

Investigation into the Interaction between Automatic and Autonomous Systems

A. Dunlop and R.T. Watkin
Roke Manor Research Ltd, Romsey, Hampshire, SO51 0ZN

Abstract

The requirement for automatic systems in a predominantly autonomous system is identified in this work, and the focus is placed on automatic protection systems present in vehicular systems. The impacts of an automatic protection response are explored for components of an autonomous system. The work further analyses methods by which an autonomous system could detect and identify an automatic protection response. Knowledge of 'normal' behaviour in an autonomous system is a possible method identified in this work for detecting when a disruptive event occurs; the concept of what constitutes 'normal' behaviour is identified as an area for further investigation.

Keywords: Automatic Protection Systems, Autonomous Systems, Plan Repair

Introduction

This paper reports on work undertaken within the SEAS DTC Innovation Fund project IF061 [1]. This study investigated issues surrounding the interactions between reactive automatic systems such as automatic protection systems (APSs) and deliberative autonomous control systems within a UXV platform. The aim is to characterise some of the automatic systems that a deliberative autonomous system might need to take into account, and to understand the implications of those automatic systems in architecting deliberative autonomy for UXVs. The work also addresses methods by which autonomous control functions can detect and respond to the execution of these automatic systems.

Key areas of investigation:

- Identify how plans are affected by an APS response and how to limit any disruption to those plans
- Explore what happens when an APS response results in mission goals being unachievable
- Investigate how the world model is affected by an APS response

- Identify methods of detecting and identifying APS responses
- Investigate how autonomy is affected by an APS response

Automatic and Autonomous Systems

In SEAS DTC 'Architecture for Distributed Power Management for Autonomous Unmanned Vehicles' (PPEM008) report [2] an automatic system is defined as 'a system that has fixed choice points, programmed with a number of fixed alternative actions which are selected by the system in response to inputs from particular sensors'

Automatic Protection Systems (APS) are a particular class of automation in a vehicle system which is concerned with executing a predefined action in the event of specific anomalous circumstances or threats. APSs are designed into vehicle systems in order to protect or preserve the vehicle, or to limit ensuing damage.

Some examples of APSs include:

- Full Authority Digital Engine Control (FADEC):
 - engine stall / surge protection
 - flame-out recovery
- Electrical surge protection systems

- Collision avoidance systems (e.g. TCAS)
- Fire suppression systems

Autonomy is also defined in [2]: ‘Autonomy is distinguished by the need for decisions to be made at any time, with some appreciation for the circumstance of the current situation (often referred to as situation awareness).’

Given the inflexibility of automatic systems in comparison with autonomous systems, in terms of their reactive nature and having fixed decision points and no understanding of the world, an obvious question might be, “why should the system not be fully autonomous?” The answer partly lies in what the main use is for automatic systems, even in manned vehicles, which is the protection of the vehicle and the safety of its operator and those who interact with it. In an immediate safety-critical situation it is vital that actions are taken quickly when there may be limited or insufficient time for the deliberation that would occur in purely autonomous systems. The immediate response to set triggers, as well as the guarantee that the response will follow predetermined procedures provided by automatic systems, may be what is required to allow unmanned systems to meet the demands of regulators and gain certification.

Automatic Protection Systems and Deliberation

The outcome of an automatic response will have implications for a plan that an autonomous system is following. For example an APS on a UAV could react to a power surge by disconnecting a faulty generator from the main power train. Plan execution could be compromised by having less overall power available once the generator has been disconnected.

An APS can compromise the applicability of a plan in two principal ways: modification of the constraint set, and breach of the system’s normal behaviour (as explained later in this paper) envelope. A plausible but expensive response to the modification or breaking of the constraint set is to perform a complete replan based on the new state of the world. However by understanding how a plan is affected by changes to the constraint set, it may be possible to devise a method of identifying those parts of the plan which need repair, without incurring the expensive computational burden of a complete replan.

