

A first approach to the closed-form specification and analysis of an autonomic control system

Simon Dobson, Eoin Bailey, Stephen Knox, Ross Shannon and Aaron Quigley
Systems Research Group, School of Computer Science and Informatics,
UCD Dublin, IE
simon.dobson@ucd.ie

Abstract

Control systems must increasingly be designed to involve collections of hardware and software components, both of which may evolve over the lifetime of the system, and which are expected to provide self-managing, adaptive, autonomic behaviour. Understanding the behaviour such a system will exhibit under any specific conditions is a significant design challenge. We present a model derived from approaches to modelling dynamical systems in which the adaptive behaviour of an autonomic system may be described and analysed as a whole. We explain our ideas with reference to a hybrid hardware/software system, and argue that it generalises to other classes of autonomic systems.

1. Introduction

Software architectures involving control systems have long been used to abstract away some of the complexity of managing a set of interconnected software components. However, the complexity of the control systems themselves has recently been approaching the complexity of the systems they manage, and the costs of developing and maintaining these systems has risen accordingly.

In recognition of this growing problem, a new branch of software engineering was created which has come to be known as autonomic software. Autonomic software systems are designed so that they can make modifications to their own execution, adapting to changes in situation and learning from the outcomes of those decisions. Eventually, this will allow the development of autonomic systems which are entirely self-managed and require a minimum of human interaction.

Autonomic software systems will often take a multitude of contextual information into account when making decisions. This information is gleaned from environmental sensors, and sensors fitted as part of the autonomic system it-

self, which is the case for the systems built into modern automobiles. The system extracts relevant sensor data from the environment, filter and distill them, and transform them into relevant situational information so that decisions can be made in context. This allows the system to adapt in response to changes internal and external to the system [6].

The provision of a control system which manages and regulates the behaviour of subordinate components can be thought of as the introduction of a distinct layers in the architecture. We suggest that software with autonomic behaviours can be built in at any level of the architecture. Rather than write entirely new systems, in which autonomic capabilities are programmed at every level, we suggest the establishment of autonomic control systems, that have the capability to manage systems that are themselves not inherently autonomic. This allows them to be applied to systems that are already deployed, and could bridge the divide between traditional static systems, and fully adaptive systems.

In this paper we explore some ideas for modelling autonomic systems *as a whole* rather than as collections of components with possibly unclear interactions. We adopt some techniques from dynamical systems theory to construct models of systems whose properties can be examined, and show how new aspects can be added (and removed) in a well-founded manner. We conjecture that this approach can be applied to improve the analysis of a wide class of autonomic systems.

The rest of this paper is organised as follows. Section 2 surveys related work in the area of software control systems, including a basic background on autonomic systems. Section 3 presents our approach for developing an autonomic control system. Section 4 concludes with some for future work.

2 Related work

2.1 Control systems

A control system is a set of devices which manages the behaviour of other devices or systems. Control systems have been used in the automotive industry [8], space exploration [18], and power plants [4] – we will focus on the first here

In 1988, Uwe Keincke discuss the need for automotive control systems [7]. Since then, control systems such as ABS (Anti-lock Braking Systems) and engine control have been implemented and are used in all modern day cars.

Lennon et al. [17] implemented a control system using fuzzy logic in order to decide when the brakes should be applied. This was created an addition to ABS. ABS measures wheel lock from the pressure on the brakes by the driver, the velocity of the vehicle and the wheel speed. Brake temperature is also an important factor, as hot brakes require different braking behaviour to cold brakes. This paper describes a learning process which takes braking patterns and temperatures into account to add to the performance of the ABS.

Patterson and Nielson [11] describe a control system for handling speed control in a diesel engine. The purpose of this is to reduce drive line oscillations (shaking caused by engine turning the driveshaft) in a car. This is done by monitoring the oscillations in the car depending on how much diesel is fed to the engine, and how this affects the speed of the wheels.

Kiencke et al. [9] describe the impact of automatic control of vehicles in recent years. They suggest that there is a trend towards multiple embedded components in a vehicle. These components would be ad hoc, allowing for changes in the structure of the network. They then identify a problem where the greater the number of separate components, the greater the chance of mis-communication, which could lead to incorrect behaviour. This is of particular importance for critical systems like drive-by-wire. It is therefore clear that a management system must be in place to control the data flow through the system, and to manage the addition of new components such as sensors. They also forecast a movement from the control of individual components in a system to a more high-level control system.

