Designing Behaviour Based Systems Using the Space-time Distance Principle.

Christopher Harper, Alan Winfield
e-mail: cjharper@avian-technologies.co.uk
alan.winfield@uwe.ac.uk

Intelligent Autonomous Systems Engineering Laboratory, Faculty of Engineering,
University of the West of England, Coldharbour Lane, Frenchay, Bristol BS16 1QY

Abstract.

This paper presents a methodology for constructing behaviour-based systems for use in safety critical applications. The specific requirements of constructing a rigorous safety argument for a system prior to its introduction into service tend to discourage design methods based on evolutionary and reinforcement learning techniques. A set of general design principles is proposed for construction of subsumption architecture systems, which can be applied deductively to identify behaviour modules and their organisation within the architecture. In addition, a general procedure for design of the behaviour module functions is also discussed. This design method is based on the use of Lyapunov stability techniques. An experiment is discussed, still in progress, which applies the methodology to a small behaviour-based system for an Inverted Pendulum apparatus.

1 Motivation for the Work.

This paper is a review of a PhD project at the University of the West of England (UWE) Intelligent Autonomous Systems Lab, investigating design methods for behaviour-based autonomous agents that will allow their practical use in *safety critical systems*. The general philosophy underlying the behaviour-based approach has the potential to be very useful in safety critical systems design.

Most currently proposed design methodologies for behaviour-based systems advocate a machine learning or evolutionary approach. Typically, an agent such as a mobile robot is built with few (if any) functions or rules governing its behaviour, and is then subjected to a series of experiments, in which the control laws for behaviour are built up through its built-in learning mechanism. Eventually the robot acquires the behaviour patterns needed to perform its required tasks. This is essentially a process of *inductive reasoning*; the system is built by developing general rules/laws of behaviour based on a specific set of examples.

This approach is in marked contrast to traditional engineering methodologies, in which a system design is constructed by following a set of design rules or guidelines. The design is constructed (more or less) by a process of *deductive reasoning*, applying the general design rules to a set of initial system requirements that define the specific nature of the system. The system is then tested against its requirements to determine the extent to which the process was successful.

Although evolutionary methodologies are a powerful means for producing design solutions for autonomous agents, there are problems with the philosophy of the approach that may prevent their use in safety critical systems. These systems require that a *safety case* be prepared prior to their use in service. A safety case provides a rationale to explain why the system should be considered acceptably safe. For systems that have the potential to cause very serious accidents, it is necessary to design and build the system, and develop its safety case, with the highest possible level of rigour. This ensures that safety regulators, owners, and users/customers of the system can have confidence that the system presents an acceptable risk.

Proc. TIMR 01 - Towards Intelligent Mobile Robots 2001.
Technical Report Series, Department of Computer Science, Manchester University, ISSN 1361 - 6161
Report number UMCS-01-4-1. http://www.cs.man.ac.uk/csonly/cstechrep/titles01.html

In general, scientific rigour is achieved through deductive reasoning. Popper [Ref.1] has demonstrated that scientific theories developed by means of inductive reasoning contain flaws where proof of the theory depends on proof of the inductive system of reasoning, which cannot be done without introducing logical inconsistencies. However, a deductive system reasoning, in which theories expressed in general terms (without reference to any specific instantiation of the problem) are subjected to falsification tests, are not subject to this logical flaw. With respect to the problem of designing systems, a scientifically rigorous methodology must develop solutions by deducing a design solution from general principles rather than inducing a solution from training examples. This contradicts the general philosophy of reinforcement learning techniques, and may require considerable discipline on the part of evolutionary development techniques if the same issue is to be avoided.

If behaviour-based autonomous agents are ever to be used in safety critical industrial applications (and for robotic systems this may constitute the majority of foreseeable applications) then system design methodologies must be developed that incorporate a more deductive approach than many contemporary proposals, e.g. Cliff et al [Ref.2]. The development of such a methodology is the motivation of this work.

2 Selection of System Architecture.

A number of architecture types have been reviewed for their suitability to safety critical systems applications. The original survey was based on references from the early 1990s [Refs.3, 4, 5]. However, Arkin [Ref.6] has recently published a comprehensive survey of behaviour-based research from the mid-1980's to the near-present (1998).