Two major categories of planner, sequential and hierarchical, were considered to investigate further the effects of automatic protection systems on different plans. For each, a strategy for how to limit the extent of replanning necessary after an APS occurrence was devised.

The approach for sequential planners (those which produce plans as a sequence of tasks for execution, as though from a single self-contained program) is to annotate plan elements with the constraints impinging on them. When constraints change, or are removed altogether, then it is (notionally at least) easy to identify the plan elements in need of repair. This is illustrated in Figure 1.

Hierarchical planners produce hierarchies of representations of a plan. The plan is typically represented in gradations of detail in the hierarchy. The consequences of an APS response will likely affect several levels in the plan hierarchy. For efficient plan repair / replanning it is desirable to identify the highest-level plan in the hierarchy of plans which is no longer feasible following the APS response. When the APS interruption has been mapped to a particular level in the hierarchy, replanning consequences propagate in the hierarchy, as shown in Figure 2.

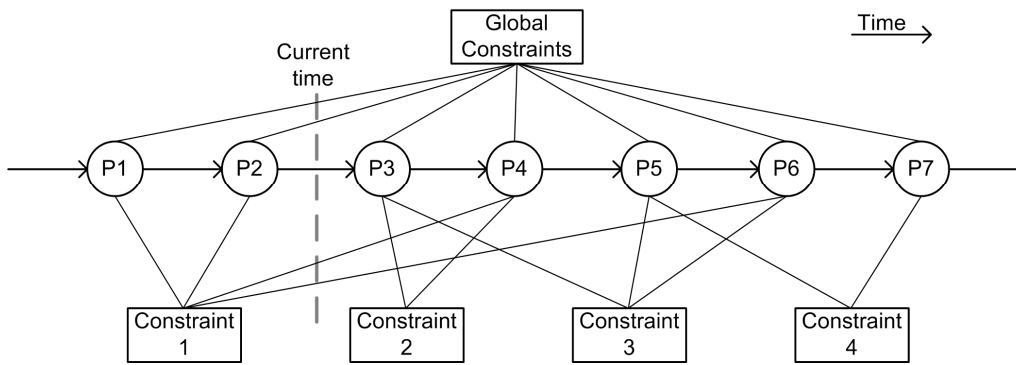


Figure 1: Sequential plan showing constraints

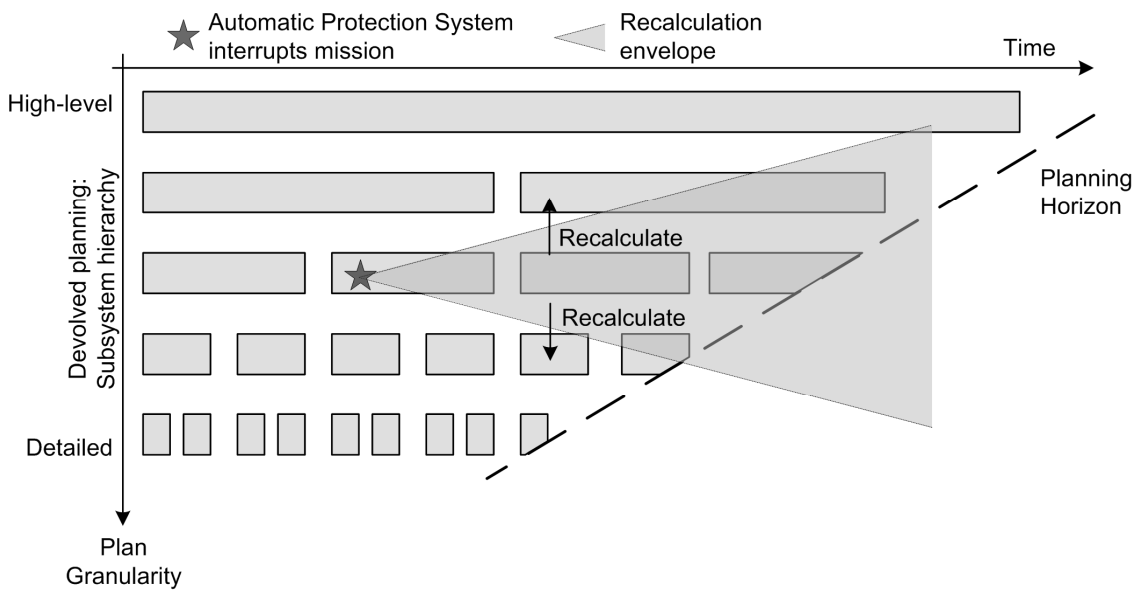


Figure 2: Hierarchical plan with recalculation envelope

Following an APS response replanning should be performed on the affected plan to find a new plan which can achieve the goal(s) of the original plan under the new or modified constraints placed on it. If a new plan can be found it should replace the current plan at this level which will result in the sub-plans of this plan also requiring replacement using the normal hierarchical planning mechanism. When replanning is performed on the affected plan it may not be possible to formulate a suitable new plan that will achieve the original plan's goal(s) under the new set of constraints resulting from an APS response. This means it is not possible to isolate the effects of the APS response to the affected plan and its sub-plans, as above, but replanning must be performed higher up the hierarchy.

Clearly, this recursive upward propagation can eventually reach the top of the hierarchy, and a full replan will be necessary. In some circumstances a significant amount of replanning effort may be expended before reaching the top of the hierarchy, more effort than a straightforward full replan from the highest level upon detecting the APS response. However, deciding when and whether to abandon repair and attempt a full re-plan depends on a number of factors. An APS response will affect one or more platform assets, thus reducing the search space of potential replan solutions (in HTN terms, fewer 'primitives', and hence fewer 'methods' are available). Therefore any analytical comparison of the search space dimensions before and after an APS response is application-dependent.

