such technology, on the other hand, can also result in poor systems monitoring and a reduction in overall performance, just as too little trust can also lead to over-monitoring, which detracts from a user's capacity to carry out other tasks.

4.3.2 Systems V&V for Autonomous UVS

Developers naturally strive to achieve 'best practice' by implementing basic rules of thumb, keeping their designs simple, providing suitable documentation and creating initially stable designs. Empirical evidence suggests, however, even using a combination of peer review, static code analysis, subroutine and algorithmic testing, unit testing, component testing, functional testing (including human, hardware, and software-in-the-loop), integration testing, system testing and qualification and acceptance testing that it is practically impossible to provide bug-free software on an unblemished processor past a certain level of sophistication [145]. This is due to a number of reasons:

- While the majority of production systems are built to a specified level of quality, they are also built to a budget and schedule;
- The complex, cluttered, dynamic and unstructured operational domain into which they are inserted differs substantially from their military test environments;
- The engineers and programmers cannot a priori anticipate all possible contingencies; and,
- The software involved usually contains many lines of code.[51]

Moreover, in a network of autonomous UVS we are considering the interaction between multiple software modules running on different processors, operating systems, and with architectures possibly unknown to each other in advance and in all probability across a range of UVS platforms and environments. In the final analysis, therefore, we can be reasonably sure that in addition to autonomous UVS operations entailing considerable fundamental uncertainty, at some level the system will probably malfunction and that we are unlikely to be able to predict the specific nature or timing of these failures.

Nevertheless, UVS are non-unique in that there are several examples of intelligent systems whose malfunction may have severe consequences. Such systems require a great deal of care in regard to their design, operation, and maintenance. Moreover, increased safety in these safety-critical systems must typically be traded against criteria such as usability, cost and performance. In order that safety is not inappropriately compromised at the expense of one of these other criteria, sound ethical judgements must be made. Typically, these cost-capability decisions are made on the basis of some statistically significant criteria such as "The life expectancy of a human shall not be altered by using such a system." However, such criteria do not provide us with any absolute measure of what constitutes *safe* or *known malfunction*. Clearly the system will need to be designed to 'world's best practice', but this on its own is probably insufficient; not least because such standards and practices are currently informal and therefore not legally binding (see Legal Issues).

As previously indicated, the sum behaviour of an autonomous UVS is a function of the interactions between its multiple interacting and independent elements and its human supervisors. Furthermore, as the degree of autonomy increases it becomes increasingly difficult to predict the sum state of the system. Moreover, the system is actually a function of a number of linked processing elements (hardware and software) and humans (programmers, engineers and users) and becomes significantly more complex as these systems are networked

[51] Standard V&V techniques usually deliver between 97%-99% (or higher) overall code-defect removal for embedded systems if all steps are carried out correctly or perhaps 85% for application and information systems software [93].

either to each other or other technologies. As there are multiplicities of architectures, data formats, operating systems, programming languages, compilers and communications protocols, not to mention an almost infinite variety of hardware combinations.

From a technical perspective, however, we may consider two systems elements: hardware and software. Traditional hardware systems embody much of their functionality in the components that comprise them so they are relatively easy to model, they fail statistically through use or external damage, and their reliability is fairly predictable. Consequently, hardware systems can be analysed relatively simply and straight-forward tests can be formulated to prove their integrity before permitting their operational use. Furthermore, engineers can usually solve the problems of poor reliability with hardware redundancy.

When functionality is instantiated in software, however, the sheer number of states and a lack of regularity usually makes it much harder to bound the possible failure modes, and hence to devise tests against them.[52] Furthermore, there are frequently many subtle and often unexpected interactions between modules. As a result, a complete analysis of all possible failure modes and their potential impacts may not be practical. Furthermore, redundancy does not usually solve software reliability problems as software fails almost always as a result of some latent design error. Hence, failure of a critical sub-component is often highly correlated with the failure of a duplicate backup system, unless different software designs are used. As a result, while it will increase cost, building-in redundancy may not improve the reliability of UVS.

An increase in the reliability of autonomous UVS will come from the development of affordable software Verification and Validation (V&V) strategies that reduce costs and compress production schedules. However, although there are a range of systems engineering and other analytical techniques available for evaluating the likely performance of software (e.g. [115] [186] [189] [167] [222] [223] [247]) and current certification practices have historically produced safe and reliable control software for many complex systems, verifying and validating software that controls the key functions of next-generation UVS poses significant challenges in terms of providing the requisite levels of confidence. As a result, current techniques are unlikely to be cost-effective for a number of reasons.

- Application of existing V&V strategies is a non-trivial undertaking [222] [223], and it is highly likely that modelling and measuring the reliability, usability, testability, portability, and understand-ability of the critical elements of the UVS software will be a major undertaking in itself. This is because almost all of the ‘intelligent’ functions in a next generation, autonomous UVS will be software modules that are likely to be distributed across a number of programs and processors with no one processor, program or programmer knowing the full extent of individual

[52] Using the ‘rule of thumb’ (see Footnote 27 in Looking Forward) this means that there will be around 91,000 test cases required for an embedded software system that has 10,000 function points, although this number could be much higher (perhaps by a factor of ten) due to specific test-driven development [93].

outcomes or determinations. Moreover, if the UVS forms part of an NCW environment, software modules originating at another node in the network could be executed within the UVS that has potentially formed on some ad hoc or other ‘unpredictable’ basis.

- Although many of the functional components of an autonomous UVS are likely to be based on independently mature technologies, the sum behaviour of the UVS will be a function of the interactions between these components, the human supervisors (via an HMI), and a range of other external technologies that may include other UVS.
- UVS are polymorphic because they interpret data from a number of different perspectives and manipulate information in accordance with environmental conditions, the nature of the mission, and the problem at hand.
- Some sophisticated UVS may be designed to be intentionally unpredictable so as to inject a degree of creativity into the UVS mission, as predictable systems are not necessarily optimal for military operations. From a V&V perspective, consistency is obviously desirable, but in an adversarial context the capacity to predict exactly what the UVS will do may well be disadvantageous. Consequently, a balance will need to be struck between consistency and unpredictability that allows the programmers to understand, trust, and hence verify the software.

Testing software will need to be geared toward the verification of four key, high-level requirements: the loss of control, survivability, UVS performance, and safety (which includes compliance with the Laws of Armed Conflict). There are a number of issues:

- The software designer is not usually an expert on sub-component design;[53]
- Next-generation UVS may replace human ability and judgement and our comprehension of higher order cognitive functions is not yet well-framed;
- Improvements in automated testing regimes may reduce labour costs and testing hours, but may not reduce them sufficiently relative to the emerging requirements.
- Most software requirements are incomplete (i.e. we will probably need to specify the unwanted as well as the desirable behaviour of the UVS and its intelligence);
- UVS will often be used in contexts for which they were not designed (i.e. we need to understand how the software operates across a broad environmental spectrum);
- Software is often changed (i.e. hardware fixes usually result in the recovery of a system’s functionality, but minor software ‘fixes’ may introduce new faults);

[53] There is evidence to suggest that embedded software engineers attend domain-specific events rather than mainstream computer shows or software engineering conferences [93].

- The impact of software changes are non-linear (i.e. a small modification may have significant – even catastrophic – results on system performance);
- The state of the UVS often depends on its past history in intricate ways that may involve several components or other sources of a non-deterministic nature;
- Adaptive learning techniques can adjust their own logic during execution and some software techniques have the potential for self-healing. Both would render obsolete any certification process;
- It may not be possible to exhaustively test the IDT software because the number of states is so large;
- Reliability criteria may be driven by payload or may take on other forms to accommodate the functionality of the remote user;

As pointed out in UVS Components, many of the functional elements of a UVS will be embedded software units with the potential for significant human impact. Consequently, the defect potential and removal needs to be monitored closely or there will be serious issues of liability (see Legal Issues). Furthermore, such quality control issues will clearly impact schedule, cost, and systems reliability. Finding and fixing bugs will therefore be the most expensive activity in UVS software development [93].

Having said all this, let us now return to the notion that a UVS control system needs to perform a number of tasks: avoid obstacles, avoid other UVS, operate within its performance envelope and, once these priority tasks have been accommodated, undertake higher order tasks such as mission planning, navigation, surveillance, reconnaissance, target location, sensor scheduling, coordination, communication, executive decision-making, etc. Now, rather than attempting to consider the autonomous UVS at the system level, we can apply techniques employed by [115] [167] and [186] and divide it into its constituent autonomous functions or categories of autonomous software. Analysing the individual requirements of these constituent functions then reduces the complexity of the V&V task somewhat.

Mission and Trajectory Planners

Planners typically make decisions by projecting action into the future on the basis of a model of the UVS, its current and potential behaviour and the environment, and then evaluating the outcomes according to a cost function or some other selected criteria. The evaluation function then represents the UVS objectives and constraints through the return of high values for plans that meet mission goals without violating the performance envelope of the UVS. Typically, this involves some form of search through a set of potential plans until an acceptable or feasible plan is found. Consequently, the key is to apply pruning techniques so that only successful plans are likely to be generated. There are four major risks (listed in order of increasing severity) [167]:

- The plan makes inefficient use of resources;
- The plan could not be generated;

- The plan generated is not feasible; or
- The plan places users of the environment, supervisors or the UVS at risk.