2.2 Autonomous automotives

IBM have begun research into using autonomous computing in automotives with the DySCAS (Dynamically Self-Configuring Automotive Systems) project [2]. The idea behind this project is to bring autonomous computing to automotive software in the simplest and most extensible way possible. The project is still in the early stage of research.

Scarlett [13] proposes a system that allows for the automatic discovery of devices, self-optimisation, self-diagnostics and self-healing. This will mean that the ECUs (Electronic Control Units) in each car, which in the past were used in isolation, are linked, meaning that the system as a whole is less susceptible to failure.

2.3. Control in distributed systems

Control systems are widely used in networks and distributed systems. Abdeen et al. [1] describe the use of Markov decision processes to decide on how data is routed through networks of different capabilities. They describe the situation of a PDA user watching a video while walking between networks of different band widths. The system adapts appropriately so as to keep the video streaming by changing compression levels and frame rates.

This system displays some properties of an autonomic control system – adaptation and decision making – but does not self-diagnose or self-heal.

2.4. Autonomic systems

Autonomic computing and communications endeavour to develop approaches to systems design which facilitate self-management, self-optimisation, self-protection, self-healing and other “self-*” properties. The area has received considerable academic and industrial interest.

Autonomics embraces an approach which “closes the loop” between sensors and system behaviour. Sensed data is analysed and used to affect the system’s behaviour, and the effects of these adaptations may then in turn be observed and lead to further adaptations (figure 1).

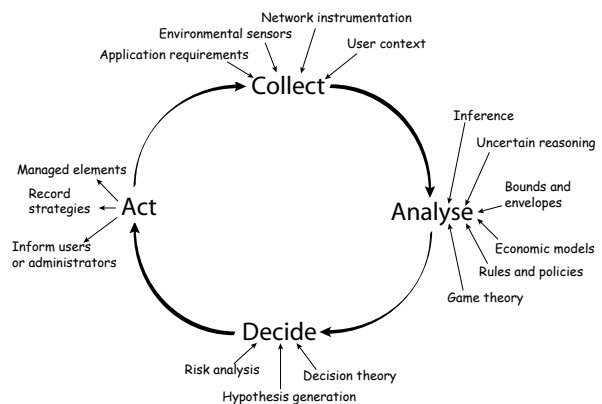


Figure 1. The autonomic control loop (from [5])

Many examples of autonomic systems can be found [14, 15], all of which have, by their nature, an element

of control, enabling them to adapt to situations to maintain service. Despite this, these autonomic systems are designed with specific goals in mind, for example resource management [16]. This specificity does not enable the easy transfer of the systems autonomicity to other domains. The autonomic control system proposed in this paper is a first approach to specifying such a system.

Sensor fusion and *system aspects* are two approaches proposed here, and believed by the authors to be part of the foundation on which a general autonomic control system will be based. Sensor fusion is the composition of disparate data sources into a comprehensive and meaningful data space. System aspects allow system administrators to monitor specific sensors for the beneficial, or indeed the detrimental, effects they have on the system as a whole.

Sensor fusion takes multiple inputs from separate sources and outputs information that should be more informative and useful than if the inputs had been taken independently, for example extracting depth information by using two images of a location taken from different viewpoints. A number of authors describe approaches for sensor fusion [3, 10]: the application of sensor fusion to the autonomic control system described in this paper will be entered into more detail in 3.1

Separating system aspects will enable system designers to quickly determine the impact of adding a specific data source will have, *via* sensor fusion. Adding a new sensor to an autonomic system should alter the adaptive space which an autonomic system uses to determine valid system adaptations. This alteration could be either a constriction of the adaptations available to the system, a broadening of the adaptations, or have a negligible consequence on the system, each of these scenarios is informative to the system designer.

If the additional data show that areas within the adaptive space that were previously believed to be safe adaptations are now unsafe, we have constricted the behaviours available to the system. As an example, consider a self-organising collection of robots. Initially the robots have wireless communication capabilities and the ability to move around a set space. Their goal is to maintain wireless connectivity while also exploring their environment. The addition of a memory capability to the collective would constrict the locations visited by ensuring that each time a robot explores a new area, no other robot visits that location again.