The candidate architecture types were:

1. Brooks' Subsumption Architecture [Ref.7]
2. Connell's Subsumption Architecture [Ref.3]
3. Action-selection Networks [Ref.8]
4. Rosenblatt and Payton Architecture [Ref.9]
5. AuRA [Ref.10]
6. Biologically-inspired Neural Networks [Ref.11]
7. Dorigo and Colombetti Architecture (BAT Methodology) [Ref.12]

These architectures were reviewed with respect to a list of the architectural attributes that were considered to be desirable for safety critical applications. These attributes are listed below:

(a) Modularity: Faults occurring in one module are less likely to have direct cascade consequences on the operation of other modules (although it is possible for modules to interact through the environment).

(b) Determinism: Although a system interacting with a non-deterministic environment must inevitably display non-deterministic behaviour, it is generally considered important that the transfer function of the system itself be deterministic.

(c) Reflexivity: The term "reflexivity" in this paper is used to describe the degree to which perceptions are directly coupled to actions. Highly reflexive architectures will have lower internal complexity in their design, which tends to reduce the complexity of its failure behaviour.

(d) Dependence on Inductive Techniques: Since learning or evolutionary design is undesirable for safety critical applications because of the potential weakness of inductive reasoning, architectures whose associated development methods are evolutionary are less preferable than architectures that can be designed either by hand, or by deductive design principles.

A comparison of the different types of architecture shows that Connell’s variant of Subsumption Architecture [Ref.3] (referred to by Arkin [Ref.6] as "Colony Architecture") is the best candidate. This type of behaviour-based system architecture has the greatest number of features that make it attractive for application to safety critical system problems.

The architecture uses a minimal set of behaviour module types and arbitration mechanism types. This use of a specific subset of Subsumption Architecture is analogous to the use of so-called "safe subsets" of programming languages, such as the Safety Annex (H) of Ada 95 [Ref.13]. Many of the selection criteria above also apply to conventional software technologies, and deterministic subsets of programming languages are defined for similar reasons.

3 The Space-time Distance Design Methodology.

This section proposes a design methodology, which employs a deductive approach to the construction of subsumption architecture behaviour-based systems. The overall objective is to define an approach that contains sufficient deductive philosophy to satisfy the needs of making a safety argument for a system. The principles of the methodology, and their corollaries, are listed below.

Principle 1: Any behaviour module that is completely suppressed or inhibited by another should be removed from the system architecture design.

Corollary 1: All behaviour modules in a given system architecture shall be able to gain control of the system actuators for at least a fraction of the system's operation time.

Corollary 1 raises the problem of how to determine what fraction of the system operation time should be occupied by any given behaviour module. This problem is addressed by applying a second principle, called the *Diminishing Activity Property*:

Principle 2: The Diminishing Activity Property - Behaviour modules with a longer average period between periods of activity must occupy a higher-level position in the Subsumption Architecture to which they belong.

Justification for this principle is provided by the theorem that the Diminishing Activity Property defines the condition of minimum interference between behaviour modules. Formal proof of this theorem is provided in Appendix A of this paper. If a given Subsumption Architecture is organised such that the property is satisfied, then the amount of actuator time available to each behaviour module is optimised relative to one another.

Since the Diminishing Activity Property defines the lowest degree of interference between behaviour modules, a design methodology for Subsumption Architecture should attempt to generate this property automatically as a result of applying the methodological rules. This is achieved by adhering to the Space-time Distance Principle, as explained below.

The Connell variant of Subsumption Architecture makes use only of behaviour modules with no internal representation of the state of the system. Therefore, the only way in which the Diminishing Activity Property will be established in a given Subsumption Architecture is by nature of the environmental interaction displayed by each behaviour module. In order to define rigorously the behaviour of a Behaviour Module, a number of principles should be adhered to, about the specification of behaviour:

Principle 3: System goals are defined as states or regions of state space, towards which the system state must converge.

