ON STABILITY OF BEHAVIOUR-BASED SYSTEMS [1]

Jacek Malec [*,**] **Dimiter Driankov** [*]

[*] *Department of Computer and Information Science, Linköping University, S–581 83 Linköping, Sweden*
(jam,ddr)@ida.liu.se
[**] *Department of Computer Engineering, Mälardalens Högskola Box 883, S–721 23 Västerås, Sweden*

Abstract: Behaviour-based control is becoming a well-established control paradigm. However, despite of its wide use, the design of behaviour-based control systems is still more an art than engineering. It requires manual tuning of numerous parameters and extensive simulation and experimentation in order to design a collection of individual behaviours that can interact with each other in a consistent and cooperative manner. Therefore, it is important that more attention is paid to modelling and analysis methods that may lead to better understanding of the processes occurring in a network of interacting behaviours. In this paper we report a first step towards a method for modelling and analysing behaviour-based control systems. We introduce a network model of a behavioural controller (i.e., a collection of interacting behaviours) and study the concept of stability for such a network employing techniques from graph theory.

Keywords: behaviour-based control, intelligent control, stability

1. INTRODUCTION

Behaviour-based control is becoming a well-established paradigm for controlling autonomous agents, both material, like robots, and immaterial, like softbots. However, despite of its wide use, this type of control lacks the formal analysis and design techniques available in other control approaches. Design of behaviour-based control systems (BCS) is more an art than engineering and requires often manual tuning of numerous parameters and extensive simulation and experimentation in order to design a collection of individual behaviours that can interact with each other in a consistent and cooperative manner. Moreover, this way of building BCSs does not offer any guarantees for successful operation: unexpected (untested) environmental conditions may force the BCS to act erratically or simply fail due to unexpected interaction between behaviours.

Therefore, it is important that more attention is paid to modelling and analysis methods that may lead to better understanding of the processes occurring in a network of interacting behaviours. This in turn may also provide a basis for more systematic BCS design methods and better scalability of the behaviour-based approach to control of autonomous agents.

In this paper we report a first step towards a method for modelling and analysing behaviour-based control systems. We introduce a network model of a behavioural controller (i.e., a collection of interacting behaviours) and study the concept of stability for such a network employing techniques developed during the early 70s in graph theory.

[1] This research has been supported by the Swedish Research Council for Engineering Sciences.

In Sect. 2 we motivate our work in the context of related research in the area of BCSs. In Sect. 3 we introduce a directed, weighted graph model for interacting behaviours. In Sect. 4 we present the basic results related to pulse stability in weighted, directed graphs obtained by Roberts in (Roberts, 1978). In Sect. 5 we give an example of the application of pulse stability to a network of interacting behaviors. Finally, in Sect. 6 we provide some preliminary conclusions.

2. MOTIVATION

The main property of BCSs is that they consist of a number of self-contained routines (called behaviours) that are capable of executing in parallel. Each behaviour reflexively responds to sensor readings by sending a control signal to a particular actuator which in turn generates an actuator output (action). This is done in a tight sensor-action loop which is independent of the sensor-action loops of other behaviours and can perform without extensive communication with them. Thus, the coordination between behaviours can be achieved via an arbitration scheme that makes a collection of behaviours, run in parallel, to perform in a consistent, functional way.

There are many ways for solving the arbitration problem and these span across several dimensions like priority, applicability, utility, dominance, and activation level of behaviours. In the actuator arbitration associated with the Brooks' subsumption approach (Brooks, 1986) the behaviour inconsistency is resolved by fusing competing control signals to the same actuator using a predefined priority scheme. For instance, higher layer's composite behaviour may inhibit lower layer primitive behaviour's actuator output. The priority scheme may have fixed or variable priorities (e.g., a function of varying goal priorities, varying reliability of sensors, etc.), but in any case the actuator output or the control signal to the same actuator that is issued by a behaviour with the highest priority is the **only** one allowed. In other words, according to this type of arbitration scheme a behaviour is either active or inactive.