A broadening of the adaptive space would occur if a sensor was added that enabled the determination of the strength of the wireless signal between the robots. Previously extra co-ordination would have been required to ensure that the robots did not travel further than a set distance apart, guaranteeing a connection. With the new sensor installed the extra data will allow the robots to put further distance between themselves. The equivalent within the adaptive space

is the expanding of the safe adaptations.

If adding a sensor does not affect the system at all, then in the system's current guise that sensor is a waste of resources. This does not however rule out the sensor *ever* being useful: it may be that, when used in conjunction with other sensors not yet added, the sensor is beneficial.

When dealing with a multi-dimensional space, determining change is achieved mathematically, informing the system designer of these changes is more problematic, due to the difficulty in presenting information from a space higher than third order. Aspects facilitate the generation of informative visualisations via data "freezing": holding some of the variables constant and displaying the rest to the user. The system is decomposed into lower-order spaces, and the areas of greatest change are shown to the system designer.

2.5. Whole system understanding

A lack of understanding regarding the way in which systems interact coupled with the requirement to pre-program the manner in which systems react to change is an issue with many legacy systems. Properly engineered complex systems should consist of a number of distinct units or components, held together by a shared logic. Each of these components has a role in the overall operation of the system, and each individual component has local knowledge and a specified interface. However, since the components will have no knowledge of the system as a whole, it is difficult for an *individual* component to take *independent* action that will result in a benefit to the *entire* system.

In order for a system to make changes to its own operating procedures, it must have the ability to interpret how a change will impact other elements in the system – requiring an understanding of the entire system. This is more problematic than it first appears: allowing a system to make changes to its own operation will, by its nature, alter the model of the system. The system must therefore also be able to alter the model such that it reflects the changes made.

The potential for error when using a pre-programmed model approach is not insignificant: if the model does not capture a detail of the system that is later altered, the model of the altered system will be incomplete at best and inaccurate at worst.

2.6 Interaction between components

A well-designed system should consist of components, or layers, permitting easy alteration to the system should a component be found to require changes. An example of such a system is the OSI protocol reference stack, within which a clear definition of roles exists. In practice it is often found that there is no perfect delineation, and interactions not shown in the specification exist between the layers. It

may also be the case that interactions are programmed to intentionally exist between layers. In either case the interactions need to be identified to ensure smooth evolution.

Expanding the previous example to include multiple systems, each composed of multiple components and connected *via* network links, a large number of interactions can emerge. Systems are required to interact to perform many of their functions. At the simplest level an issue only arises when two components attempt to alter some aspect of the system in a conflicting manner. This can lead to changes being overwritten, or to oscillations between states. Neither of these outcomes is desirable.

An autonomic control system is capable of identifying these issues, and presents a solution: the extra context available to an autonomic control system coupled with the ability to learn from previous events, and the system model, provides more scope for the correct application of control.

3 Modelling approach

The terminology used above to describe autonomic behaviour in terms of dimensions and degrees of freedom is suggestive. It suggests that an autonomic system is engaged in optimising its behaviour by moving within some constrained space of possibilities. The problem, then, is to provide a mechanism by which we can capture the effects which various sensors and adaptive strategies will have on the overall, visible behaviour of the autonomic system. The approach we are currently developing borrows from techniques developed in physics for a similar purpose.

3.1 Adaptive behavioural spaces

A simple physical system such as a pendulum can nevertheless exhibit a range of interesting behaviours. The system in figure 2(a) consists of a pendulum attached to a solid block, swinging at a small angle of displacement. At any point the pendulum's state can be completely described by two variables: its displacement d from the vertical mid-line, and its velocity v . For any given initial displacement d_0 we may plot displacement against velocity on a graph. For a frictionless pendulum this results in a diagram like figure 2(b): the pendulum's velocity increases until it reaches a maximum v_{max} as it cross the mid-line ($d = 0$); it then decreases to 0 when $d = -d_0$, before reversing the motion back to d_0 . The behaviour repeats indefinitely.

Thus, figure 2(b) completely describes the pendulum system. Knowledge of one of the variables in the system allows us to infer the other: given a velocity we can identify the displacements at which it occurs; given a displacement we can determine possible pendulum velocities; and the plot completely captures all legal states of the system. A differ-

ent initial displacement d'_0 will result in a *different* circle, but the overall behaviour remains completely predictable¹.