An APS response can result in constraints being added, modified or removed or in the violation of the existing constraints. As described above, plans will be affected by this and in some cases replanning will have to take place. However, there will also be occasions where the planner will be unable to find a suitable new plan (the current mission goal is not satisfiable under the prevailing conditions). If this is the case then to find a suitable new plan, there are four actions which can be taken, which are:

1. No action at all, resulting in loss and possible destruction of the platform.
2. Reverting to a contingency goal (self-destruct, back to base, etc).
3. Drop low priority goals to enable higher priority ones to be met. Requires a mechanism to be in place to attach priorities to goals.
4. Relinquishing autonomy to a human operator.

It may be possible to exploit the predictability of APS responses (but not pre-empting the triggering circumstances) to mitigate the cost of plan repair during and after the APS response. If the planner has a model of a particular APS response, then a number of strategies could be employed.

- Situated actions - pre-computed responses to predefined cues.
- Conformant planning - creates a plan which will work for a range of known contingencies.
- Contingency planning – generates plans with contingent branches.
- Condition checking – checks pre- or post-conditions (or both) of each action for applicability.

World Models

Autonomous systems require some method of storing and maintaining situational awareness to support planning and decision making. This is often achieved by

maintaining a world model, which is a collection of information that is held to describe an autonomous system's status and that of its environment.

In the world model a model of the autonomous system itself (self-model) can exist. This is an estimate of the current state of the autonomous system. By monitoring this, it would be possible to spot changes in the system's own state that might indicate an APS response. This could be evaluated using a new type of constraint called a 'normal behaviour' constraint. This is described further in the next section.

Mechanisms called Truth Maintenance Systems (TMS, sometimes also referred to as belief maintenance systems) try to keep the 'truths' or 'beliefs' of the system consistent by resolving conflicts. They keep information about the dependencies of one piece of information on another enabling more efficient modification and repair of the model and allowing them to infer other pieces of information from those gathered. The three most widely used Truth Maintenance Systems are:

- Justification-based TMS (JTMS)
- Assumption-based TMS (ATMS)
- Logic-based TMS (LTMS)

These are all slightly different in how they construct beliefs with JTMS and LTMS supporting only one context (combinations of assumptions), while ATMS will support a number of different contexts. JTMS does not support negation in its justifications, whereas LTMS does.