Arbitration schemes that subsume the one described above, but are much more flexible are employed in fuzzy BCS. Here each behaviour is represented by a collection of fuzzy if-then rules. The antecedent (if-part) of each rule is defined over the fuzzy space of sensory measurements needed for the activation of a behaviour and the consequent (then-part) is defined over the fuzzy space of the control signal to the actuator associated with this behaviour. Inconsistencies between behaviours arise since two or more behaviours may have fuzzily overlapping antecedents and/or consequents.

In (Pin and Watanabe, 1993) the inconsistencies are resolved by using elaborate de-fuzzification of competing control signals, either to the same or to different actuators. The de-fuzzification mechanism uses the notion of dominance between fuzzy sets expressing the size of competing control signals to the same actuator or the size of competing actuator outputs. Thus, the arbitration is realized by using the information already available in the fuzzy if-then rules and no additional priority scheme is required.

In (Tunstel, 1995) the arbitration scheme is realized by an additional collection of fuzzy if-then rules representing a flexible priority scheme. The antecedent of a rule is again defined on the fuzzy sensory space that activates a particular behaviour. However, the antecedent is defined over the fuzzy space of the "applicability" of this behaviour. Inconsistencies arise due to the fuzzily overlapping sensory domains for the different behaviours: overlapping antecedents imply the simultaneous presence of a number of applicable behaviours each to a different fuzzy degree of applicability. Here again a de-fuzzification is employed to produce a crisp degree of applicability for each competing behaviour which in turn is translated into a corresponding degree of activation for these behaviours. Thus, in contrast to Brooks' arbitration scheme, a behaviour can be active/inactive to a degree, rather than being just active or inactive.

A similar arbitration scheme is employed in (Ghanea-Hercock and Barnes, 1996). Here each behaviour is defined as a stimulus/response function. Each such function has an associated stimulus/utility function that describes the "importance" (or utility) of a behaviour's response (performance) to the environment at any point in time. This utility function is represented as a set of fuzzy if-then rules mapping sensory input (if-part) to behaviour's utility. It also applies a negative feedback on the utility of behaviours that are simultaneously activated. Thus, the contribution (activation) of each behaviour is adjusted dynamically to suit the current sensory input. Inconsistencies are resolved by linear superposition of individual "utilitor" vectors for each behaviour. Such a vector for a particular behaviour consists of values for its current utility and performance.

It is important to observe here that none of the above approaches to arbitration uses a "global model" of the possible inhibiting or enhancing influence that behaviours may have on each other independently of their current priority, activation level, utility, etc. It is after all these inherent influences between behaviours that cause inconsistencies, and which magnitude can then be further increased or decreased depending on the current priority, size of activation level, utility, etc. If knowledge about the degree to which one behaviour inhibits another behaviour is available in advance then obviously the activation level of the first one will always be larger than this of the second one. Consequently, no arbitration scheme can reverse this situation; it can only passively reflect it by always assigning a higher activation level to the first behaviour and a lower activation level for the second one. More intricate relationships between behaviours may occur when the influence which one behaviour has on another one is propagated further to the rest of the behaviours via a network of interacting behaviours.

The above observations are reflected in a number of other approaches to arbitration (Maes, 1991; Payton, 1990; Saffiotti *et al.*, 1993). A network describing the influences (inhibition or enhancing) between behaviours is constructed in advance and used to propagate change in the activation level of one behaviour to the rest of the behaviours. Such a network of behaviours has basically two uses: i) to compute the activation levels of behaviours given the current (or desired) activation level of particular behaviour(s), and ii) to help isolate consistent subsets of behaviours.

Since this latter approach is more general and the previously discussed ones are implicit in it, we are going to use it as the background for the model of interacting behaviours we are going to present and analyse in the following sections.

3. A MODEL OF INTERACTING BEHAVIOURS

A network of interacting behaviours can be modeled as a directed graph D, where nodes represent single behaviours, and arcs represent the presence and direction of influences between behaviours independently of the current size of their activation levels. In principle, this is the formal representation of the works by Maes (Maes, 1991) and Payton (Payton, 1990).

Each node, u_i is associated with a function of time v_i representing the evolution of the behaviour's activation level, that is $v_i : T \to dom_i$, where dom_i is the activation level domain for the behaviour u_i, and T is the time domain (discrete or continuous).