This technique is known in physics as a *phase space plot*. The diagram has one axis for each degree of freedom of the system, either an input parameter (displacement in the pendulum case) or a dependent variable (velocity). Additional parameters add additional axes: we might, for example, treat the initial displacement d_0 as a parameter, leading to a three-dimensional surface describing the behaviour of the system for each value of d_0 (which would in fact be a cone, at least if we restrict ourselves to small displacements).

The important point is that values of the parameters of the pendulum system are strictly constrained to lie on the surface described by the phase space plot. In fact there is even more structure: at any time t , the state of the system at $t + \delta t$ is constrained by the dynamics of the system – the pendulum cannot suddenly reverse in the middle of a swing. This additional structure forms a vector space, with the vector at each point on the surface mapping that point to another point also on the curve.

How does this relate to autonomies? An autonomic system is one whose behaviour is expected to vary in response to changing environmental factors, so as to optimise in whatever sense its own service delivery. This does not of course leave the system free to make *any* adaptation, since its behaviour must remain within an acceptable design “envelope”, defined externally to the system itself [6]. The more freedom we allow within it, the more possible adaptations we provide. It follows that a system that is very highly constrained will have few opportunities for adaptation than one with fewer constraints, and that a system that is sensitive to more environmental context and has more independent or semi-independent components to its behaviour will have more “space” in which to adapt than one with fewer such parameters.

Hopefully the analogy is clear. The autonomic system forms a *behavioural space* consisting of one dimension for each input parameter (direct or contextual) and another dimension for each aspect of its observable behaviour. The relationships between these dimensions – the possible behaviours that should be visible under each combination of inputs and context – form a surface within the multi-dimensional space². The autonomic system is constrained to “live” on this surface, with any adaptations it makes taking it to another point on the surface. Any point *not* on the surface is “illegal” and should never be observed.

¹For a pendulum with friction, the pendulum returns to a displacement $d = d_0 - \delta d$, and so eventually decays to the pendulum sitting still on the mid-line. The plot would then describe a spiral to the origin.

²Actually it may be a more complex structure rather than purely a surface, but that does not affect the discussion at this level.

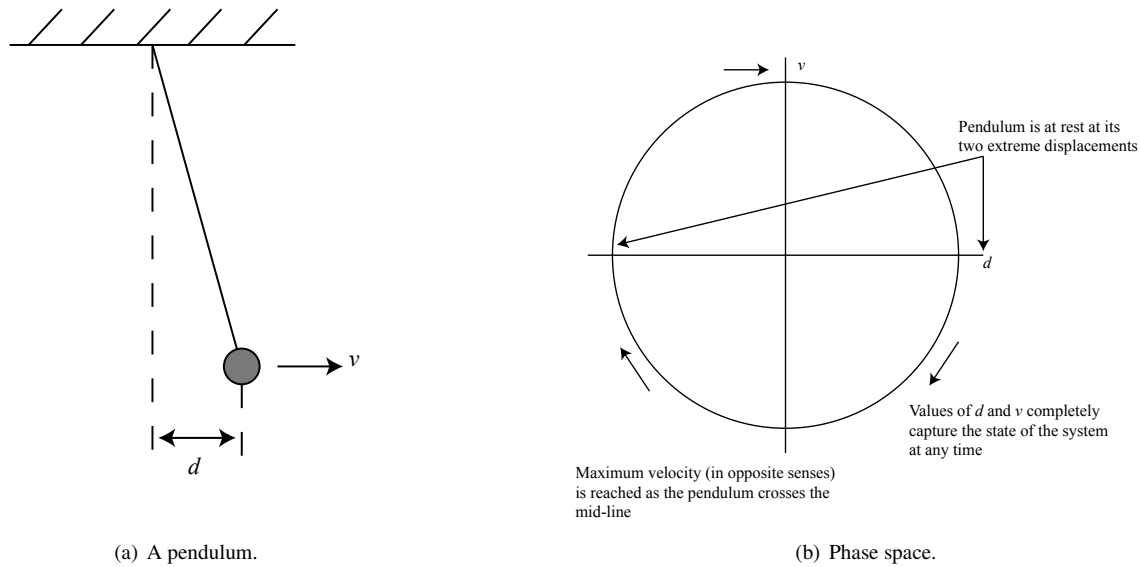


Figure 2. A simple dynamical system

3.2 Example: a braking system

If the above sounds a little abstract, let us consider a more concrete example. As part of a project in automotive software engineering, we are considering the modelling of a vehicle braking system augmented with additional sensors, with a view to providing a framework within which to study the composition and impact of various design options. In this section we will outline a model of the adaptive behavioural space of a vehicle's braking system. The model we use is only a toy, and should not be taken as a model of a real vehicle's behaviour or control: our intention is purely to illustrate the application of a whole-system adaptive behavioural space to a system involving hardware and evolving software components.

Our control scenario is easily understood (figure 3). A vehicle moving along a road encounters an obstruction, and must stop before colliding with it.

The parameters for the braking system are straightforward. The vehicle moves with initial velocity v_0 at an initial displacement d_0 from the obstruction at displacement d_{obs} . The brakes of the vehicle, when applied, exert a force that retards its velocity. If we assume that the mass of the vehicle is fixed, we may regard this force as a change $-b$ in velocity, which is proportional to the degree $p \in [0 \dots 1]$ to which the brake pedal is pressed.

The visible behaviour of the system may be described simply the vehicle's velocity v and displacement d at any time. The desired behavioural envelope of the system is equally simple: we want v to fall to zero before $d = d_{obs}$.

What is the behavioural space of this system? To stop

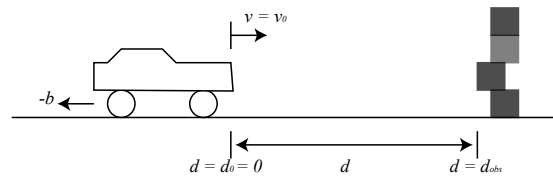
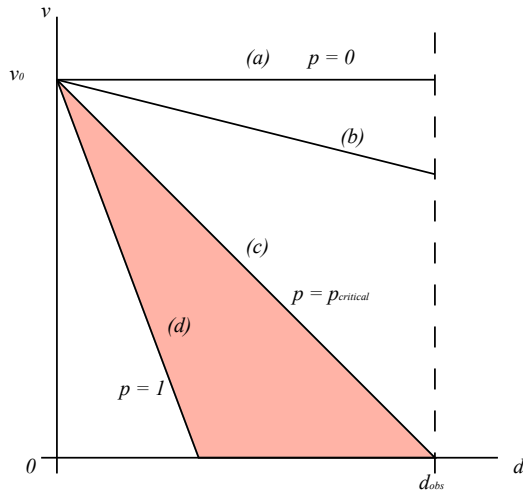


Figure 3. A combined hardware/software braking system.

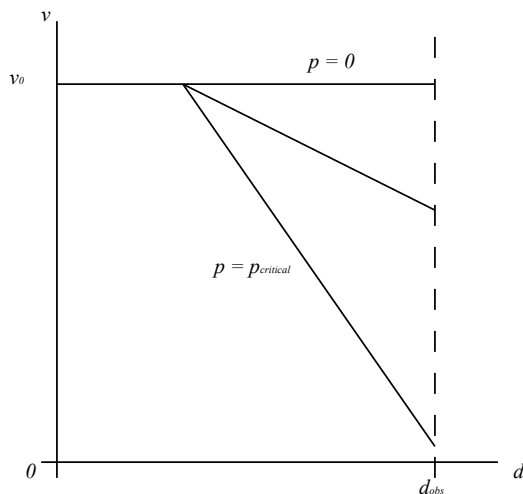
safely – to remain within the behavioural envelope – a driver must apply a braking force sufficient to stop the car in the time remaining before hitting the obstruction. The possibilities are shown in figure 4(a). If the driver applies no brakes ($p = 0$, line (a)), or applies them too gently (line (b)) he impacts with the obstruction. There is a critical braking pressure $p_{critical}$ at which he stops *at* the obstruction (line (c)). Braking more strongly than $p_{critical}$ stops the car before the wall, within the behavioural envelope (shaded area). Applying the brakes fully (line (d)) stops the car in the shortest physically possible distance.

Of course the driver may brake later, figure 4(b). At a certain point, no amount of available braking force will stop the car before impact.

To this basic model we may now add additional factors.



(a) The vehicle's behaviour under different brake pressures.



(b) The effects of late braking.