A TMS is combined with a problem solver to comprise a reasoning system. They create an iterative loop whereby the problem solver (with its domain knowledge) infers possible beliefs, which it tries to justify to the TMS. The TMS must decide if these are valid (based on the currently held beliefs) and perform maintenance on the new belief set which it

gives to the problem solver to restart the process with inference.

The TMS aids the problem solver by trying to converge beliefs towards those which satisfy goals. This implies that the TMS has a means for measuring how far the current beliefs are from those of the goal state, in other words a metric function of this distance to desired beliefs. Notionally, the metric function takes several variables, and large negative gradients in the metric function would indicate significant changes in the world state away from the desired goal state. Assuming for the moment that the metric can be stated analytically as a function of several variables, then anomalous (or anomalously large) changes away from the goal state can be measured by comparing the value of a suitable partial derivative – expressing dependency on a subset of the metric variables – with the instantaneous value of some heuristic threshold function:

$$\frac{\partial^p M(\mathbf{x})}{\partial x_1 \partial x_2 \dots \partial x_p} \leq T(x_1, x_2, \dots, x_p) < 0 \quad (1)$$

$$\mathbf{x} \in \mathbf{R}^n, 1 \leq p \leq n$$

In practice, the metric function might not be (wholly) analytic, and equation (1) would be replaced with a suitable rule set.

The problem solver could identify some of these changes as being large enough to be an anomaly and could, from a pre-computed set of anomalies, infer a particular APS response has occurred.

Constraints

Planners typically must consider constraints when formulating a plan. Some major types of constraint are; resource constraints, temporal constraints, spatial constraints and situational constraints. In addition to the categories of constraints described a new type of constraint is introduced, called the normal behaviour constraint. The purpose of normal behaviour constraints is to

provide an envelope around the expected behaviour of the autonomous system. The breaking of these constraints can be imagined as the AS straying out of this envelope which acts as an indication that there is something unusual in the behaviour of the system. The constraints should be specifically designed to create an envelope which the AS will stray out of when an APS response has occurred. The APS response will not however be identified from the violation of an isolated behavioural constraint but instead as a pattern of behavioural constraints violated in combination. To make this possible a set of rules is required which will evaluate broken behavioural constraints.

Normal behaviour constraints are considered as being ‘soft’ constraints, where one of these constraints being broken will not necessarily result in the need to replan. Indeed it is unlikely that one of these constraints being broken in isolation will be of any significance however, when a number of them, specified in a APS identification rule, are broken they will effectively be treated as ‘hard’ constraints and therefore prompt the plan monitoring to replan.

Two important factors when designing constraint monitoring must be taken into consideration. These are the frequency at which the constraints are evaluated and the degree of hysteresis used when deciding if a constraint has been broken. Getting the frequency wrong can result in a waste of processing power, if it is too high, or reacting too slowly when a constraint is broken, if the frequency is too low. The result of getting the degree of hysteresis on constraints wrong are constraints which are being constantly broken in normal operation if it is too low or APSs not being detected if it is too high.

Architectures for APS detection

From the concepts reported here, three architectures for APS detection can be concluded:

1. Truth Maintenance Systems
2. Rule-based monitoring
3. Precondition checking

The first two will be explored in more detail here, precondition checking has already been discussed.

TMS Solution

Truth Maintenance Systems solve goal problems by attempting to converge 'beliefs' held in the Problem Solver to a set of beliefs which satisfy the goal. This is achieved by iteration: a belief set leads to inferences in the Problem Solver, supported by justifications (see Figure 3). The TMS uses these justifications to update, or maintain, beliefs in data, which in turn are used by the Problem Solver to make new inferences.