Clearly, we need to worry most about the last one. However, to assure ourselves that safety is not an issue we do not need to verify the entire planner, only its evaluation function. If the evaluation function is correct then the UVS and/or users cannot be placed at risk. Fortuitously, the evaluation function is likely to be based on algebraic expressions or software techniques that rely on physical laws and/or techniques that have been used for decades. The main difference will be the absence of human oversight and ‘double-checking’ of results. As a result, we must hold the evaluation functions to higher standards of verification, but this is more a matter of degree, than novel concept [115].

The generation of infeasible plans is more serious than the absence of a plan as the latter is simply a fault. That is, the system should be able to detect that it has not generated a plan and can automatically invoke some sort of recovery procedure. The generation of infeasible plans requires the UVS to have an understanding of its state and may therefore be verified by evaluating some of self awareness criteria (see Dynamic Planning).

The use of resources is context-dependent and should be evaluated thus. For example, an autonomous planner might propose to do something a human would not. However, in terms of verification, this situation should be considered against the case where the mission may not have taken place rather than against more absolute conditions.

Navigation

Verification of autonomous navigation software is very challenging as it is mission-critical and relies upon the complex integration of algorithms in the context of a system embedded in a complex environment. Furthermore, some navigation capabilities depend upon the self-awareness of the UVS and/or its capacity to cue sensors to maximise its potential for observing certain types of data and hence its ability to perceive and predict its environment in the presence of uncertainty.

The inability to fully or exhaustively test software is not a concern in and of itself as many non-trivial systems cannot be exhaustively tested. Furthermore, exhaustive test is not required to produce reliable software. For example, UVS systems – like software programs – have structure and what often passes for exhaustive testing is in fact only sparse testing from the range of all possible states. In other words, the behaviour of the UVS in one state is not always independent of its behaviour in other states. As a result, testing the IDT in one state may provide information about other states, which can be grouped with respect to particular properties of concern. The key to making the IDT reliable is then to design it in such a way as to make its structure testable; or at least to allow its states to be decomposed into a tractable number of groups with respect to particular properties of interest.

Executive Decision Makers

A key challenge for an autonomous UVS is the application of work organisation. In other words, only after the basic patterns of stability and predictability have been determined can the UVS be productively applied. Good decision-making agents are therefore analogous to sequencing engines as they reduce the complexity of the executive decision-making by breaking it down into component-decisions that are simpler to make. This enables us to put greater structure into the verification problem and also to act on the sub-decisions and the information pertinent to each of them. However, such agents typically have complex semantics and comprise multi-threaded algorithms, which are prone to race conditions, deadlocks, and non-deterministic behaviour [115].

Another issue for V&V is that irrespective of the technologies used, the mere act of removing the human introduces risk because IDT, which have control over the UVS, can make errors that lead to mission termination or system failure. However, if we consider the problem in context, the autonomous function is likely to have been introduced because of some human failing: an inability to react quickly enough, the monotony of the observation task, etc. That is, the technology is embedded into the UVS because the processes involved are currently unreliable as a result of their involvement with humans. To design a robust IDT, therefore, we simply need to repeat the process and aim to design it with high levels of self-awareness. In other words, when the software fails the IDT must never fail to recognise that one of its software components have failed. In this way, unreliable UVS components can be combined into an overall architecture that has a fall-back recovery procedure. Acting on the component decisions in this way also aids in reducing the time it takes to verify decision-making software by allowing the information to be presented in a more organised fashion, thus saving valuable time interpreting what might otherwise be a complex situation. For instance, a verification strategy might order sub-decisions according to a hierarchy in which they have been determined to achieve traceability. This allows us to ‘step’ through any decisions that might need to be made [222].

4.3.3 Simulation-Based V&V

Simulation-based V&V is a flexible framework for simulating, analysing and verifying autonomous UVS. Essentially, an instrumented test-bed consisting of the actual control software and processors is embedded in a simulated operating environment. Conventional and model-based testing is then combined: the real software and hardware is executed and verified rather than an abstract model derived from the system; yet the simulated environment allows execution ranges over an entire graph of possible behaviours rather than a suite of linear test cases [115]. Ideally, each internal software state is marked to identify that it has been tested to avoid redundant testing or note any variation.

Simulation-based V&V avoids the need for developing separate models for verification purposes and, more importantly, the need to scrutinise each violation against the real system to see whether it corresponds to a real or a modelling inaccuracy. On the other hand, while simulation-based verification provides important potential gains in scalability, automation and flexibility, it is generally less efficient than model checking verification techniques [247].

To enable controlled execution, instrumentation must be introduced into both the software under scrutiny and the test environment. Furthermore, if the test-bed is capable of iterating over all alternate events at each state, back-tracking to previously visited states and detecting states that produce similar behaviour it will constitute a virtual machine with a fully controllable state space. To constrain the state space, however, the environmental component of the test-bed must usually be restricted to a well-defined set of vignettes or scenarios. We may then use such a tool in three ways:

- By applying the simulation-based verification approaches described above;
- As infrastructure for developing a program framework for autonomous UVS; and
- As a framework for evaluating and diagnosing concepts of use.

Aside from the system under test, the tool will require three components [223]:

- A diagnostic component capable of interpreting the physical system;
- A simulator for the physical system on which the diagnosis is performed; and
- A driver for generating commands and faults according to user-provided scenarios

To verify the system, the tool should then run through all conditions specified in the scenario, back-tracking as appropriate to explore alternate steps and executions. At each step, the tool should also check for error conditions and, if an error is reported, record and report the sequence of events that led to the current state. Verification of the diagnostic software is also required as the key to the voracity of such a test-bed is its ability to accurately observe or infer information on the behaviour of the system under test. This is contextually dependent and must take into account the run-time conditions under which it should be possible to acquire certain information.

Finally, although not strictly the same as V&V, Accreditation must also be considered for autonomous systems. For example, how many hours and under what conditions should we test a UGV to ensure it does not lose control?[54] Furthermore, what protocols and safeguards must we instantiate and test to ensure that such systems cannot be intentionally or inadvertently subverted and do we even know whether this is a real issue, and if so, how to characterise this task? There are also issues of test infrastructure, such as whether or not the existing test facilities, designed mainly for manned systems, are adequate and (say) how test data will be collected when the instrumentation normally mounted on a vehicle is larger than the vehicle itself (e.g. a MAV or small UGV in a sewer).

[54] During the 2005 DARPA Grand Challenge, and without warning, one UGV that was performing perfectly well suddenly left the course and almost hit a building, only missing it because the chase vehicle activated the UGV's e-stop; not something that may be an option for vehicles engaged in combat.

4.3.4 Health and Usage Monitoring

Persistent autonomous military UVS operations will place great emphasis on health, usage monitoring, and fault detection, isolation and recovery systems as such systems must not only recognise that something has gone wrong, but also determine what has gone wrong; and leave the UVS in a safe state by restoring its functionality in the face of failure. In any system that interfaces with humans, however, the overall output will be affected by the physical or cognitive workload of the human and the limited physical and processing abilities of the UVS. Moreover, a UVS that is able to perceive its environment through limited sensor modality may induce or suffer from 'cognitive blindness' [118] when the UVS or its supervisor focuses (or fails to focus) on a particular environmental event; or attends to the demands of a particularly onerous task triggered by such an episode.

In other words, the separation between the operator and the UVS deprives the human of a range of sensory cues that are available to the pilot or driver of a similar manned vehicle. Furthermore, rather than receiving the sensory input directly from either the vehicle or the environment in which the vehicle is operating the UVS operator receives only that sensory information provided by onboard sensors via a data link. The sensory cues that are typically lost include visual, olfactory, auditory, kinaesthetic and vestibular input. For example, an actuator malfunction may be signalled to the pilot of an aircraft via visual, auditory, and haptic feedback. In contrast, for a UAV this failure may be indicated solely by perturbations of the camera image. This manifests itself in two ways. For tele-operated UVS this is felt in terms of the operator's moment-to-moment control of the UVS; for more autonomous UVS, the vehicle's health and status at any instant are unknown.

The end result is that a considerable amount of data must be relayed from sensors and systems onboard the UVS to the operators at the GCS. This data must also be processed and presented to the users in such a way as to simultaneously minimise their workload in regard to monitoring it and maximising their capacity to interpret and understand it, which is in addition to any information needed to maintain task situational awareness, control the vehicle or progress towards mission objectives. Furthermore, the potential for controlling, coordinating and monitoring the states of multiple vehicles using a single operator diminishes exponentially with the increase in the number of vehicles, unless the vehicle's situational awareness is determined autonomously. To avoid network latencies and communications scheduling problems (that are additional to any required for mission completion), this processing must take place onboard the UVS.

As a result, the absence of an embedded pilot or driver promotes the need for a Health and Usage Monitoring System (HUMS) located onboard the UVS. Such systems must autonomously process, interpret, and deliver meaningful information about the status of the UVS platform, its sensors, and sub-systems. The key requirements are that it monitors the performance of UVS at both the holistic and functional component level in order to detect anomalous behaviour, characterise its nature, extent and seriousness, and report it to operators within useful timescales. Ideally a HUMS will also attempt to mitigate any potential damage, perhaps by affecting a repair.