The magnitude of the influence $w(u_i, u_j)$ between every two behaviours u_i and u_j is placed as the label on the arc connecting them and specifies what amount of u_i's activation is propagated to u_j. The sign of the label can be either positive (enhancing), i.e., $w(u_i, u_j) > 0$ or negative (inhibition), i.e., $w(u_i, u_j) < 0$. This directed, weighted graph has a compact representation in terms of its adjacency matrix $\mathbf{A}$.

This model of a network of interacting behaviours is based on the following assumptions:

- T is discrete due to the sampled nature of the sensory input, and isomorphic with the natural numbers.
- $dom_i = dom$ for all i that is, all activation level domains are of equal size.
- The propagation of influence from one behavior to another one takes one time step that is, from t to $t + 1$.
- Some of the behaviours have their activation level increased or decreased spontaneously as the effect of the interaction with the environment, or by some internal mechanism (e.g., the increasing need for re-charging). This change in activation level is represented by $p_i(t)$. Then this activation level is propagated to other behaviours, increasing or decreasing their activation levels.

This propagation of influences may lead to two alternatives:

- A stable level of activation, i.e., all behaviours achieve some constant level of activation such that without any further spontaneous change their activation levels don't change.
- A unstable level of activation (some activation levels increase or decrease infinitely, or become saturated).

The first alternative prompts the question: What has to change in an already stable network of interacting behaviours so that the network becomes unstable? In other words: What are the limits of the network's stability? The second alternative raises the question: How an unstable network of interacting behaviours can be transformed into a stable one and what will be the consequences of such a change?

In the following section we introduce the technique that might be used to analyse a network of interacting behaviours, modelled as a directed, weighted graph, for the purpose of identifying stable and unstable activation levels.

4. PULSE PROCESSES

The material in this section is a direct recapitulation of some results obtained by Roberts in from (Roberts, 1978).

Let D be a weighted directed graph (digraph) and $u_1, u_2, \ldots, u_n$ its vertices. We assume that each vertex u_i attains a *value* or level $v_i(t)$ at each time t, and that time takes discrete values,

$t = 0, 1, 2, \ldots$. We are interested in the *change in value* (or *pulse*) at vertex u_i at time t. This will be denoted by $p_i(t) = v_i(t) - v_i(t-1)$ if $t > 0$.

For $t = 0$, $p_i(t)$ and $v_i(t)$ have to be given as initial conditions. Then the interpretation of the weight $w(x, y)$ induces the following rule for change of values:

$$v_i(t+1) = v_i(t) + \sum_i w(u_j, u_i) p_i(t).$$

Therefore

$$p_i(t+1) = \sum_i w(u_j, u_i) p_i(t).$$

The initial conditions $p_i(0)$ and $v_i(0)$ together with the equations above are said to define an *autonomous pulse process.*

Let $\mathbf{A}$ be the *weighted adjacency matrix* of the weighted digraph D, the $n \times n$ matrix where its i, j entry is equal to $w(u_i, u_j)$ if there is an arc from u_i to u_j and equal to 0 otherwise. The last equation can be rewritten as

$$P(t+1) = P(t)\mathbf{A},$$

and hence

$$P(t) = P(0)\mathbf{A}^t.$$

It follows that

$$V(t) = V(0) + P(0)(\mathbf{A} + \mathbf{A}^2 \ldots + \mathbf{A}^t).$$

We say that a vertex u_i is *pulse stable* under an autonomous pulse process if the sequence $\{|p_i(t)| : t = 0, 1, 2, \ldots\}$ is bounded, and u_i is *value stable* under the pulse process if the sequence $\{|v_i(t)| : t = 0, 1, 2, \ldots\}$ is bounded. Value stability implies pulse stability. However, the converse is false.

We say that a weighted digraph is pulse or value stable if every vertex of the digraph is. If a digraph is pulse or value stable under *all* autonomous pulse processes, then it is a symptom of its global, structural stability, independent of the initial perturbation.

The following results are the most interesting ones in the context of networks of interacting behaviours.