Belief convergence implies the need for a mechanism by which the Problem Solver



Figure 3: Functions in a TMS (modified from Figure 1 [3])

Rules Based Solution

The roles of world models, planners and monitoring when operating autonomously in the real world have been identified earlier. Figure 4 depicts these key components and their logical relationships.

The core of the architecture is a world model, which holds two types of

information. Dynamic real time information from sensor perceptions and asset self-check update the world model during operation, and fixed encyclopaedic facts about the behaviour and structure of the vehicle provided in advance by subject matter experts.

can measure whether its current set of beliefs satisfy the goal. Clearly, the mechanism can be a two-valued function: either the goal is satisfied or it is not. On the other hand, the mechanism can be viewed as a metric function (which may be analytical in several variables including time) on belief space. The metric measures the 'distance' between the current belief set and the goal. This concept can be exploited for APS detection (and for anomalous behaviour in general) by evaluating how the belief metric values change with each pass through the belief / inference / justification / maintenance / belief-update loop.

Large negative gradients in the metric function – indicating that beliefs are now sharply diverging from the desired goal – can be used to infer significant changes in the world state. Because justifications are explicitly recorded in the Problem Solver, and because the re evaluation of beliefs is explicit in the TMS, it is notionally possible for the reasoner to infer those elements in its 'self model' which are now behaving abnormally

The encyclopaedic information encompasses the structure of the 'self-

model’, and crucially the ‘normal behavioural model’ of the system (both discussed before)

The planner component reasons over the information in the world model to produce plans. Plan execution is monitored; both classes of monitoring identified above occur concurrently and independently.

Monitoring of reality versus the expectation in the plan does not draw on the behavioural model of the system. Only the immediate comparison between live

context (reality) and the plan (expectation) matters to this kind of monitoring.

A rule-based approach is suggested for implementing the normal behaviour monitoring. Rule sets offer a number of advantages in this regard. Firstly, rules can be encoded from subject-matter expert knowledge; secondly, rules are explicit statements of fact about system behaviour; and finally, efficient rule manipulation algorithms exist, such as implementations of the Rete algorithm for many to many pattern matching.

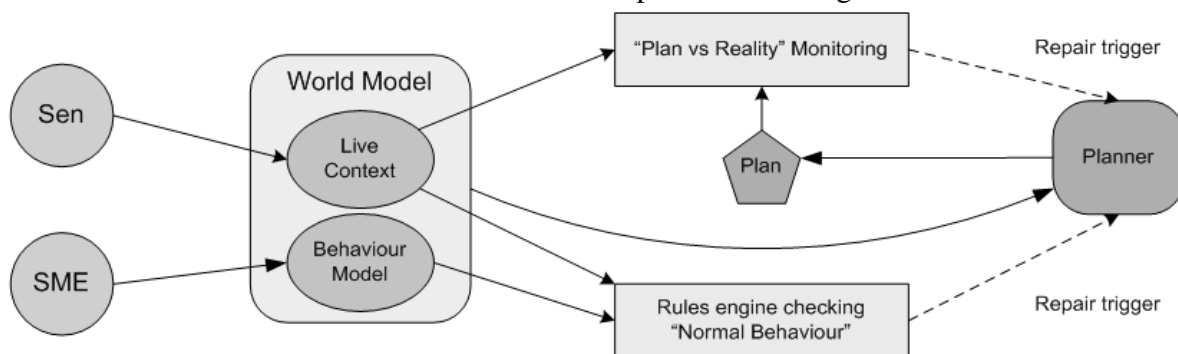


Figure 4: Simultaneous model and plan monitoring (Sen = Information from Sensors, SME = Information from Subject Matter Experts)

Relationship between APS and Autonomy

There are varying degrees of autonomy which a system could have from completely controlled by a human (i.e. not autonomous) to being fully independent of an operator (i.e. fully autonomous). The level of autonomy can be categorised in terms of a balance of authority between the system and the operator(s). One categorisation of autonomy is the NATO Pilot Authorisation and Control of Tasks (PACT) framework [4] which has six categories.