Typically, HUMS will make use of analytical models of hardware sub-systems to provide estimates of the anticipated sensor observations and/or vehicle responses to actuator commands. They avoid the additional cost (and weight) of redundant hardware and can determine lost functionality at a sub-system level. They employ hypothesis-testing and robust estimation techniques to detect and isolate these failures, which can correspond to failed actuators, sensors or other systems failures that cannot be adequately assigned (e.g. a UGV has become bogged in wet mud). Typically, the statistical tests also look for changes in the statistical properties of any variables so that the HUMS can perform prognostic analysis on the likely failure trajectories or adapt maintenance regimes.

The detection of anomalous events requires an array of suitably placed and networked sensors, a strategy for acquiring and then processing the data, knowledge of the operating environment of the UVS, and the potential impact of likely threats and stressors. Based on this schema any damage must then be characterised and prioritised in terms of the vehicle and/or its mission in order for the HUMS to autonomously determine the urgency with which a response needs to be mustered. In an ideal system, the HUMS will also use its array of sensors to deduce information relevant to events leading up to the anomaly to identify and possibly isolate its cause. Finally, the HUMS should formulate a response option in the form of a sequence of actions or recommendations to operators that are achievable within the window of opportunity pertinent to the seriousness of the anomaly.

Clearly it helps to anticipate the type of events or anomalies that a UVS might experience, and these may be broken into two broad categories: external (environmental) anomalies and internal (vehicle-based) anomalies. External anomalies are likely to be dependent on the environment and therefore the type of platform or mission. For example, mud and water may enter the mechanical systems of UGVs and UAVs may suffer from icing on their wings. On the other hand, internal anomalies are likely to be broadly similar across UVS from each of the environmental domains even though their nature, frequency and severity are likely to be vehicle-specific and/or dependent upon operating conditions (and hence indirectly lined to their environments). Examples of internal anomalies include the failure of functional components (sensors, navigation/control systems, communications, propulsion, energy storage, etc) and the mechanical failure or degradation of materials, structures or interfaces. Clearly, in order to be of use a HUMS must measure a spectrum of mechanical, electrical, chemical and software-execution properties over a wide range of temporal and spatial scales and adaptive and reinforced learning techniques are particularly useful in determining the frequency and location of any sampling regimes.

Adaptive learning techniques are particularly useful for fault detection and diagnosis relative to unanticipated events. There are three primary categories of technique: model approximation, supervised learning and adaptation, and reinforced learning.[55] The regression techniques typically employ the use of

[55] These techniques have been used to model complex and non-linear systems such as aircraft flight dynamics, space vehicle control systems, jet-engine combustion, and helicopter gearboxes [202].

networks of radial basis or other functions to represent complex physical processes that are otherwise hard to model. These are then used to generate models for hypothesis testing within state estimators. Frequently, these techniques are supported by simulation data to provide an initial training set, whereupon they are then supported by data collected during field trials and operations.

The supervised learning techniques use a learning paradigm to select an optimal or good action to be implemented given the current state of the system. The learning is said to be supervised as the selection of a good action is based on a network of coefficients trained through human supervision or simulation. Once the system has been trained via this supervision, the system has the ability to generate 'good' actions given an arbitrary system state. Reinforced learning techniques are currently immature, but are capable of learning without a priori knowledge of a value function; that is, the technique learns the value function and evaluates goodness 'on the fly.' Reinforcement learning techniques are typically computationally intensive and are not usually able to run in real time on PC-based architectures. One such technique might learn a model of the vehicle and run it 'backwards' – i.e. take raw sensor data and commands sent to the hardware and find the most likely state of the given model that explains the observed measurements. However, in practice, the quality and robustness of this technique is likely to depend entirely on the accuracy of the model [291].

4.4 Multi-vehicle Systems

While it is relatively easy to build larger UVS that operate long enough and can travel far enough to perform useful military functions, these UVS are usually very costly to acquire, run and operate. The development process for many of these larger military vehicles also parallels that of their manned counterparts, which stresses longer life, higher levels of maintainability, multi-role capability and high reliability. The resulting systems are therefore more expensive with life-cycle costs and logistic complexities approaching those of manned platforms. Moreover, the continued drive for cost effectiveness, stand-off weapons delivery, precision engagement, the pressure for smaller operator footprints and higher workload environments, and the capacity for cooperatives of multiple UVS to accomplish tasks that are difficult or impossible for single UVS have all combined to increase interest in networks of smaller unmanned vehicles with increased automation.

As a result, Affordably Expendable[56] multi-UVS cooperatives are gaining prominence as they can be developed to carry out high value, high risk missions that are beyond the capability or justifiability of larger, single-vehicle systems. There is, of course, no free lunch. Even though smaller, less expensive, lighter systems lend themselves to being placed in harm's way, and their spatial benefits present opportunities not afforded single UVS, they are generally less capable than

[56] The concept of affordable expendability relies upon the notion that the useful life of the capability is a function of its constituent payloads and technologies rather than the physical life of the airframe [267].

their larger, more strategic counterparts, which tend to have longer ranges and carry more capable payloads.

For example, to have a 90% confidence in the classification of a target it is generally accepted rule-of-thumb that an image must have 16 pixels across the narrowest relevant dimension of the target. Consequently, to (say) recognise facial features (1 cm resolution) from a range of 1km requires a camera aperture of about 10cm. In other words, very small UAVs, which carry small sensors, need to approach their targets relatively closely, while larger UAVs are able to stand off considerably further and achieve the same end. Since the size of the camera aperture is proportional to the range to the target (for the same image resolution), to recognise faces from 10km requires a 1m aperture. As a result, if high-resolution images of significant swathes of the earth's surface are required, a high-altitude reconnaissance UAV needs to be relatively large to accommodate the necessary camera. Alternatively, a number of much smaller and lower flying UAVs must cooperate to achieve the same end – and must fly much lower. Then again, another effective operational combination is to have larger, high altitude detector/classifier UAVs cross-cue smaller "examiner" UAVs. This combination also lends itself to lower resolution imaging radars that can probe clouds, working with higher resolution optical imagers that do better at lower altitudes in clearer, cloudless atmospheres.

Furthermore, an equivalent problem to the above EO example exists for the acoustic sensors used on UUVs as the smaller UUVs cannot carry the larger, longer-range sensors. As with imaging radars, to some extent the physical laws limiting acoustic sensor resolution can be overcome by single or multi-vehicle Synthetic Aperture Sonar (SAS). Moreover, larger UUVs that cannot approach their targets closely enough to overcome the limited transparency of water can deploy smaller UUVs that carry optical sensors and can approach their targets more closely than their larger counterparts.

This lack of individual capability may be offset by the increased affordability of the multi-vehicle systems, our ability to derive process gain by networking the UVS and sensors (potentially achieving multi-aspect SA across the environments) and our capacity to withstand losses due to conflict or malfunction. Furthermore, a distribution of autonomy throughout multi-UVS cooperatives provides redundancy through the system's ability to re-allocate tasks and objectives, thereby increasing the number of objectives that can be met and the overall probability of mission success.

The endurance of a UVS depends upon its stored energy divided by its minimum power requirements and energy storage density for any given material is fixed. As a result energy storage scales (approximately) according to volume. Consequently, the range of a UVS is roughly proportional to the cube of its characteristic dimension, limiting our capacity to build arbitrarily small UVS. This presents practical problems of getting the (usually) slower and lower altitude UAVs to their required locations if they are not launched locally. One attractive option in this regard is – when they work with larger UVS – to have the larger ones deploy the smaller (usually expendable) ones, that can then be used for final target confirmation.[57]

The development of arbitrarily large networks of small UVS, however, is constrained by the requirements of internal communications essential for UVS coordination and network functionality as these smaller UVS must also contend with the inverse square law for omni-directional communications range requirements, at least for signal acquisition. A range of other key considerations for multi-UVS cooperatives include [108] [151] [174]:

- The number of assets in the cooperative could potentially be large
- Scalability is desirable as UVS may leave or join the cooperative
- Humans must be able to set goals for and interact with the UVS
- The health of the UVS and their sub-systems need to be monitored
- Each individual UV in a team needs to possess its own complex behaviour
- Each team within the cooperative should possess its own complex behaviour
- Humans can be supervisors as well as controllers of individual UVS or their payloads
- Supervisors and UVS are potentially distributed over a wide geographic area
- The integration may take place within a single environment or across them
- The cooperative should exhibit a highly fluid team-tasking and structure
- Operations occur in an environment that displays adversarial behaviour
- Situational awareness events can require a high speed response
- There may be various supervisors (of varying authority)
- There is a high probability of losing resources

Unfortunately, most UVS require the full attention of at least one and usually two or more skilled operators, and the ratio of personnel-to-vehicle rises to around 4:1 for even the small tele-operated UGVs when maintenance is taken into account [234]. This ratio is significantly higher for larger UVS such as Global Hawk, where the ratio is closer to 20:1. Clearly, given that most humans cannot manage multiple high-speed cognitive tasks in parallel significant advances in automation are needed if multi-UVS cooperatives managed by a small number of humans are to become militarily and economically viable.

To this end, there are a number of variables that must be considered when determining effective operator-to-vehicle ratios [108]:

- The spatial and temporal complexity of the environment
- The cognitive workload, training, experience, etc of the users
- The level of trust exhibited by the users and the reliability of the UVS
- The adversarial nature and/or temporal dynamics of any human tasking

---

[57] An example of this concept is the Finder UAV, developed by the Naval Research Laboratory which can be deployed from a long-endurance Predator UAV.