Corollary 2: System hazards are defined as states or regions of state space, away from which the system state must diverge.

The above principle, and its corollary, are the basis for the mathematical specification and design of behaviour modules using Lyapunov Stability techniques, as described in Section 4. There is justification, both mathematical [Ref.14] and biological [Ref.15], for defining behaviour in terms of dynamical stability or equilibrium of processes. Each behaviour module in a given subsumption architecture is designed to cause the system to achieve a certain goal state (or region of states) while simultaneously avoiding certain hazard states (or regions of states).

Principle 4: Behaviour modules must have *differential transfer functions*, generating control activity only when changes in observed state occur, and generating no activity when the system is convergent on its goal states, or divergent from its hazard states.

When the motion of a system converges naturally on the goal state(s) of a given behaviour the associated behaviour module should not generate any control activity, because it is not necessary to do so. This gap in module activity allows lower-level behaviour modules to gain access to the system actuators without being suppressed or inhibited.

The differential nature of behaviour module transfer functions means that activity will only be generated when the appropriate stimulus condition appears at the system sensors. Processes in the environment, with which the behaviour module is intended to interact, generate these conditions. The interaction can be viewed as a feedback loop in which, from the system perspective, information flows from the behaviour module (via its output signals) to the environmental process and back to the module through its sensors. Since, according to Principle 4, behaviour modules only act on *changes* in state, the frequency of response to a change in environmental state will depend on the time taken for information to propagate through the flow path from the environmental process to the behaviour module and back. Given a finite speed of information flow through the loop, the duration of that flow is dependent on either on the spatial range of the process or the response time. Distant processes will take longer for the information to flow out and back, and processes with a characteristic delay will take longer to generate appropriate state changes. Since a behaviour module has no internal state, it will not be able to distinguish make the distinction that one process might be distant and another might be near but have a built-in delay; it will simply generate signals at the same frequency for both processes.

Since an environmental process at a given spatial range or temporal delay will generate module activity at a certain frequency, the spatio-temporal 'distance' of processes can be used as a design parameter that governs a module's position in its subsumption architecture. Interactions with 'distant' processes are likely to have lower activity frequencies in their respective modules, thereby giving rise to the following principle, the *Space-time Distance Principle*:

Principle 5: The Space-time Distance Principle - higher-level behaviour modules shall manage interactions with environmental features that are more distant. Alternatively, higher-level behaviour modules shall manage interactions with environmental features with longer response times Hence, subsumption architecture layers represent interactions at increasing ranges and over increasing lengths of time.

Section 4 describes the use of Lyapunov stability methods for the design of behaviour module transfer functions. This ensures the stability of system behaviour. However it is usually the case that stability can only be established or proved within closed state-space regions around the Goal State of a behaviour module function. Any excursion of the system from this neighbourhood will result in the system becoming unstable, and the affected behaviour will be erroneous or even potentially hazardous for a safety critical system.

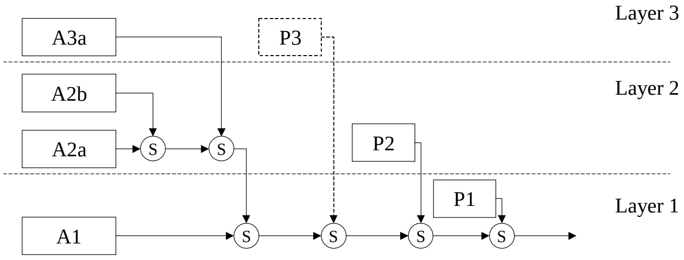
This is a potential problem for subsumption architectures, where the actions of a behaviour module at a given level can disturb the state variables used by lower-level modules. The lower-level modules cannot correct the disturbance because the higher-level module suppresses them, and hence the system leaves the region of stability (sometimes referred to as the *Viability Region* [Ref. 16] of one or more modules).

In order to provide some defence against such events, the following principles are proposed:

Principle 6: Subsumption architecture systems should be built with two internal hierarchies, *Achievement Hierarchies* and *Protection Hierarchies*. Achievement Hierarchies contain Achievement Modules that perform the mission of the system. Protection Hierarchies contain Protection Modules that guard against excursions of the system state from the stability region of their associated Achievement Module.