Figure 4. Manual braking.

A distance sensor One useful optional accessory on high-end cars has been a distance sensor, a forward-looking proximity sensor that, if it detects an obstruction within a certain threshold distance, applies the brakes automatically without driver intervention. We may model this using the graph in figure 5(a): when the distance to the obstruction reaches the last point at which braking is possible, the braking system applies the brakes as hard as possible to stop the vehicle. This is a simple example of autonomic control: the system acts without user intervention to keep its behaviour within its envelope, facilitated by a new sensor.

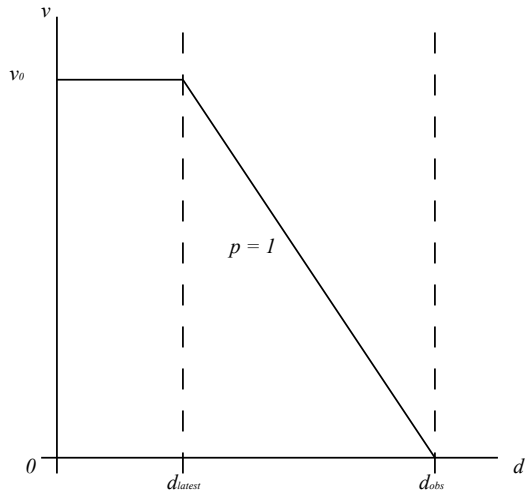
Since emergency stops are likely to be uncomfortable for passengers, we might modify the behaviour in figure 5(a) to apply more progressive braking (figure 5(b)): as the distance gets more critical, the vehicle brakes harder.

Of course, the real world is unlikely to work like this: a driver is more likely to observe an obstruction but brake too gently than to ignore the obstruction entirely. This allows us to model adaptive control. The distance sensor provides us with a model of the “edge” behaviour needed to stop the vehicle safely. If the driver brakes harder than this – so as to stay within the behavioural envelope – the vehicle need perform no action; if however the driver brakes too gently, the vehicle may apply additional braking force. Having added a distance sensor to the car, we can therefore use its braking profile as a lower bound on the amount of braking needed in any particular circumstances. In terms of the behavioural space, adding the distance sensor “clips” the possible states of the system combining figure 4(b) with figure 5(b) – something which is hard to display graphically but easy to describe mathematically.

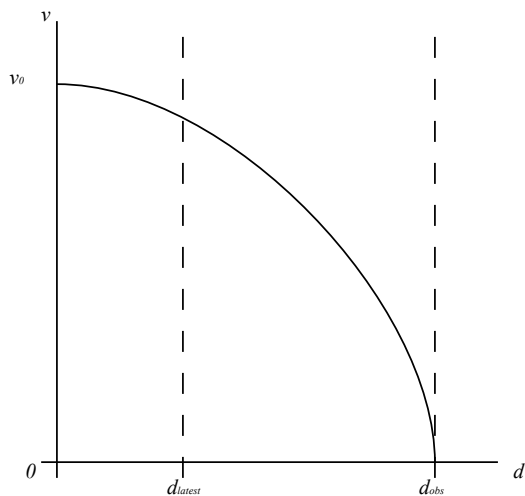
We therefore have a model of the impact that a distance sensor should have on the behaviour of the complete system. We also introduce the potential for autonomous braking in a way that is clearly related to other factors within the system, and in a way that allows us to predict the overall adaptive behaviour *a priori*.

More realistic brake behaviour An automotive engineer would observe that real brakes are not the linear devices we have suggested. As the brakes are applied, they retard the vehicle and dissipate its kinetic energy as heat. If the brakes heat up too much they either lock fully on or fail. For a given braking pressure, we therefore have a braking profile that changes over time (figure 6). This function again amends the other curves: if we brake at a certain distance and pressure, we need to ensure that the force we exert is sufficient to stop in time given the constraints of brake heating.

Additional sensing Additional sensors can now be added in a principled way, and their impact assessed:



(a) Last-moment braking.



(b) A more progressive braking regime.

Figure 5. An autonomous distance sensor.

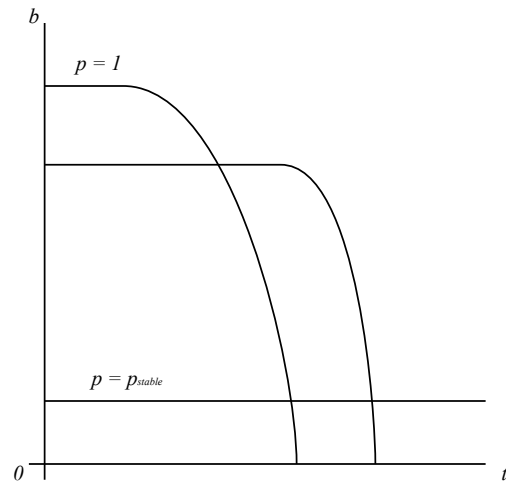


Figure 6. The effect of brake heating on braking velocity.

- A road surface sensor could characterise the road and apply less force on rough surfaces
- An external temperature sensor might decrease braking force, and decrease the emphasis given to brake heating, in cold conditions when roads are likely to be icy
- A inclinometer could add an extra dimension of road incline to the behavioural space, and brake harder on steeper inclines

The point of this is not to suggest new vehicle designs, but rather to illustrate the point that additional sensors can be characterised by the effect they have on the adaptive behavioural space, in conjunction with the characteristic curves of other sensors and actuators.

3.3. Analysis and inference

What impact does this model have for design and analysis? From a design perspective, we can describe the desired behaviour of the system in terms of its visible characteristics. Adding new sensors can (and indeed should) change the responses of the system, as long as the result remains within the desired envelope.

We have spoken so far about the use of sensors to change adaptive behaviour, but the closed-form nature of the model allows us to perform the reverse analysis too. If we apply

a certain braking force under certain conditions, we expect the vehicle to behave in a particular way: if it does not do so, we can make inferences about the effect of external conditions, and hypothesise their nature. An example of this occurs when driving in Australia, where (because of the large temperature variations the surface is subjected to) roads that appear to be dry and clear can actually become slippery with very little water on them.

If we brake with a certain force and do not decelerate as we expect, we might next time apply *more* force to compensate. This allows the adaptive behaviour to adapt itself over time in accordance with sensed conditions as to its own actual versus predicted behaviour – the essence of autonomic control. This has the effect of distorting the adaptive behavioural space gradually as the vehicle learns new parameters to apply. There is a wealth of literature on how to perform such learning; our model gives a direct geometric meaning to such changes, and allows them to be studied in a principled way that will ensure the system remains within its design envelope.

3.4 Generalisation

If these ideas applied only to sensing and physical systems, they would be of limited utility in autonomic computing. However, this does not appear to be the case. Other applications appear to be amenable to a similar modelling and analysis approach.

One example we are interested in is the notion of routing and network management with cross-layer optimisation [12]. A router might be regarded as having a behavioural space controlled by its bandwidth, the congestion on the network, the queue of packets and so forth. In response to increasing congestion, a simple router has only a limited range of options. If however we add a cross-layer “sensor” able to (for example) determine that, of the two video streams being transmitted, one has priority over the other, the router might preferentially drop packets from the less important stream. Alternatively, it may be able to proactively drop the resolution of the less important stream to create more bandwidth.

Each such strategy manifests itself as a change in the dimensionality of the adaptive behavioural space, coupled with a change in the visible responses of the system. One might argue that users are only interested in best-effort delivery of streamed video, in which case the resolution of the streams – while available to the router internally as an adaptation parameter – does not form part of the constraints on the behavioural envelope. Indeed, this almost suffices to give a formal meaning to the notion of best-effort in streaming.

The overarching point is that behaviour and adaptation do not occur in isolation, but rather appear as a constrained

system of responses to changing conditions. There is a serious design question about the complexity of the intertwining of the responses, but at an analytical level we can generate and analyse closed-form behaviours for systems of real-world complexity.

4. Conclusion

Autonomic systems are intended to demonstrate adaptive behaviour, optimising their performance without straying outside an acceptable behavioural envelope. In this paper we have presented an initial approach to describing and proving such properties for hybrid control systems, applying ideas derived from the modelling of dynamical systems to creating and analysing a system with considerable potential for software control. The various degrees of freedom of the system are constrained, with new dimensions both influencing and being influenced by other dimensions within the model. We believe that this approach, whilst still at an early stage of development, has the potential to be applied to a range of autonomic systems.

There are of course major issues remaining. Perhaps the most pressing is to determine whether *all* autonomic systems, or all components within a system, can be described in this way, and whether mixing discrete and continuous variables causes problems. We conjecture that the approach is in fact fully general, but this remains to be demonstrated.

A model may be descriptive without being useful. It is not immediately obvious how a model such as that described translates into software. We have a strong interest in programming models for autonomic systems, and especially in translating high-level, semantically well-founded models into applications in as natural a way as possible. This will be an active area of future research.

Acknowledgements. This work is partially supported by Science Foundation Ireland under grant numbers 03/CE2/I303-1, “LERO: the Irish Software Engineering Research Centre” and 04/RPI/1544, “Secure and Predictable Pervasive Computing.”

References

- [1] M. Abdeen and M. Woodside. Seeking optimal policies for adaptive distributed computer systems with multiple controls. In *Third International Conference on Parallel and Distributed Computing, Applications and Technologies*, September 2002.
- [2] R. Anthony, A. Rettberg, O. Redell, T. N. Quereshi, M. Torngren, C. Ekelin, and G. de Boer. Dynamically reconfigurable automotive control systems. In *Advanced Automotive Electronics*, January 2007.

- [3] L. Coyle, S. Neely, P. Nixon, and A. Quigley. Sensor aggregation and integration in healthcare location based services. In *First Workshop on Location Based Services for Health Care (Locare'06)*, 2006.
- [4] R. Dimeo and K. Lee. Boiler-turbine control system design using a genetic algorithm. *IEEE Transaction on Energy Conservation*, 10(4):759–759, December 1995.
- [5] S. Dobson, S. Denazis, A. Fernández, D. Gaïti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli. A survey of autonomic communications. *ACM Transactions on Autonomous and Adaptive Systems*, 1(2):223–259, December 2006.
- [6] S. Dobson and P. Nixon. More principled design of pervasive computing systems. In R. Bastide and J. Roth, editors, *Human computer interaction and interactive systems*, volume 3425 of *LNCS*. Springer Verlag, 2004.
- [7] U. Kiencke. A view of automotive control systems. *Control Systems Magazine, IEEE*, 8:11–19, 1988.
- [8] U. Kiencke and L. Nielsen. *Automotive Control Systems, For Engine, Driveline, and Vehicle*. Springer Verlag, 2nd edition, 2005.
- [9] U. Kiencke, L. Nielsen, R. Sutton, K. Schilling, M. Papa-georgiou, and H. Asama. The impact of automatic control on recent developments in transportation and vehicle systems. *Annual Reviews of Control*, 30(1):81–89, 2006.
- [10] A. Kofod-Petersen and A. Aamodt. Contextualised ambient intelligence through case-based reasoning. In *8th European Conference on Case-Based Reasoning*, pages 211–225, 2006.
- [11] M. Pettersson and L. Nielsen. Diesel engine speed control with handling of driveline resonances. *Control Engineering Practice*, 11(10):319–328, July 2003.
- [12] M. Razzaque, P. Nixon, and S. Dobson. Demonstrating the feasibility of an autonomic communications-targeted cross-layer architecture. In *Proceedings of the 14th International Conference on Advanced Computing and Communications*, 2006.
- [13] K. Scarlett. Towards autonomous automotives. <http://www-128.ibm.com/developerworks/autonomic/library/ac-autos.html>, December 2006.
- [14] K. Stoy. Controlling self-reconfiguration using cellular automata and gradients. In *Proceedings of the 8th International Conference on Intelligent Autonomous Systems (IAS-8)*, pages 693–702, 2004.
- [15] G. Tesauro, D. Chess, W. Walsh, R. Das, A. Segal, I. Whalley, J. Kephart, and S. White. A multi-agent systems approach to autonomic computing. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 464–471, 2004.
- [16] E. Thereska, M. Abd-El-Malek, J. J. Wylie, D. Narayanan, and G. R. Granger. Informed data distribution selection in a self-predicting storage system. In *Proceedings of the 3rd International Conference on Autonomic Computing*, pages 187–198, 2006.
- [17] L. William. Intelligent control for brake systems. *IEEE Transactions on Control Systems Technology*, 7(2):188–202, March 1999.
- [18] R. T. Wood. Autonomous control capabilities for space reactor power systems. In *AIP Conference Proceedings*, volume 699, page 631, 2004.