- The low-level (“navigation and mapping”) capabilities exhibited by the UVS
- The degree of high-level (“task-organisation”) automation exhibited by the UVS
- The capacity of the UVS to dynamically adapt these levels of automation
- The amount and nature of information passed between the user and the UVS
- The extent to which any decision-making may be distributed and/or centralised
- The capacity of the UVS to autonomously form into or dissolve from teams
- The degree to which the UVS is able to monitor/adapt to its own state and health
- The degree to which the UVS/humans are able to monitor systems performance
- The degree of network and/or processing latency inherent in the system

Multi-UVS research has its origins in the 1980’s and the field is still new enough for none of the topic areas to be considered mature, although some areas have been explored more extensively than others. Initially, a great deal of the research was based on the social characteristics and behaviour-based paradigms of biological systems such as ants, bees and birds. This early work demonstrated that the use of simple, local control rules allowed robots to mimic the foraging, flocking, aggregation and trail-following characteristics of these biological systems. Furthermore, the introduction of dynamics into the simulated eco-systems allowed the multi-UVS teams to demonstrate emergent cooperation resulting from selfish interests.

This work was then extended to incorporate studies in predator-prey systems, although much of this work was carried out in simulation and much of it focused on the development and evaluation of various pursuit policies. As a consequence, adversarial engagement between multi-UVS, such as that found in higher order biological systems, tends to have been studied in domains such as robot soccer (e.g. [153] [158]) or from the perspective of expected capture times and the sensing capabilities of the pursuers [152].

Much of the early work also tended to focus on using reactive or deliberative techniques (see Planning in Dynamic Environments). More recent work has used the benefits of each, creating hierarchal systems that engage low-level reactive planners under higher-level deliberative ones. Using modern, powerful processors these hybrid techniques are now sufficient to provide dynamic planning solutions for single UVS, but not for multi-UVS cooperatives. In part this is because many techniques “repair” their previous plan by optimising against information observed in the vicinity of the UVS location; a condition violated when multiple UVS operate in a geographically dispersed formation.

The challenges for multi-UVS arise predominantly out of determining the strategy that maximises overall systems performance, where such strategy decisions include whether the control should be explicit or implicit, whether the origin of the tasking should be distributed or centralised, the extent of the communication, the complexity and power of heterogeneity versus the relative ease of homogeneity, and the nature of the individual motivation (i.e. selfish or socialised) [76].

To achieve task and resource allocation in dynamic, adversarial environments a number of researchers have used free market economic theory, auction strategies and biological inspiration [116] [45] [84]. Another classical approach is to start by building terrain or world maps and then develop and execute the relevant strategies in known environments. There are several techniques available for building maps, but most of the common ones are based on Bayesian estimation and Extended Kalman Filters (e.g. [48]). Unfortunately, even two-dimensional map-building processes are time consuming and computationally intensive. Furthermore, many techniques assume accurate maps and worst-case motion for the adversary, which with noisy observations and inaccurate maps usually leads to overly conservative policies for pursuing the adversary.

As a result, a number of researchers have now applied game theory to the problem and combined the map-building and pursuit-evasion policies into a single probabilistic framework, some with autonomous (UAV-based) supervisory UVS [283]. A number of researchers have also considered active evasion strategies based on partially observed Markov decision processes (POMDP's), usually based on vision-based sensors and executed in simulated environments [129]. Others have used optic flow to determine the number of moving evaders as well as their position and orientation [284]. All of these approaches, however, designate the roles of the UVS prior to the commencement of the games as either pursuer or evader; they do not provide, for instance, the evader with the policy option of countering their pursuers by becoming the hunter.

Game theory appears to provide this option, with another attraction being its capacity to model a multi-UVS task (such as search, surveillance and target tracking in an adversarial environment) within a framework that provides the flexibility to use different solutions or role-playing concepts: one based on the cooperative behaviour of the participants and another based on non-cooperation. Application of these concepts in the field of economics has accounted for the lack of altruism shown by participants, which has resulted in untenable cooperative frameworks – unless cooperation is enforced by a third entity. Additionally, as [263] has shown the non-cooperative Nash strategies perform better than the cooperative ones in the presence of noisy sensors, unreliable UVS or faulty communications. This is because the uncertainty maps derived from the contributions of each cooperating UVS changes with time in a manner unknown to the other agents. In such situations the cooperative decision-making breaks down.

Many of the multi-UVS coordination issues such as task allocation, path and trajectory planning, formation optimisation and pursuit-evasion strategies are now becoming well understood, although demonstration of them using real UVS in outdoor and unstructured environments (i.e. as opposed to simulation) has been rather rare. More recent research has focused on motion coordination within the context of behaviour coordination such as target search and feature-tracking behaviours. As a result, research into path planning and control, multi-UVS task/resource allocation, behaviour coordination and communications has become coupled. This is largely because the structure of the multi-UVS cooperative changes with time and the properties of the cooperative change with structure. That is, the position of a UVS within its cooperative's structure and relative to its goals determines its projected sensing options, prospects for information gain, and capacity to (say) accurately explore, map and locate key features in its environment; just as its inherent capabilities, sensing options, scheduling of payloads, and so on impact the potential UVS trajectories, behaviours, feature tracking accuracy, communications strategies, etc.

Recently several have researchers attempted to address these coupled tasks as a single technique (e.g. [197]), whereas previously the problems of and approaches to communications and sensor scheduling, feature tracking and trajectory control were largely de-coupled and addressed using independent algorithms and strategies and then combined using some form of executive controller or architecture. More recent work attempts to manipulate the sensing process in order to maximise the information gain and feature location estimation, without using any a priori information. When the sensors are passive, this introduces a number of aspects that are not under sensor control (i.e. when precisely observations are made and what the observations are of), both of which have an impact on the development of longer-term scheduling strategies for the sensors and UVS.

Multi-UVS behaviour is often instantiated through the coordinated grouping of individual UVS into teams, the members of which take (or are instructed to take) a decision to commit to a particular task but who receive common reward for task achievement as a result of team decisions. The team members receive information about their environment and progress towards their task through observations and communications with each other, whereupon they take decisions based on their respective information. Teams can be self-organising or commanded through a centralised authority (and hybrid schemes exist also). In the case where the teams are self-organising, information may be explicitly or implicitly shared, where explicit communications is the specific act of conveying information from one UVS to another and implicit communications is the synchronism of UVS action through shared understanding.[58]

4.4.1 Multi-UAV ISTAR Example

For context, let us consider the case of a multi-UAV cooperative tasked with surveying a potentially hostile region of interest[59]. There are clearly a range of platform, mobility, propulsion, and energy issues that need to be addressed for such a system. As with the rest of this text, these are not dealt with here, except to note that the shortcomings and vulnerabilities of larger, slow-moving UAVs in this context are well known and have been described elsewhere (e.g. [102]). The cooperative must undertake a number of tasks:

---

[58] A classic example of implicit communications is lions stalking their prey. They do not communicate yet still synchronise their actions on the basis of their perception of the environment and a knowledge of the other lion's location, actions, etc.

[59] The example may be easily translated into a UUV, USV or UGV context.

- Based on a priori information about (say) target distribution and mission priorities allocated by a commander, the mission planning software must generate a series of near-optimal trajectories for each of the UAV to follow such that they visit as many regions of opportunity and interest as possible, while simultaneously avoiding as many hazards as possible.
- The optimisation of these trajectories must be based on (potentially time-varying) cost functions that allow for such things as: the distribution of payloads within the cooperative; the prioritisation of targets; the robustness of the proposed solution to operational and environmental uncertainties; the individual capabilities of the participating platforms; the benefits that derive from the association of the UAVs into teams; the communications and sensor scheduling requirements between the platforms to enable this cooperation, 'no-go' and 'difficult-to-go' zones and any UAV deconfliction requirements.
- Once underway, based on a change in the environment observed by one or more sensors onboard each UAV, the system must respond by dynamically re-calculating trajectories, re-allocating task/team associations and enabling payload and/or platform actions (within the constraints outlined above) based on the manipulation and fusion of the new data.
- Similarly, based on a change in the environment provoked by one or more of the UAV payloads or actions (e.g. jamming, UAVs joining the group), the system must dynamically re-calculate their trajectories, associations, etc.
- Based on a change in an operator's priorities or task objectives the system must respond by dynamically re-calculating their trajectories, associations, etc.
- Finally, all of the computational processing and communication must be achieved within the physical and electrical resources of the UAV and in real time.