Theorem 1. Suppose D is a weighted digraph and A is its weighted adjacency matrix. If D is pulse stable under all autonomous pulse processes, then every eigenvalue of A has magnitude less than or equal to one.

Theorem 2. Suppose D is a weighted digraph and A is its weighted adjacency matrix. If every eigenvalue of A has magnitude less than one, then D is pulse stable under all autonomous pulse processes.

Theorem 3. Suppose D is a weighted digraph and A is its weighted adjacency matrix. Then D is value stable under all autonomous pulse processes if and only if D is pulse stable under all autonomous pulse processes and 1 is not an eigenvalue of A.

Theorem 4. Suppose D is an integer-weighted digraph and

$$C(\lambda) = \lambda^n + \sum_{k=1}^{n} b_k \lambda^{n-k}$$

is its characteristic polynomial. Suppose that some $b_k \neq 0$ and let s be the highest subscript such that $b_s \neq 0$. If D is pulse stable under all autonomous pulse processes, then $b_s = \pm 1$ and $b_i = b_s b_{s-i}$, for $i = 1, 2, \ldots, s-1$.

Theorem 5. Under the hypotheses of the previous theorem, D is value stable under all autonomous pulse processes if and only if D is pulse stable under all autonomous pulse processes and

$$\sum_{i=1}^{s} b_i \neq -1.$$

5. STABILITY

We will now directly apply the results from the previous section to study the stability of a network of interacting behaviors modeled as outlined in Sect. 3.

Let us take as an example the simple network of behaviours analysed by Maes in (Maes, 1990) and shown in Fig. 1. We have annotated the arcs of the network with the magnitudes of influence, i.e., $w(u_i, u_j)$ quoted in (Maes, 1990). A + denotes a source of energy flowing into the network - in terms of pulse processes these arrows point to the nodes where pulse processes have their origin. Therefore, in cases when the network is not pulse stable (i.e., not stable for all pulse processes), it might be especially interesting to analyse just processes originating in those nodes.

Let us interpret the network in terms of the results from the previous section. Relabeling the nodes as follows: SAND-BOARD $\rightarrow u_1$, PICK-UP-SANDER $\rightarrow u_2$, PUT-DOWN-SPRAYER $\rightarrow u_3$, PUT-DOWN-SANDER $\rightarrow u_4$ yields the following adjacency matrix $\mathbf{A}$:

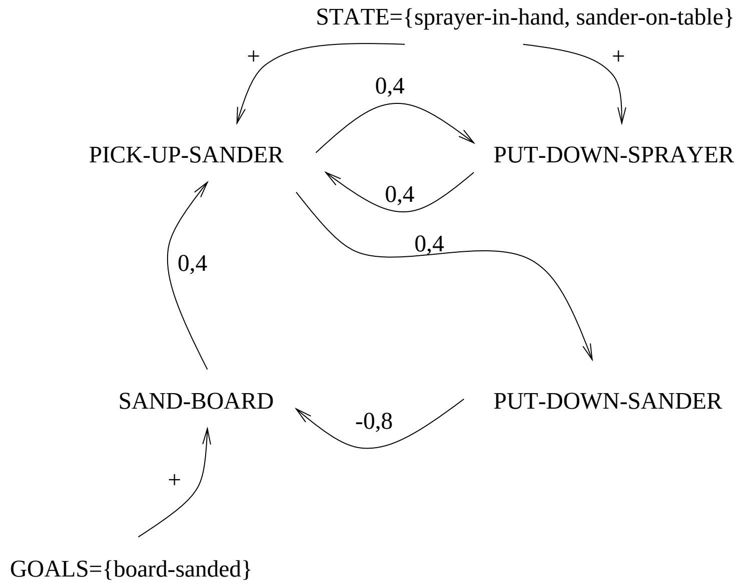

Fig. 1. A snapshot of spreading activation in a behaviour network.

	u_1	u_2	u_3	u_4
u_1	1	0.4	0	0
u_2	0	1	0.4	0.4
u_3	0	0.4	1	0
u_4	-0.8	0	0	1

The characteristic polynomial of $\mathbf{A}$ is $C(\lambda) = (\lambda - 1)(\lambda^3 - 3\lambda^2 + 2.84\lambda - 0.712)$ and it has the following roots: $1, 0.3914, 1.3043 + 0.3431i, 1.3043 - 0.3431i$. As a consequence of Theorem 4.3 ($\mathbf{A}$ has an eigenvalue 1) we can see that the network cannot be value stable under all autonomous pulse processes. Therefore of some nodes might grow infinitely for some stimuli.

It has been proposed in (Maes, 1990) and (Payton, 1990) that the node with the highest activation level (i.e., the one that becomes "active") gets its activation level reset to 0. However, this resetting of "activation" seems to be introduced as a necessity to avoid instabilities in the network rather than as a design principle, and cannot be considered as a feature. Some more advanced models of interacting behaviors (see Sect. 2) allow multiple behaviours to be active (to a different degree) at the same time. The mechanism from (Maes, 1990) introduces moreover that at such activation/resetting point the "energy sources" might change their places so that other nodes obtain new inflow sources and thus the topology of the network might change as well. This property is a desired one: it allows to model changing goals of the BCS and to plan (reactions) leading to their satisfaction. For those cases, the results presented in Sect. 4 would appear even more useful, as for any given stimulus ("energy" inflow to the network) we can decide whether the activation levels of the nodes are going to eventually stabilize or not. However, this topic is outside the scope of this paper.

It seems that the questions stated earlier, namely: i) What are the limits of network's stability?, ii) How an unstable network of interacting behaviours can be transformed into a stable one and what will be the consequences of such change?, can be (at least partially) answered using the pulse stability technique described here. The toy example above could illustrate just very simple and straightforward calculations, but one can imagine much more advanced uses of these results like, e.g., analytical search for stability regions in a specific network of behaviours, or deciding the consequences of introducing a coupling between two nodes while realizing particular task (i.e., being subject to a specific pulse process).

6. CONCLUSIONS

In this paper we have presented a technique that allows one to analyse networks of interacting behaviours and, in particular, their stability: local (subject to a specific pulse process) and global (for all pulse processes). We have provided motivation for our research, describing various approaches to behavioural control, and we have observed that there is little available to help the engineer in the process of designing such control systems. We consider the possibility of formal analysis of behaviour networks as one step towards remedying this situation. We have presented selected results from (Roberts, 1978) and illustrated their direct applicability using a simple example from the literature.

This is obviously only the very first step in an interesting direction. In the future we intend to investigate the possibility of thorough analysis for time-varying (non-stationary) networks, networks with nonlinearities like thresholding, and the questions of on-line computations supporting dynamicreconfiguration of networks.

7. REFERENCES

Brooks, Rodney A. (1986). A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation* **2**, 14–23.

Ghanea-Hercock, R. and D. Barnes (1996). Coupled behaviors in the reactive control of cooperating mobile robots. *Int. Journal Advanced Robotics*.

Maes, Pattie (1990). Situated agents can have goals. *Robotics and Autonomous Systems* **6**, 49–70.

Maes, Pattie (1991). A bottom-up mexhanism for behavior selection in an artificial creature. In: *From Animals to Animats* (J. Meyer and S. Wilson, Eds.). MIT Press.

Payton, David W. (1990). Internalized plans: A representation for action resources. *Robotics and Autonomous Systems* **6**, 89–103.

Pin, F.G. and Y. Watanabe (1993). Driving a car using reflexive fuzzy behaviors. In: *Proc. 2nd FUZZ-IEEE Conference*. San Francisco, California. pp. 1425–1430.

Roberts, F.S. (1978). *Graph Theory and Its Applications to Problems of Society*. Society of Industrial and Applied Mathematics. Philadelphia, PA.

Saffiotti, Alessandro, Enrique H. Ruspini and Kurt Konolige (1993). Blending reactivity and goal-directedness in a fuzzy controller. In: *Proc. 2nd IEEE Conf. on Fuzzy Systems*. pp. 134–139.

Tunstel, E. (1995). Coordination of distributed fuzzy behaviors in mobile robot control. In: *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*. Vancouver, Canada. pp. 4009–4014.