Principle 7: A protection module shall suppress (i.e. take higher priority than) all achievement modules of a higher level than its associated achievement module, and their associated protection modules.

Since all higher achievement modules can suppress a module at a given level, they can potentially overwhelm it. Therefore the associated protection module must take priority in the system architecture over the higher-level modules, and their protection modules, as required by Principle 7. This implies that the subsumption architecture should have a format shown below.



A*n* denotes an Achievement Module
P*n* denotes a Protection Module

Figure 1: Achievement and Protection Hierarchies within Subsumption Architecture.

Note that the highest achievement module cannot be suppressed, so it does not necessarily need to be added unless it is advantageous to the system mission to do so. The Inverted Pendulum experiment described in Section 4 has a simple architecture containing two achievement modules and one protection module (see Fig.7). There is no protection module for the top-level achievement module.

Protection modules may also be added to a system to protect against specific system hazards. In safety critical system development projects, a hazard analysis is required to identify as comprehensive a list of hazards as possible, and there is a requirement to reduce these risks to acceptable levels. One means of risk reduction could be the inclusion of protection modules for specific hazards, at the appropriate spatio-temporal distance level in the architecture.

4 Behaviour Module Design Using Lyapunov Stability.

Safety-critical applications are required to demonstrate high levels of rigour in the development of systems. If the system being developed is such that its failure could cause injury or even fatality, then it is necessary to use the best methods available to ensure that the integrity and reliability of the system design meets acceptable standards. To achieve this aim, the methodology proposed throughout this paper has been based on mathematical theorems, for example Theorem 1 in the Appendix.

The methodology in this report provides a rational basis for specifying behaviour modules and organising them into a coherent architecture. A complementary method has been developed for the detailed design of their individual transfer functions, which is consistent with the requirements Principles 3 & 4, and associated Corollary.

The behaviour module design method is based on the use of *Lyapunov Stability Theory* [Refs.17, 18], which define a set of theorems for proving that the state trajectories of a system will converge on an equilibrium state, or set of states. This has close correspondence with Principle 3 and Corollary 2, which require the properties of convergence or divergence with respect to goal and hazard states. Hence, by proving the stability of a behaviour module transfer function with respect to its specified goals, and instability with respect to its hazards, assurance is gained that the behaviour module possesses the appropriate safety properties.

Traditionally, Lyapunov Stability analysis has been difficult to apply, because the theorems require that the transfer function be of a form where changes in state are a direct function of the actuation state variables themselves. This is often difficult to achieve in a practical system. To improve the utility of the method, an extension of the basic theorems has been developed, allowing stability to be proved in terms of second derivatives of state variables rather than first derivatives. The new theorem has been named the *Second Order Lyapunov Stability Theorem*. A mathematical proof for this theorem has been developed, and will be published in forthcoming papers (this paper is concerned mainly with the Space-time Distance Principle). Use of the Second Order Lyapunov Stability Theorem is a significant enhancement to the utility of Direct Lyapunov Design method, allowing it to be used for many behaviours in which a system's position is governed by changes in actuation force. Most movement behaviours such as obstacle avoidance, navigation, etc. fall into this category.

Lyapunov stability theorems are usually applied as an analytical technique on an existing control system function. However, this has often been a problem because while the theory enables the stability of a system to be proven once a candidate Lyapunov Function is found, there is no systematic method for discovering such a function. In order to overcome this problem, a method is proposed whereby a simple Lyapunov Function is selected, and the behaviour module transfer function chosen such that the stability of the behaviour module function, defined by the above conditions, is proved. To date, this has been carried out using a three-step process:

(1) generation of a piecewise constant transfer function,

(2) application of a representative sample from each piece of the system state space to the pre-selected Lyapunov Function,

(3) iteration through a range of actuation parameter values to determine which action best satisfies the (Second Order) Lyapunov stability criteria.

This method has been called *Direct Lyapunov Design*, to reflect the fact that the stability theorems are used as a means of direct synthesis of the behaviour module, rather than an indirect means of analysis.

5 Current Experimental Work.

The first experiment using the Space-time Distance Principle, currently in progress, is an Inverted Pendulum (a.k.a. “Cart-Pole”) experiment. This is a reference bench-top experiment used by many different control system technologies, ranging from basic PID control to neuro-fuzzy techniques [Refs. 19, 20]. The experiment has been chosen to allow a comparison of controllers developed using the Direct Lyapunov Design method described in section 3.3 with existing control techniques. The basic experimental rig is similar in nature to other inverted pendulum experiments, and is illustrated below.

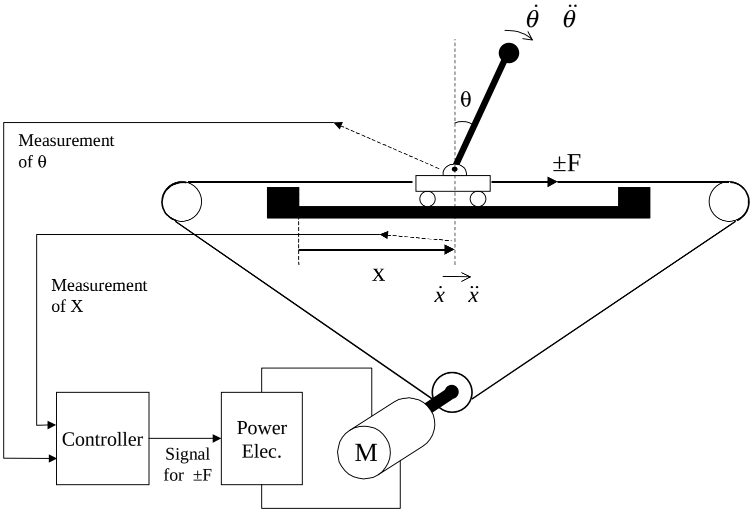


Figure 2: Inverted Pendulum Experiment.

The experiment contains a micro-controller based control system, which will be downloaded with a program implementing a simple subsumption architecture system with two behaviour layers.

In order to define a subsumption architecture controller, it is necessary to choose a *reference point* with respect to which the controller will sense its environment. Inspection of the experiment in Figure 5 shows that the pendulum pivot mounted on the cart is the obvious candidate. Both the actions applied by the controller and the parameters measured by it act through this point. Selection of controller reference points is an issue for which deductive principles have not yet been established, and is the subject of future work.

Behaviour patterns for the subsumption architecture modules are defined with respect to the reference point of the controller. It is necessary to identify the spatio-temporal distance categories that specify the layers of the architecture. Given the simplicity and restricted environment of the Inverted Pendulum, only two layers have been defined. The lowest layer is termed the Cart Frame and contains behaviours between the cart and pole. The second layer is termed the Arena Frame and contains modules that govern the interaction between the cart-pole unit and its local environment. Since cart and pole are mechanically linked, the effect of control actions will be instantaneous and their relative distance is zero. However, features of the environment outside the cart (the ends of the track, or the centre position) can be potentially at a distance from the cart, and require a finite (if short) time on which to converge. For this reason, the Arena Frame should occupy the higher position in the architecture. A few specific behaviour modules within these two layers have been selected, and the intended architecture is shown below.

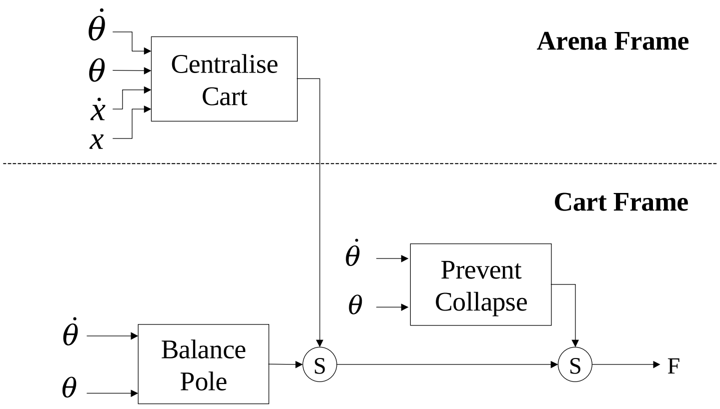


Figure 3: Inverted Pendulum (Cart-Pole) Subsumption Architecture.

The lowest priority module, called "Balance Pole", aims to balance the pole vertically at whatever arena position the cart happens to be in. An Arena Frame module called "Centralise Cart", whose goal is to move the cart to the central position of the arena, suppresses the pole-balancing module. As discussed in Section 3.4, the architecture consists of achievement and protection hierarchies, albeit with only one or two modules each. The architecture contains a protection module for the Cart Frame layer, called "Prevent Collapse", which will act to restore the motion of the pole if its motion is potentially unrecoverable and likely to lead to the pole falling flat.

The protection module is a necessary component of the subsumption architecture because the cart-pole machine is a *non-minimum-phase* mechanical system - applying a control force in a given direction causes the cart to move in one direction and the pole to rotate in the reverse sense. The relative motion of the two bodies is out of phase, hence the name of this type of system. This is a potential problem, as actions required to centralise the cart will contradict the actions required to balance the pole. Therefore, the Centralise Cart module can only be allowed to operate when its actions do not jeopardise achievement of the Balance Pole module. This is the reason for the Prevent Collapse module within the architecture.

At the time of writing, a software simulation of this system has been developed, and the Balance Pole module has been developed using the Direct Lyapunov Design procedures discussed in Section 3.3. Development of the Centralise Cart and Prevent Collapse modules is still in progress. The ultimate aim of this experiment is to demonstrate the subsumption architecture on a physical hardware version of the experiment, using a standard Inverted Pendulum apparatus used at UWE for undergraduate courses in neuro-fuzzy control. This will allow the performance of the subsumption architecture, built using Space-time Distance and Direct Lyapunov Design methods, to be compared with controllers developed using existing techniques, for example fuzzy control and conventional PID control. Results of simulation performance are not available at time of writing of this paper, but will be available in the near future.

6 Related Work.

The work described in this paper was inspired by the ideas of Dennett [Ref. 21], who proposes that intelligence has evolved in biological creatures by a process of incremental extension to the spatial range and duration of interactions with the environment. This corresponds to the spatio-temporal distance concept for defining the layers of subsumption architecture.

Other researchers are also working on the development of methodologies for building behaviour-based agents. In particular, the work of Pfeifer & Scheier [Ref. 22] proposes eight principles to govern the design, for example:

- The "Complete Agent" Principle (requirement for self-sufficiency, embodiment, and situatedness)
- The "Cheap Design" Principle (requirement for parsimonious design, which exploits system dynamics)

The Pfeifer & Scheier methodology is to some extent complementary to the one presented in this paper. The principles proposed by Pfeifer & Scheier are heuristic in nature, but are wider in scope, covering issues such as the characteristics of agent environments and the design of agent body plans. The design principles defined in this paper are more strictly rational, but apply only to subsumption architecture and behaviour module design. The methodology in this paper does not address the wider issues. Therefore, both methodologies might form components of a wider project management scheme for agent design that may be developed in the future.

Other research, in particular the work of Menzer, Steinhage, and Erlhagen [Ref. 23], has also used stability theory as the basis for behaviour-based system design. They have developed a system technology and a supporting design methodology, based on proofs of stability of the behaviour patterns of the system. Their method is comparable to the Direct Lyapunov Design aspects of the work in this paper, but includes technology and methods for design arbitration schemes as well as the basic behaviour module design. However, their work does not provide a wider rationale for defining and organising the behaviours within an overall system architecture. Therefore, it may be possible to merge the Space-time Distance Design aspects of this paper with the detailed behaviour design aspects of [Ref. 23], to integrate the benefits of the two methodologies.

Viability Theory [Ref. 16] has also been proposed as a basis for behaviour-based agent design. The basic definitions of viability are very close to the definitions of Lyapunov stability, so it may be possible to incorporate elements of Viability Theory into the Direct Lyapunov Design procedures, if it should prove advantageous to do so.

7 Conclusions.

The design methodology proposed in this report follows a set of general principles and theorems, and therefore does not contain the flaws associated with inductive reasoning that may be inherent in learning or evolutionary techniques. The principles are general in nature, and therefore should be reasonably applicable to a variety of application problems.

While the theoretical aspects of this proposed methodology are quite well developed, experimental validation of the ideas is still at an early stage. There will doubtless be some modification of the methodology in the light of practical experience of system designs. It is anticipated that many of these modifications will arise from the specific control issues of different types of machine, which have different dynamical properties that affect the nature of agent-environment interactions.

It is an important aspect for the development of safety critical system technologies that they are sufficiently mature to be usable for safety critical applications. Novel technologies often present problems because the assumptions and constraints that are inevitably built into them must be well understood if these limitations are not to cause accidents. The current work has begun the maturation of the technology by starting with simple experiments, and the scope of experiments will be expanded gradually. Lessons in applying the design methodology are sure to be learnt at each stage.

8 References.

1. Popper K.R., *The Logic of Scientific Discovery*, Routledge 1980
2. Cliff D., Harvey I., Husbands P., *Explorations in Evolutionary Robotics*, Adaptive Behaviour Vol.2 No.1, pp 73-110, MIT Press 1993
3. Connell J.H., *Minimalist Mobile Robotics: A Colony-style Architecture for an Artificial Creature*, Academic Press 1990

4. Maes P., *Behaviour Based AI*, Proc. 2nd Intl. Conf. Simulation of Adaptive Behaviour (SAB 92) p2-10, MIT Press 1992
5. Maes P. (ed.), *Designing Autonomous Agents: Theory and Practice From Biology to engineering and Back*, Robotics and Autonomous Systems Vol.6 Nos.1-2, Elsevier 1990
6. Arkin R.C., *Behaviour Based Robotics*, MIT Press 1998
7. Brooks R.A., *A Robust Layered Control System for a Mobile Robot*, IEEE Jrn. Robotics and Automation, Vol. RA-2 No.1, p14-23, March 1986
8. Maes P., *Situated Agents Can Have Goals*, Robotics and Autonomous Systems Vol.6 Nos.1-2, p49-70, Elsevier 1990
9. Payton D., Rosenblatt K. *et al.*, *Do Whatever Works: A Robust Approach to Fault-Tolerant Autonomous Control*, Jrn. Applied Intelligence 2, p225-250, Kluwer Academic Publishers 1992
10. Arkin R.C., *Integrating Behavioural, Perceptual, and World Knowledge in Reactive Navigation*, Robotics and Autonomous Systems Vol.6 Nos.1-2, p105-122, Elsevier 1990
11. Beer R.D., Chiel H.J. & Sterling L.S., *A Biological Perspective on Autonomous Agent Design*, Robotics and Autonomous Systems Vol.6 Nos.1-2, p*169-186*, Elsevier 1990
12. Dorigo M. & Colombetti M., *Robot Shaping*, MIT Press 1998
13. Barnes J., *Programming in Ada 95*, Addison Wesley 1996
14. Arrowsmith D.K. & Place C.M., *An Introduction to Dynamical Systems*, Cambridge University Press 1990
15. McFarland & Bosser, *Intelligent Behaviour in Animals and Robots*, MIT Press 1993
16. Lyapunov A.M., *The General Problem of the Stability of Motion*, Taylor Francis 1992
17. Slotine J-J. & Li W., *Applied Non-linear Control*, Prentice Hall 1991
18. Aubin J-P., *Elements of Viability Theory for Animat Design*, Proc. 6th Intl. Conf. Simulation of Adaptive Behaviour (SAB 2000), MIT Press 2000
19. Barto A.G, Sutton R.S. & Anderson C.W., *Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems*, IEEE Trans. Systems, Man and Cybernetics, Vol. SMC-13 No.5, p834-846, September 1983
20. Michie D. & Chambers R.A., *BOXES: An Experiment In Adaptive Control*, Machine Intelligence 2, p137-152, Oliver and Boyd 1968
21. Dennett D., *Kinds of Minds*, Phoenix Publishers 1997
22. Pfeifer R. & Scheier C., *Understanding Intelligence*, MIT Press 1999
23. Menzner R., Steinhage A. & Erlhagen W., *Generating Interactive Robot Behaviour: A Mathematical Approach*, Proc. 6th Intl. Conf. Simulation of Adaptive Behaviour (SAB 2000), p135-144, MIT Press 2000

Appendix A: Proof of Minimum Interference between Behaviour Modules Organised Using the Diminishing Activity Property.

Theorem 1: The Diminishing Activity Property defines the condition of *minimum mutual interference* between behaviour modules in a Connell-type subsumption architecture (with suppression or inhibition arbitration only).

Proof:

Consider two Behaviour Modules P_1 and P_2 embedded within the same subsumption architecture. The activity of each module has the form illustrated below:

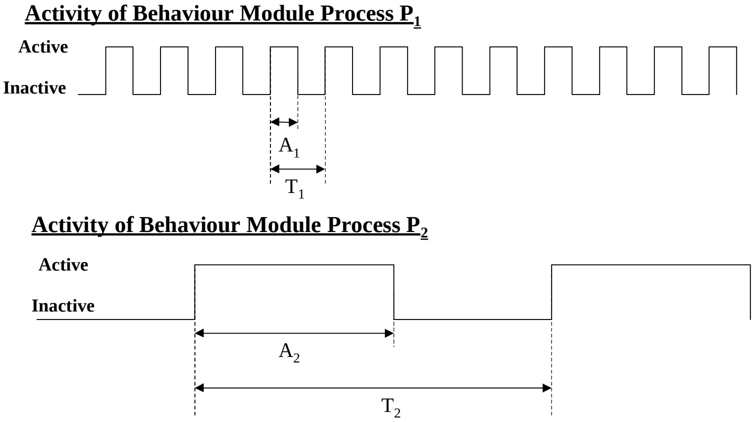


Figure 4: Relative Activity of Behaviour Module processes.

Each module process P_n has an average period T_n between successive bursts of activity. The average duration of activity is A_n. By definition, all A_n values are less than their respective T_n values.

We define process P_2 to have a longer average period between activities than P_1. Let k be the constant of proportionality between the two:

$$T_2 = k.T_1 \qquad (1)$$

We define the *mark/space ratio* R_i of a process as the proportion:

$$R_i = \frac{A_i}{T_i} \qquad (2)$$

Let the *interference* on a process be defined as the suppression or inhibition of its outputs by another process. For processes P_1 and P_2 the interference of one process on the other can be defined in terms of the proportion of time they are both active in a given period of the suppressed process.

If P_1 suppresses P_2:

$$\text{Average proportion of } T_2 \text{ in which } P_1 \text{ is active} = \frac{A_1}{T_1} \cdot \frac{T_2}{T_1} = k \cdot R_1$$

$$\text{Proportion of } P_2 \text{ activity lost due to } P_1 \text{ suppression during } T_2 = k \cdot R_1 \cdot \frac{A_2}{T_2} = k \cdot R_1 \cdot R_2$$

If P_2 suppresses P_1:

$$\text{Average proportion of } T_1 \text{ in which } P_2 \text{ is active} = \frac{A_2}{T_2} \cdot \frac{T_1}{T_2} = \frac{R_2}{k}$$

$$\text{Proportion of } P_1 \text{ activity lost due to } P_2 \text{ suppression during period } T_1 = \frac{R_2}{k} \cdot \frac{A_1}{T_1} = \frac{R_1 \cdot R_2}{k}$$

Since $k > 1$ by definition, it can be seen by inspection that the interference to P_2 if P_1 suppresses it is less than the interference to P_1 if P_2 suppresses it.

> Therefore, the minimum inter-process interference between two processes will be achieved if the process (P_2) with the longer average period between bursts of activity suppresses or inhibits the process with the shorter period (P_1). QED.

The proof can be extended to complete subsumption architecture systems by considering the combined effect of higher layers on the modules of an arbitrary behaviour module, yielding the Diminishing Activity Property as defined in Principle 5.