In the mission-planning phases multi-UAV operations require multiple aircraft to be designated pre-defined flight paths, regardless of whether or not the UAVs have the ability to cooperate with one another. Irrespective of whether these pre-defined flight paths are generated using a route-planning algorithm or manually by an operator, the resultant trajectories must conform to acceptable levels of airspace deconfliction in terms of the temporal and spatial separation between the aircraft.[60]

If the UAVs are networked and can coordinate their efforts then after the mission plan is uploaded one or more of them may dynamically and continuously

[60] Trajectory deconfliction and collision avoidance for multiple UAVs within a single environment implies similar route re-planning requirements separated mainly by their time scales Deconfliction is a medium-long range task that attempts to avoid a collision while still allowing the UVS to remain within some predetermined navigation corridor, maintain time-on-target, conserve fuel, etc. Collision avoidance is a last minute, emergency manoeuvre aimed solely at preventing vehicle loss or damage – and does not take mission completion into account [268].

adapt their flight paths (e.g. in response to a target detection) to increase the effectiveness of their overall search strategy and/or capacity to prosecute the targets. Consequently, even if only one UAV needs to deviate from its pre-planned trajectory (autonomously or under the control of an operator), the rest of the cooperative must also have the capacity to dynamically adapt their trajectories safely. Additionally, constraints must also be placed upon the degree to which the manoeuvring UAVs are allowed to adapt their trajectories (for instance to fly within safe performance envelopes).

The deconfliction algorithms must also be able to accommodate ‘blunders’, where one vehicle in a cooperative deviates from its intended path for unforeseen reasons. In this case, other UAVs must then manoeuvre to avoid collision and maintain adequate separation. Generalised solutions to coordinated airspace deconfliction control and assignment problems for UAVs are non-trivial, particularly when there are cooperation constraints imposed (e.g. communications ranges and schedules, minimum or maximum airspeed velocities, collision avoidance, sensor field of view, scheduling, etc). Moreover, these generalised solutions do not usually lend themselves to an extension of simple two-UAV control and assignment problems [156]. Regardless, the environment must be monitored and the appropriate state information collected and disseminated within the cooperative so that an estimate of the current situation (i.e. UAV position, velocity, and altitude) can be provided to evaluate the likelihood of conflict and guarantee a period of conflict-free trajectory for each UAV while it carries out its higher order tasks.

Based on a priori knowledge an operator must also designate the location, dimensions, and orientation of an area of interest and whether or not it is known to contain objects of interest, targets, no-fly zones, communications dead-spots, areas of threat and/or terrain obscuration, etc. Thereafter, based on the payload configuration, target locations, priorities, etc flight trajectories for each UAV involved in the mission must be calculated such that (say) the probability of detecting the targets is maximised. Ideally, given an area of interest, an automatic route planning algorithm will calculate search patterns for the group (e.g. using probability maps divided into discrete cells [48]) optimised under constraints such as: maximise the probability of detection; minimise the time to detection, minimise the number of UAVs required; maximise the robustness of the search to aircraft loss; minimise the amount of network traffic required; and/or coordinate the timing of specific UAVs activities.

There are several techniques designed to search spaces for optimum solutions under multiple constraints. Broadly, the techniques fall into two distinct classes: algorithms that only evaluate complete solutions and algorithms that evaluate partial or approximate solutions. However, most traditional approaches suffer from either being too time consuming or getting trapped in local minima. This is primarily an issue for the dynamically unfolding component of the UAV cooperative’s task as during the pre-mission planning phase, centralised and potentially even computationally intensive team-based coordination and decision-making techniques can be employed. To this end, the required temporal and spatial allocation of payloads, tasks, and communications resources can be computed and iterated using centralised optimisation techniques and hierarchal command structures (e.g. exhaustive search, local search, greedy algorithms, branch & bound, divide & conquer, taboo search, simulated annealing, A* algorithm, evolutionary algorithms, etc) [188].

As the mission data must also be conveyed to rear command echelons these optimisation algorithms can also be run continuously as a means of (latently) evaluating the progress of the UAV cooperative against the specified mission criteria. There is, of course, a danger that these algorithms will provide false insight into the UAV cooperative's progress as the data observed by the UAVs in real time may only be shared locally (in real time) and a sanitised or processed version passed to the rear echelons.

Once the UAVs have commenced moving along their pre-planned trajectories, they must then autonomously and dynamically respond to the detection of targets and threats that might 'pop up'. These responses may be stimulated by either their own onboard sensors or those onboard other UAV within the cooperative. Alternatively, the UAV may need to respond to a change in priority imposed externally by the supervisors. In order to do this, each UAV must have goals and priorities assigned, which in turn requires a set of metrics that enables them (individually and collectively) to evaluate situations and events. These metrics enable the executive controllers to autonomously choose between competing goals, to assign resources to the task, and generate priorities that maximise pay-off and minimise cost. Ideally, both predicted and observed situations are evaluated so that resources can be allocated ahead of time and 'stand-offs' and conflicts can be avoided.

One of the main concerns for the distributed instantiation of dynamic control is that the controllers can potentially implement different strategies for the same goal (based on different perceptions observed by UAVs in different environments). Another is that in a large cooperative, the controllers may implement multiple copies of the same plan [242]. The latter is usually due to an inability in large cooperatives to share all the information observed by each of the members of the UAV cooperative. However, strategies for identifying these duplicate plans within the network and then pruning them exist [166].

In addition to the UAV responding dynamically (as individuals or collectively) within the wider cooperative to sensed opportunities and threats, in order to maximise the system's effectiveness, the cooperative may also need the capacity to form teams. The motivation for team-formation (or more accurately temporal-spatial task assignment) is to improve the probability of attaining specific goals – i.e. the detection or geolocation of an emitter or the capacity of another team member to carry out their intended function more easily. In order to keep the supervisory workload to a minimum, however, the team formation needs to be self-organising, such that the formation of a team is the result of forces acting within the cooperative and between the member UVS, as opposed to being imposed externally by an operator. Two other attractive properties of self-organisation are that any formation can potentially perform self-repair and that it can respond appropriately when unusual events occur.

The fundamentals of team formation require decisions on which UAVs are associated with one another and which teams are allocated to which targets or missions. These team-level goals must then be mapped to individual UAV roles – and there may be more than one role for each UAV within each team. Moreover, as the processing capabilities of humans are limited, particularly in time critical environments, the instantiation of the teaming architectures may need to be adaptive to accommodate variations in the levels of automation. Finally, mission completion criteria must also be established.

4.4.2 Multi-UVS Coordination

At present, the algorithms that successfully support teaming activities are well suited to tasks such as search, locate, track, identify, and engage targets because these tasks are quantifiable in terms variables that can be maintained within certain bounds. The challenge is to adapt these team controllers – that are themselves often the superposition of multiple controllers – to variations in noise, dynamics that are inadequately modelled and incomplete or uncertain sensor feedback in order to minimise the deviation of the specified variables from some predicted trajectory or end-state [9].

Additionally, given the number, variety and speed with which UVS are currently being introduced into capability around the world there is a need to extend our experimentally-derived understanding of operator-to-UVS control capacity to a more generalised theory or model. This will allow predictive modelling to take place within the relevant capability context and suitable architectures to be developed. However, if possible, we must attempt to do this by following the example of [76], [215] and [216] rather than just through the expensive and time-consuming use of technology-force insertion or human-in-the-loop simulation-based experiments. That is, we must ensure that the predictive modelling and the observations match, not just in relation to decision speeds (which is a common metric currently used), but also in regard to decision quality.

Most theoretical techniques for predicting cooperative behaviour depend on the expected time that a UVS may be ignored (know as Neglect Time) [73] before its performance drops below some acceptable threshold and the average time it takes to for a human to interact with the UVS to ensure it is still working towards its mission goals (commonly know as Interaction Time). Nevertheless, as the automation is not entirely reliable and failures do not occur at discrete, neatly designed intervals we must also account for the impact of the human decision-making process on the overall system performance. In other words, as most humans can only process cognitive tasks serially we must also allow for the time it takes for the operators to appraise the general situation (i.e. to notice that there is a problem within the cooperative) and the time it takes for them to gain situational awareness by focusing their attention exclusively on the errant UVS to discern its specific problems. We must also account for any time spent on distractions generated by other incoming problems or cognitive demands.

For example, using techniques developed by [76] we can then model the latency of human interaction arising from the overlapping arrival of UVS-related problems using queuing theory and – assuming the human is a single-server network – determine capacity predictions for human-UVS interactions. We may also extend these techniques by applying optimisation strategies that allow multi-constraint optimisation (e.g. evolutionary algorithms) and then use the outcomes of these computations to determine mission and cost-capability trade-offs between the larger, more sophisticated, platform-centric UVS options and the smaller, cheaper, distributed, network centric ones.

A current limitation with the predictive modelling techniques is their reliance upon assumptions or estimates regarding the interaction, neglect and waiting times. There appears to be very little experimentally observed data as it is difficult to measure and interpret. In this regard technologies that measure the psycho-physiologic relationships may be of use, but the techniques need further development and significantly more investigation is required [74].

One of key elements in realising the goal of multi-UVS coordination is the capacity of the cooperative to coordinate the actions of the different UVS that carry a heterogeneous mix of payloads; and a significant impediment here is that many existing multi-robot coordination algorithms elicit emergent behaviour such that the individual robots follow simple coordination rules rather than complex teamwork models or goals [211]. These techniques then break down because the UVS cannot explain their actions or role to other members of their team or the humans.

The need for the warfighter to be retained within the decision-making cycle means that in addition to the integration of the sensors and platforms, the information must also be combined, suitably manipulated, and passed to a rear echelon, where it is further integrated with applications that are of service to the user, such as geospatial information, track data and imagery, and visualisation and document management tools. This fused, value-added product must then be disseminated to users in near real time to allow the monitoring and redirection of the UAV cooperative, as appropriate.

In addition to this, there is also a need for multiple levels of feedback control [294], which in turn depend upon the capacity of individuals and the cooperative to measure and prioritise their performance and actions against a number of metrics (which need to be adequately defined in the first place), and their ability to communicate (in a meaningful fashion) the success of these endeavours, both internally within the group and externally to human operators, who may be geographically removed from their location and/or of a different command echelon.

In other words, a critical supervisory element for an autonomous cooperative of UVS is the feedback mechanism that allows the human operators to understand what, how, and why the system behaves like it does. Furthermore, research indicates that when human decision-makers are put in the position of passively receiving interpretations generated by data fusion and hypothesis generation aid machines they are less able to recognise emergent problems [154]. Consequently, there is a need to represent a range of levels and types of feedback control, which in turn depend upon the capacity of individuals and the UVS (or their cooperatives) to measure and prioritise their performance and actions against a number of metrics and their ability to communicate the success of these endeavours, both internally within any UVS groupings or externally to humans.

In addition to the more easily identified end results, the quality of the system and the team processes (i.e. the performance of the UVS, their constituent components, and the human interactions with them) needs to be taken into account. In this regard, [138] has used metrics such as efficiency (the percentage of a task completed vs. the amount of resources required), stability (the variability between plans and the degree to which different operators respond to similar plans), the degree of user engagement, the level at which the user delegates control to the automation, the extent to which a user's mental model of the system predicts the effect of adjusting the weight of a particular control loop, and comparative performance (how well does the fully automated system work against the human-automated system).

Before leaving multi-UVS coordination, we return briefly to the concepts of mission complexity and scale (introduced earlier in the section on Human Systems Integration).[61] Multi-scale Complex Systems Analysis (MCSA) [34] makes use of mission complexity profiles to specify the dependence of UVS mission complexity on the scale of action required. In other words, MCSA links the variety of possible ways in which multiple UVS (or their sub-systems) can act to the number of ways and the level of scale that a particular mission can be addressed. Thus, success of the UVS cooperative requires sufficient complexity at each required scale of action. In this regard, while high complexity in and of itself does not guarantee success, even well-designed UVS or their cooperatives will likely fail if they are insufficiently complex.

This has implications for the command and control (C2) structures for multi-vehicle cooperatives of UVS as it highlights the potential limitations of certain C2 structures. For example, as [35] [36] point out if we assume that each individual has finite complexity[62], in an idealised hierarchy only the leader can organise and coordinate the entire cooperative. As a result, the coordination between these UVS is limited by the overall complexity of the leader, which in turn means that organisational behaviours of the cooperative are limited by the complexity of an individual UVS. Since coordinated behaviours are relatively large scale behaviours, this implies that there is a limit to the complexity of larger scale behaviours of the cooperative, which means that hierarchal C2 is effective at amplifying the scale of behaviour, but not its complexity [34].

By contrast, a distributed or networked C2 arrangement can have greater complexity than that of an individual element; although it should be noted that while such a network is not guaranteed to have greater complexity than its individual

[61] Mission complexity is the ratio of the number of incorrect ways to perform a task relative to the number of correct ways to tackle it, where the more likely the wrong choice the higher the mission complexity. The scale of a task is the number of actions that need to be undertaken for successful completion.

[62] For example, in terms of their processing loads or capacity to communicate over fixed bandwidths.

components, it is possible for this complexity to exist. For high complexity tasks, therefore, we are likely to consider hierarchal C2 systems inadequate and will look towards more distributed structures. The recent tendency of military organisations towards more network centric operations and organisational structures and the evolution of massively parallel computing architectures also suggests recognition of the limitations of the hierarchal control structures. It should be recognised, however, that while distributed command and control is often discussed as a panacea for problems of hierarchal control, it does not actually correspond to a specific control structure. As a result, distributing control in and of itself does not lead to effective systems or the solution of identified problems: it is the instantiation of specific distributed architectures that are effective in addressing particular problems that provide functional advantage.

4.4.3 Autonomous Multi-UVS Task Allocation

Task allocation is the problem of committing finite resources to a number of coherent tasks based on the comparison and selection among a set of available alternatives. For military UVS cooperatives this must be attempted within dynamically changing and real-world constraints such as finite time or where the achievement of one task is a pre-condition for being able to undertake another. The tasking commitments may be temporal, spectral, or spatial in nature and while many constraints are usually understood a priori, the situational awareness of the environment at any instant or from any given perspective may only be partial or conflicted.

In the context of multi-UVS cooperatives, the task allocation process attempts to address the fundamental question, "Which UVS-payload combination should execute which task when (and possibly how) in order for the cooperative to achieve its global goal?" Fortunately, this problem is also central to problems in economics, biology (e.g. the division of labour in insect colonies), network allocation strategies, and multi-processor scheduling design. As a result we may take comfort from the number, quality and variety of researchers in the field. Moreover, economics, game theory and operations research all use the concept of 'utility' (also referred to as fitness, valuation or cost), which is based on the notion that each individual can estimate the value (or cost) of executing some action. Depending on the context, however, the utility may vary from simple directly-observable metrics to sophisticated planning techniques. The only constraint on such measures seems to be that they must each produce a single scalar value that can be ordered for the purpose of sequencing the candidate tasks.

For instance, in our multi-UAV case above, we might assume that each UAV is capable of estimating both the accuracy with which its payload is able to geolocate the targets and a resource cost (i.e. time of flight or the number of UAVs lost to enemy action). We may then combine these measures using an appropriate function. Regardless of the method of calculation, it is important to try to include all aspects of agent and payload state and their environment relevant to the utility function. Even so, the utility estimates will be inexact due to sensor noise, trajectory and target uncertainty, environmental change, etc, all of which limit the efficiency with which coordination can be achieved.

Many formal command and control models still tend to target medium to large scale systems composed of simple, homogeneous vehicles for use in relatively structured environments. Consequently, though simple and elegant, these models are insufficient for complex military tasks that require precise control. For instance, in our multi-UAV scenario, the UAVs will likely need to carry a heterogeneous mix of EW payloads; broadband ones to characterise the electromagnetic environment and cross-cue narrower band ones that are better able to provide high resolution spectral observations. The UAVs will also likely locate and track their targets using a number of different techniques, each requiring different observations (e.g. for geolocation of radar targets scan-ranging, triangulation through line-of-bearing, time difference of arrival, etc); they might also have different, but over-lapping, spectral views of the environment, different threshold sensitivities and so on. Similarly, if an enemy radar were to 'light up' unexpectedly and is identified as a missile control radar (a high priority threat) decisions must now be made about which UAVs or payloads must be tasked to work together to locate the radar as quickly as possible, but still taking into account their original tasking and objectives. What sensor scheduling strategies they should employ and what trajectories they should navigate (taking account of required accuracy, time on target, airspace deconfliction, no fly zones, etc) must also be accommodated.

If the UAVs have the capacity to communicate then they can inform each other about the value that they each place on the task (relative to any cost they may incur) and thereby reach a consensus with one another about who is best placed to carry out the task. It is then a matter of allocating the mission and trajectory deviations accordingly. However, in certain situations it may be prohibitive or impossible for the UAVs to explicitly communicate their task evaluations with one another. Also complicating the situation is that a lack of situational awareness may result in the UAVs not knowing what tasks they are likely to confront in the future. For instance, the closest UAVs may have been autonomously tasked to locate the missile control radar only to find that another, even higher priority radar then lights up that matches the specific spectral characteristics of their payloads requiring them to 'drop' their mission control radar task. Other UAVs, now ill-placed relative to their initial potential, must now take up the task of locating it.

Ideally we will be able to treat task allocation as a problem in optimisation. However, we must first decide what exactly is to be optimised. Preferably this will be 'system' performance but this quantity can be difficult to define and measure at any time, let alone during the execution of a mission, particularly if we include humans in the system. Moreover, as outlined in a previous section, when we select between alternatives the impact of each option on system performance is not usually known. Consequently, some kind of unifying performance estimate is required.

Clearly, the task allocation procedure must be adaptive, but under what conditions should a UAV take tasks when the opportunity arises and when should it ignore opportunities because experience has shown that a more appropriate opportunity is likely to arise in the future? Does this affect the number or nature of UAVs required to undertake the task? Or should we use more intelligent task allocation processes to distribute ‘commitments’ to each UAV and their payloads within the cooperative. Finally, how much of this task allocation process should be handled by humans and how closely should it be integrated with higher level military command and control functions?

A comprehensive review of task allocation procedures is beyond the scope of this text. Nevertheless, three key issues include: protocols, strategies and algorithms. At a protocol level we need to understand what type of transactions are possible and devise our message structures, communications scheduling and strategies accordingly. When designing the individual UVS we need to devise strategies that best exploit these protocols. This can include the provision of feedback to users or internally to the UVS cooperative in such a way as to provide incentives to the task allocation process to adopt a preferred profile of behaviour. At an algorithmic level, this means actually solving the computational problems faced by real UVS. In other words, algorithms that recognise that solutions are infeasible and call for simplifications of the task allocation strategies because a particular computational problem is too hard to solve within a finite amount of time. We also need to understand what commitments are to be distributed; how these are to be allocated (temporally and spatially); what procedure or mechanism should be used to distribute them; and, what the objectives are behind the distribution/allocation.

Task allocation between multi-UVS cooperatives fall into one of two categories (three if we include hybrid cases): centralised or distributed. Centralised task allocation systems tend to be hierarchal in nature with computational loads that tend to be very high and that usually increase with team size: a single entity allocates the tasking commitments, possibly after negotiating over preferences with the UVS (the central entity often acting as an ‘auctioneer’ in a form of bidding, e.g. [45]). The biggest arguments against using centralised techniques are the potential for single-point failure, the necessary centralised computational capability for large numbers of UVS, and the difficulty of (dynamically) assigning the master-UVS.

Fully decentralised systems have their computational loads spread across a number of geographically dispersed UVS, and tend to be communications-based. In these systems, tasking commitments tend to emerge as the result of locally negotiated steps, which are often restricted by bi-lateral communication (although systems that allow multi-lateral exchanges have been developed). Such systems also often suffer from a parochial view of their environment and tend not to be amenable to analysis so their precise behaviour is difficult, if not impossible, to predict.

Resources: A central parameter in multi-UVS task allocation is the nature of the resources themselves: some are perishable (e.g. fuel, bombs, etc) while others are static (e.g. payloads). Of the perishable resources, some are continuous[63] (fuel) while others discrete (bombs). This often influences how the resource can be

[63] The allocation of continuous resources has been studied in depth in classical economics.

traded. We should also distinguish between different types of resource. For instance, they may or may not be divisible (network access) or indivisible (payloads); consumable (fuel) or perishable resources (time); and, resources that do not change their properties over time are usually referred to as static. Finally, some resources are better understood as part of the allocation process as whether they are sharable typically depends upon the tasking procedure rather than on the characteristics of the item itself (e.g. sensor scheduling).

Priorities: Another key parameter in the task allocation process is the prioritisation of objectives, both in terms of how they are determined and then how they are represented and communicated between the UVS. Essentially, priorities express the relative or absolute concerns or precedence of an individual or group of UVS when confronted with a choice between alternatives. They are often closely tied to the context and hence the level of automation proscribed to the UVS. However, we need to understand how the priorities and objectives are determined and what techniques are suitable for coding and representing these priorities in terms of their expressive power, succinctness and suitability to task.

There are several options for representing and mathematically modelling priorities: evaluation functions comprising an ordered scale of quantitative (or qualitative) values; an ordinal relationship between alternatives (X is preferred over Y for X<Y but not if X>Y); a binary set of good and bad states (i.e. reductionist ordinal representation); and, fuzzy expressions that articulate the degree to which X is preferred over Y. The second option allows comparison of the satisfaction between alternatives but does not express priority intensity. Nor does it allow intra-UVS comparison of priorities. Qualitative measures allow a weak form of intensity to be expressed, but are difficult for UVS to interpret autonomously. On the other hand, the set of alternatives for the first and last options is a possible value of a given set of variables. In these cases the alternatives are huge and it is not sensible to expect the humans or the UVS to be capable of ascribing priorities against such a set. For this reason, there is a need to develop strategies, protocols and languages that allow compact representation of priorities and preferences.

Other key issues pertinent to complex task allocation include [174]:

- **Synchronisation:** Which tasks require intentional (as opposed to emergent) cooperation? What are suitable measures of cooperative behaviour to assess the quality of a task allocation within a given context?
- **Complexity:** What is the overall complexity of finding feasible and optimal solutions? How much of this process can be solved locally by each UVS and how much information needs to be exchanged between the UVS to achieve this?
- **Negotiation:** For multi-UVS cooperatives that rely upon distributed processing, what are the appropriate negotiation protocols and what are the most suitable strategies for employing these protocols? For those that rely upon centralised techniques, how can we devise efficient algorithms to support complex negotiation strategies?

- **Accuracy:** How do we devise negotiation strategies that force the UVS to report their priorities truthfully, both to reduce global complexity and to enable a correct assessment of cooperative synchronisation?
- **Implementation:** What are the best practices for rapid prototype development for specific applications? What constraints does the real world impose on theoretical models and how do different coordination strategies perform in practice? What is the practical impact of allocating infeasible task commitments and how computationally intensive are theoretically intractable results?

Regardless of the task allocation policy, however, most multi-UVS research has focused on the construction, demonstration and validation of working systems rather than the more general analysis of problems and solutions. As a consequence there are now a large number of architectures, many tested in working systems or in simulation, but the field still lacks a theoretical foundation that can explain or predict coordinated multi-UVS behaviour. In this regard, [116] developed a taxonomy for robot task allocation:

- Single-payload UVS (i.e. capable of executing only one task at a time) vs. multi-payload UVS (capable of executing multiple tasks simultaneously, but from a single location)
- Single UVS tasks (i.e. each task requires exactly one UVS to achieve them) vs. multi-UVS tasks (i.e. tasks require more than one UVS to achieve them)
- Instantaneous task assignment (i.e. the available information on UVS, tasks and the environment permits only instantaneous task allocation, with no planning for future allocations) vs. time-extended assignment (i.e. more information or predictive models of what tasks may be expected to arrive in the future are available)

This taxonomy allows more formal studies to be conducted as it characterises a range of multi-robot task allocation problems, providing the possibility of provably optimal solutions for the simpler cases and insight into the more complex cases.

One final point regarding multi-UVS task allocation, the current approach to command and control is largely human and platform-centric. As a result, the scale and nature of interactions between warfighting entities has historically precluded an autonomous coordinated response to threats – except that instantiated through human-to-human interaction. This is particularly true for smaller defence forces, although the network-centric paradigm is changing this. In contrast, in an autonomous, multi-vehicle UVS environment, where each platform is potentially presented with an abundance of information derived from a range of external sensors, the assets must interpret, purify and apply this information in a manner that prevents rapid error propagation before allowing self-synchronisation of any response. Furthermore, this non-trivial undertaking must be achieved within a framework of finite resources so that the systems may autonomously coordinate their response options. It is reasonable to assume that the early instantiations of such enterprises may have a capability edge in data processing, fusion and even operational tempo, but they may not equate to the levels of ingenuity, unpredictability and sophistication enjoyed by human-to-human command and control structures used by adversaries. As a result, such systems may have vulnerabilities that are exploitable.

4.4.4 Multi-UVS Navigation, Localisation and Mapping

For UAVs, with the exception of taking-off, landing and manoeuvring on the ground, navigation is usually relatively straightforward, albeit subject to the requirements of 'see-and-avoid[64]'. On the other hand, the navigation environments of military UGVs, USVs and even UUVs are usually unstructured and therefore much more complex and cluttered. Additionally, although many civilian UVS can theoretically navigate using EO sensors, battlespace environments can be expected to be opaque to such modalities (at least part of the time). As a result, all-source navigation estimators [11] that fuse multiple sensing modalities and, on the basis of those estimates, select sensing and navigation options that optimise information gain and UVS mission goals will be required.

The basic navigation challenge is to determine the location and orientation estimates of the UVS relative to an unknown number of environmental features (usually without an initial estimate of either), the location estimates of these features relative to the UVS, and the observed variation of these features relative to aspect, occlusion, UVS motion, time, etc. As a result, translating sensor data into maps for the purposes of navigation and mission-execution is an absolute requirement of a persistently autonomous UVS. Furthermore, as it is an integral component of the UVS control system, any errors in the world map reduce the reliability and safe-operation of the UVS and hence its potential utility.

There exists a large body of work that addresses these problems as they pertain to both single and multi-UVS navigation. One of the most successful techniques is Simultaneous Location & Mapping (SLAM) [166], which concurrently builds feature-based maps of UVS environments and obtains estimates of UVS location. These have been extended using machine-learning techniques for multi-agent systems, hybrid algorithms for multi-UVS control, multi-UVS localisation and map-building, and distributed sensor fusion [90] [238] [260] [266] [271].

---

[64] There is a general need for UAVs to fly in civilian, uncontrolled airspace. In order to achieve this, they will need to meet the requirements for visual flight rules at an equivalent or higher level of safety comparable to the 'see-and-avoid' or 'detect, see-and-avoid' requirements for manned aircraft (see Legal Issues). Detect, see-and-avoid is the process of trying to detect obstacles in the path of a UAV, determining whether or not they pose a threat, and, if necessary, taking measures to avoid them. There are a range of technologies (e.g. TCAS, ADS-B) that partially satisfy these requirements, but they only aid in avoiding cooperative aircraft. Other technologies (e.g. radar) are likely to be of use due to their all weather capabilities, but the weight, size cost and power of the equipment mean that they are unlikely to be considered practical solutions for small-medium UAVs. Such a system has two basic requirements: an ability to detect objects early enough to avoid them and an extremely low false alarm rate. It is also probably a requirement that the system function at a level superior to that considered acceptable for a human being as we have a tendency to accept 'human error' as a reason for failure, but expect autonomous systems to have a much lower failure rate.

More recently, techniques have been developed that are truly distributed and some of these techniques now take advantage of the properties of the distributed cooperative to achieve mapping accuracies unachievable with single UVS [261]. The majority of the work, however, operates predominantly in two-dimensional environments and relies upon environmental perception based on EO and LADAR. Regardless of the implementation, however, for multi-UVS navigation there are two broad classes of algorithm (or three if hybrids are included): one in which every control input and observation is passed between the UVS within the cooperative, and another where all the information is sent to a central node or 'mothership' running a single filter that estimates all the vehicle and feature locations [221].

The first of these techniques places significant bandwidth and scheduling requirements on the system and individual vehicles, while the second requires the central node to be aware of its own parameters (i.e. speed, orientation and position), as well as those of its subordinate units. As the structure of the SLAM navigation problem is characterised by monotonically increasing correlations between landmark estimates [91] and decoupling the state space is non-trivial, this places considerable computational load on the central node as the full covariance matrix must typically be updated with each prediction and observation of each UVS. Nevertheless, even though the mothership philosophy suffers from a systems level vulnerability of having one node that can be targeted as a single-point of failure, computationally effective algorithms capable of processing several thousand features in real time on high-end PC hardware have been developed [275].

A better approach is for the individual vehicles to build independent maps of their environment and for these maps to be fused together to form an aggregate, global map [291]. In a patch-work fashion, each vehicle can then add the current estimate of its local environment. The map data must then be correctly associated, both map-to-map for individual and between vehicles. This is usually relatively straightforward, even if a UVS joins the cooperative [292] as long as the location of the UVS is known either in global coordinates or relative to the other UVS. Under these circumstances, any new maps can simply be correlated to the global map. Furthermore, even if the location of the UVS joining the cooperative is unknown, the process is still tractable by building a map of the local environment and using this information to determine the relationship between the global map and the new local map [30]. It is also possible for the UVS to build maps that include feature estimates of the other UVS in their state estimators if the UVS are in the appropriate sensor's field of view. Once the correspondence has been established, the relative position between the reference frames may then be estimated. These techniques are also robust to communications that suffer from latencies or outages.

It should be noted, however, that all measurements have uncertainties so the location of the UVS and targets are only estimated as probability density functions pertinent to the regions where they are expected to be. As a result, single and multi-UVS mapping and localisation techniques tend to rely on the recursive use of distinctive environmental features or landmarks that, when revisited, aid the

UVS localisation process. This in turn also helps to keep track of the location of all landmarks. In order for features to be used as landmarks, however, their location needs to be estimated reasonably accurately. Consequently, multi-UVS map correspondence and location uncertainties arise from the use of environmental features that move, noise on sensor observations, and observations of features from differing spectral, spatial or temporal perspectives [275].

Motion tracking and estimation has been added to a number of mapping techniques [66] [197], although this is largely focused on sensors that provide both range and bearing information. For example, SLAM with Generic Objects (or SLAM with GO) [296] allows the addition of motion mode information (stationary, moving, move-stop-move, etc) for the landmark, although this has to be learned from the observations. The technique is straightforward, but computationally intense and not yet real-time. In another approach, called SLAM with Detection and Tracking of Moving Objects (DATMO) [66], each new moving object gets its own statistical estimator (typically a Kalman filter) and motion mode [109]. Vehicle state estimation then takes place separately and is used to update the Extended Kalman Filter (EKF) used for SLAM. This runs faster than SLAM with GO and is more suited to real-time implementation.

As the research stands, using the appropriate sensor modalities, as well as moving object detection and track initiation, the more advanced navigation techniques are able to determine when moving objects have coalesced, moved outside a sensor's field of view, or been temporarily occluded by a stationary object. They are also robust to long sequences of data and can adapt to false measurements and their extension to multiple vehicles has also been achieved (e.g. [273]). However, as yet, the techniques have not been extended to bearings only sensor modalities; are subject failure as a result of false observations arising from platform motion; and often struggle to accurately classify slow-moving objects or those that are temporarily stationary.

Image-based navigation, and scene and structure estimation derived from it through the fusion of external sensor information (e.g. INS), is also now solvable in real time [79] [226]. However, environmental dynamics have a deleterious impact on these techniques. That said if the dynamic features are characterised correctly they can be used to aid the mapping and navigation process, and vice versa [66].[65] When range observations and a priori information about the non-stationary objects are not available, however, it is not possible to determine their trajectories uniquely unless two or more sensors are used. When such observations are made from multiple moving platforms that do not have other means of localisation, stationary environmental features must also be used [109].

At present, most maps are usually classified in terms of statistical estimates of features described by data clouds, geometric returns, RGB pixel intensities, or through the use of occupancy grid-maps that are regularly updated. A more condensed approach relies on classifiers that interpret multi-modal sensor data in

---

[65] The algorithms must have already been initialised [81] and any recursive loop-closure already performed [81], which for bearings-only techniques is non-trivial. As a result, the mapping/localisation and motion-tracking problems have been currently only been solved separately and then integrated using range and bearing information.

terms of higher level descriptors such as ‘eucalypt tree’ or ‘bitumen road’. At present extracting such descriptors robustly and in outdoor environments is difficult and frequently dependent on aspect, background, lighting conditions, context, etc. Furthermore, the shear number and diversity of potential objects – and hence the resultant searching of any hypothesis trees – means that a priori knowledge is usually required in order to classify any objects swiftly and correctly. Additionally, high resolution observations of the objects, preferably at long range (and hence large quantities of data), are also a pre-requisite for most techniques.

Currently, the error analyses associated with such feature-extraction techniques are also not well-understood. As a result, the efficient coding of these descriptor-based maps and their integration with navigation estimators are yet to be achieved. Such techniques have been researched, but do not yet run in real time [117]. Furthermore, the techniques usually require spectral and geometric correspondences to be formed so the features can be ‘fingerprinted’. There is a difference, however, between the processing requirements of an algorithm that can (say) classify terrain that is sufficiently flat and devoid of obstacles for a UGV of specified mobility characteristics to traverse, versus one that can classify environmental features completely at the descriptor level. While the latter is possible on current PC-based architectures, there is a need to use 0.1-1.0TFlop processors if such operations are to be executed in real time [117].

4.4.5 Capability and Systems Integration

Multi-UVS integration – and particularly when it crosses environmental domains – has many technological impediments. Another challenge, however, is perhaps best illustrated using the following example.

Many countries acquire their military capabilities from overseas. Typically, such acquisitions might include UAV-borne ISR or strike systems, which are effective at detecting and neutralising concentrations of enemy forces on the ground, but have much more limited effectiveness when an adversary blends with his surroundings. Hence, while major force concentrations might be eliminated, smaller enemy groups that can protract hostilities may remain. As a result, an acquisition focus might be given to the provision of theatre or tactical-level tools for optimally selecting, deploying and managing sensor assets or the development of onboard control, coordination and decision support mechanisms for multiple manned and unmanned force elements (e.g. autonomous UGVs integrated with the ISR output from the UAV feeds).

Such technology would likely be one of the key outcomes of a control and coordination research program, which could have application across several major capability domains. Such developments would also probably involve information integration for manned and unmanned systems; a key element of sensing and data fusion research, again quite possibly applicable across several capability domains. The integration might also involve an analysis of the appropriate reliability and resource allocation issues pertinent to the provision of a persistent autonomous presence on the battlefield; the major focus of research in persistent autonomy.

The indigenous development or adaptation of such technologies and operational concepts to allow the integration of these unmanned and manned systems to support operations in complex, hazardous environments might be a national priority. However, it is unlikely that knowledge of the systems at the level required for close integration or multi-UVS cooperation would be shared between nations without significant risk or cost. Solutions may, of course, be available from other overseas vendors, or by integrating the systems less tightly, however, both these solutions are likely to be unpalatable for reasons of cost or sub-optimality.

From the technology integration perspective, therefore, the designation of a single lead agency for UVS to oversee the general and cross-cutting matters pertaining to automation and introduction into capability could be beneficial. By way of example, such an agency might also oversee such matters as systems engineering, life-cycle cost management, software engineering, the development of an effective assessment methodology, and the use of modelling and simulation assessment tools as many of the lessons learnt in one environment will be directly translatable to others.

Unfortunately, the likely acquisition strategies of many defence organisations, which are largely still platform and/or environmentally based, means that it is more likely that Navy will take a lead on UMVs, Army on UGVs, and Air Force on UAVs (and for some larger defence forces each agency will likely acquire its own UVS in each domain). Given the likely focus on operational exploitation and the individual agencies' experience in each of these domains this is not a bad thing, it simply misses the opportunity to enforce a more cross-cutting systems discipline on the generic requirements of autonomous UVS.

However, as these intra-service benefits are not yet well-articulated, it may reasonably be argued that these factors are trumped by the need to exploit the service-specific requirements of UVS; just as they are for other environmentally cross-cutting endeavours such as ISR and EW. Furthermore, without the clear and focused requirements advocated by the single-service or platform-focused capability initiatives, the process is likely to suffer from diffusion and incoherence. Clearly, it is likely to take a strong advocate in high office to advance any such notion. If such an agency were ever stood-up, it should take on the role of:

- Identifying gaps in capability that can be filled by UVS,
- Identifying technology shortfalls in autonomous systems and UVS,
- Influencing the development and assessment of UVS-related operational concepts
- Providing support to UVS planning, investment, and programs,
- Influencing the direction and level of UVS-related technology effort, and
- Developing and fostering cross-environmental UVS technologies and systems

That said, the evolution of military systems towards high-tech networks of automated capabilities that are responsive to a range of information sources, and the commensurate move away from the use of humans as the command and control 'glue' traditionally used to instantiate such enterprises, will likely result in this systems integration becoming a problem common to many military technologies, not just autonomous UVS.