Automatic protection systems effectively reduce the autonomy of an autonomous system by removing their autonomy in certain situations. When an APS response occurs the APS imposes what actions to take, based on its predefined response. The AS must try to identify when an APS has occurred and deal with the changes that have occurred as a result of the APS but is

usually not allowed to interfere with the APS response.

An autonomous system has knowledge of the current state of the world whereas an APS is only interested in identifying behaviour in a fixed set of measurements which triggers it into action. This means that an APS is ‘blindly’ reacting to local events and will therefore not know the effects of its actions on the global system. It also may have no knowledge of the actions of other APSs elsewhere in the system. A situation could be envisaged where APS behaviour or that of more than one APS could make sense to the individual APS but to the autonomous system could be identified as a risk to safety or preservation of the AV (the very thing the APS is trying to prevent). For example a UAV with two jet engines has an APS on each which, in the event of a flame-out will attempt a relight of the engine. However if both engines were to

flame-out at the same time both of the APSs would be triggered simultaneously. If APSs on both engines attempt an automatic relight simultaneously this could lead to ignition problems due to a limited power source available for ignition if other key systems are also competing for power at this time.

If the protection systems are put in place so that they must run unhindered, then the autonomous system must not react until they have finished. However an alternative is to allow the autonomous system to take the decision to suppress the actions of an APS or at least highlight the problem to the operator so they can allow the suppression (however for rapid APS responses there may be no time for this).

Allowing the suppression of APSs by autonomous systems may cause problems in regard to certification. If certification demands that these APSs are in place they are likely to also demand that they are allowed to perform their actions without hindrance. The problem is that if suppression is to be permitted then regulators will have to be convinced that the autonomous system can be trusted to make the correct decisions about when to suppress APSs without causing safety issues. However the counter argument could be made that the ignorance of APSs to the bigger picture can itself be a safety issue and the autonomous system can prevent this. Suppression will result in a higher level of autonomy when APS responses are occurring.

Conclusions

The importance of detecting and identifying automatic protection responses, especially whilst the responses are in progress, has been identified. A number of strategies for plan repair and behavioural modification (action substitution, repair by propagation, situated actions) have been proposed as being applicable once APS identification has been made. A number of

concepts have been introduced in this paper, each indicating strategies for APS response detection and identification.

Future Work

The notion of a ‘normal behaviour envelope’ has been identified as a possible mechanism for APS response detection. This immediately raises the question of what constitutes ‘normal’ behaviour for an autonomous vehicle. Further study is required to characterise the normal behaviour of an autonomous vehicle.

This paper has considered theoretical strategies for managing APS responses; the impact on planners and world models has been discussed in some detail. A short prototyping activity is recommended to evaluate how such strategies might work in practice.

References

- [1] A Dunlop, *Investigation into the Interaction Between Automatic and Autonomous Systems*, SEAS DTC IF061 Final Report IF061/roke/001 (February 2009)
- [2] A Lucas and D Shepherdson, *Architecture for Distributed Power Management for Autonomous Unmanned Vehicles*, PPEM008 in Proceedings of the 2nd SEAS DTC Annual Technical Conference, Edinburgh. (July 2007)
- [3] J. de Kleer, *An Assumption-based TMS*, Artificial Intelligence, Vol. 28, No. 2, March 1986 (pp. 127–162)
- [4] A. Schulte (ed.), *Artificial Cognition and Co-Operative Automation, chapter 5 in Uninhabited Military Vehicles (UMVs): Human Factors Issues in Augmenting the Force*, RTO-TR-HFM-78, NATO Research and Technology Organisation. (July 2007)

Acknowledgements

The work reported in this paper was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence.