

# Hybrid Automata as a Modelling Approach in the Behavioural Sciences

Matthias Buntins<sup>1,5</sup>   Jens-W. Schicke<sup>2,6</sup>  
Frank Eggert<sup>3,5</sup>   Ursula Goltz<sup>4,6</sup>

---

## Abstract

Hybrid modelling techniques have evolved as a tool to design, verify and guide the implementation of embedded systems. However they can – and we think should – be used to express quantitative models about hybrid systems in other domains, e.g. empirical sciences. Since the formal structure of hybrid automata corresponds well to sequentially organized behaviour chains in living organisms, we argue for an application of hybrid modelling techniques in the behavioural sciences and, especially, psychology.

We try to answer the question how human drivers move onto a freeway and at the same time use this research as our testbed for using hybrid automata within behavioural sciences.

*Keywords:* Hybrid automata, hybrid systems, human psychology

---

## 1 Introduction

Hybrid automata have been developed as a formalism to model systems in which discrete control logic interacts with a real-valued reality and facilitate mathematical proofs about their behavioural properties. The discrete control logic usually conceptualised is silicon based, but hybrid automata make no such presumption. Hybrid automata are classically used for modelling embedded systems and their software in different fields of technical engineering. The focus lies on dependability of such systems and hence on formal verification of properties. Several model checkers [6,4] have been developed for this purpose. Besides their formal semantics, hybrid automata offer a pleasing visual notation accessible with only a minimum of formal training.

---

<sup>1</sup> m.borgstede@tu-bs.de

<sup>2</sup> drahflow@gmx.de

<sup>3</sup> f.eggert@tu-braunschweig.de

<sup>4</sup> goltz@ips.cs.tu-bs.de

<sup>5</sup> Research Methods and Bio-Psychology, TU Braunschweig

<sup>6</sup> Institute for Programming and Reactive Systems, TU Braunschweig

However, hybrid systems occur not only in computer science and other engineering disciplines, but also in many other fields. Autonomous hybrid systems can be found in biology, psychology and – as a multi-agent setting – social sciences in general. One aim of these sciences is to generate a conceptual model of the respective systems, and to test it empirically. A formal notation would improve both accessibility and comparability of different models. Ideally, such a formal notation would not hinder readers unacquainted with the notation from fully understanding what the author wishes to express.

We will propose here to use hybrid modelling formalisms, in particular hybrid automata, as a modelling notation in sciences other than computer science. We argue, and will show in this paper by a prototypical application, that the formal apparatus of hybrid automata is especially suitable to construct, simulate and test empirical models of complex behaviour in humans and other organisms. Of course, it would be an interesting perspective to use the power of model checkers as a formal verification tool in this new application domain. However, the empirical nature of these sciences is more closely reflected in simulation approaches.

The application we have chosen to illustrate our approach is the behaviour of human drivers in traffic. The obvious hope is that a model of a human driver will enhance safety and efficiency, e.g. by using it for developing suitable driver assistance systems. Hence a number of approaches in psychology currently investigate the quantitative modelling<sup>1</sup> of car driving with drivers as single subjects. This is in contrast to other quantitative models of behaviour, which usually refer to the behaviour of larger populations of organisms (for example in the domain of predator-prey interaction [3]). In the domain of more complex behaviour of single subjects, the approach is usually to construct a model of cognitive processes, which are intrinsically unobservable (see [2]). Additionally, these models usually imply that complex behaviour is under the control of conscious psychological processes – an assumption which can hardly be justified when dealing with genetically determined or highly consolidated behaviour. As a consequence, plausible quantitative models of complex behaviour on the single subject level can hardly be found at all. Instead of presenting a formal model, researchers in these fields often resort to semi-formal descriptions to explain the interactions between various variables of the subject in question. In contrast to these approaches, we concentrate here on a behavioural model on the single subject level (i.e. a model describing the behaviour of a single organism), restricted to rather simple stimulus-response relations. We will introduce a modelling approach based on so-called reinforcement values and formalise it in terms of hybrid automata. We have implemented a simulation framework for hybrid automata coupled with an optimization process based on generic algorithms. We present first simulation results showing the feasibility of the approach.

Section 2 will introduce hybrid automata formally. Section 3 introduces a conceptual model for behaviour using reinforcement values and its formalisation in terms of hybrid automata, and illustrates the approach by some simpler examples

---

<sup>1</sup> In the current paper the term "quantitative" refers to models specifying the relations between continuous variables.

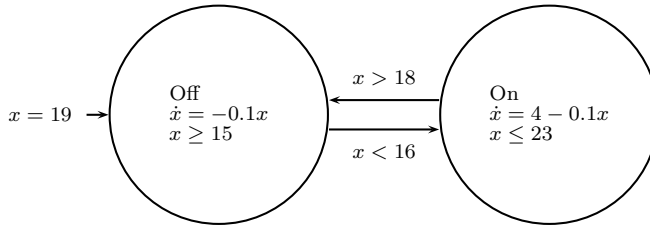


Fig. 1. The obligatory thermostat example from [5].

of behaviour. Section 4 will start with a discussion of existing work on modelling human drivers. We then model and simulate the behaviour of drivers entering a freeway. Finally Section 5 will, besides summarising, suggest certain extensions of hybrid automata to ease modelling human behaviour.

## 2 Hybrid Automata

Our description of hybrid automata follows [5] closely.

A hybrid automaton is a model of a hybrid system, i.e. a system consisting of a discrete (usually controlling) part, and a real-valued and continuous part. Formally, a hybrid automaton is a finite automaton extended with a set of real-valued variables and various possibilities to modify and test these variables.

**Definition 2.1** A *hybrid automaton* consists of

- a finite set  $X = \{x_1, x_2, \dots, x_n\}$  of real-valued variables, and two related sets  $\dot{X}$  and  $X'$  denoting the derivatives during continuous change and the new values after discrete change respectively,
- a finite directed multigraph  $(V, E)$  (*control modes* and *control switches*),
- three functions *init*, *inv*, and *flow*, labelling vertices with predicates over  $X$ ,  $\dot{X}$ , and  $X \cup \dot{X}$  respectively (*initial condition*, *invariant condition*, *flow condition*),
- a function *jump*, labelling edges with predicates over  $X \cup X'$ ,
- a finite set of events and a function *event* labelling edges with events.

The prototypical example is that of the thermostat of Figure 1. The thermostat consists of two discrete control modes, aptly named “off” and “on” and a single variable  $x$  denoting the temperature. The *init* function maps “on” to a constantly false predicate and “off” to  $x = 19$ , as denoted by the single arc without a start on the left. The system starts with the thermostat switched off and the temperature at 19 (degrees Celsius for example). As long as no discrete control switch occurs, the temperature sinks with  $\dot{x} = -0.1x$ , i.e. exponentially. Once the temperature sinks below 16, the thermostat may switch to “on”. Even though the figure seems to indicate the jump predicate to be  $x > 18$  it is usually understood to implicitly include  $x' = x$  as well (as a discrete jump in the control structure only uncommonly induces a discrete jump in the real-valued part of the system). The automaton might however decide not to switch on immediately but wait a bit longer. While remaining in the “off” state however, the temperature shall not fall below 15, as

denoted by the invariant *inv*, and the automaton must switch to “on” at that point at latest. The “on” state works similar, and for sake of brevity we omit a detailed (and purely redundant) description here.

### 3 Hybrid modelling of behaviour

This section introduces our methodology of modelling behaviour and its formalisation in terms of hybrid automata, illustrated by rather simple examples of behaviour. Section 4 will then introduce our specific application and an as-to-date tentative model of human drivers in greater detail.

#### 3.1 Hybrid Automata as a formalism for the behaviour of organisms

The behaviour of organisms can be structured as a hierarchy of different behaviour systems and subsystems (see Figure 2). Depending on internal states and external influences the organism may change from one behaviour system to another. The controlling variable of such categorical behaviour changes is thought to be an internal representation of all relevant situational factors, weighted by their anticipated outcomes – the so-called *reinforcement value*. The reinforcement value can be thought of as a “common currency” by which all anticipated consequences of a behaviour are evaluated. Switching from one behavioural state to another may occur within a single behaviour system, as well. An illustrative example are so called Sequential Action Patterns (SAPs). SAPs are fixed behavioural sequences, which are elicited by a biologically relevant stimulus [7]. In each state of the sequence there is a set of variables which control the behaviour of the organism in a certain way. The influence of the controlling variables in a certain behavioural state could – in principle – be specified by a set of differential equations. If certain variables reach a critical value, however, the organism switches to another state in which a different set of variables may control the behaviour. It may also occur that the same variables influence the behaviour in a different way. The change from one behavioural state to another within an SAP may thus be expressed as a discrete transition from one set of differential equations to another.

We will model an SAP as a hybrid automaton by representing each discrete state of the organism as a control state of the automaton. The assumed variables of the organism are represented by variables in the automaton and the differential equations governing the variables of the organism are associated to control states as indicated by the mapping from behaviour state to control state.

#### 3.2 Examples of SAPs

The potential of this approach will be illustrated in this section by a few examples for sequentially organized behaviour in animals and humans.

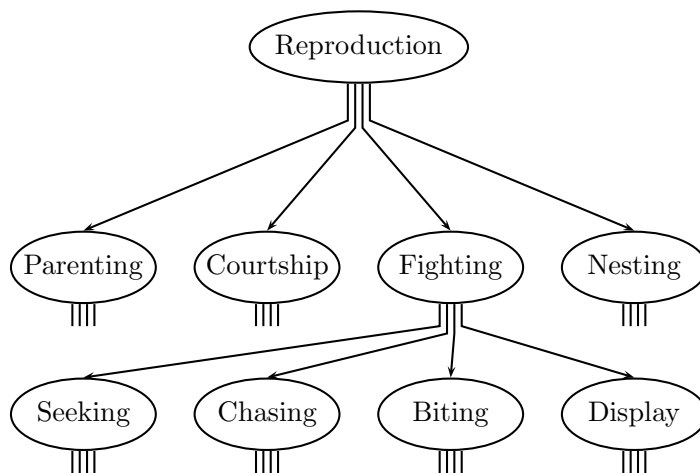


Fig. 2. Hierarchical organization of reproductive behaviour (adapted from [7])

### *Mating*

The three-spined stickleback (*Gasterosteus aculeatus*) is a species of small fish with a rather complex mating behaviour, which is organized in three distinct stages. At first, the male establishes a territory and constructs a nest by digging a hole in the ground and collecting soft material. In this stage the presence of a female does not have any influence on the male's behaviour. As soon as the nest is finished, however, females entering the territory are courted and induced to lay eggs in the nest. Once the eggs are laid the presence of a female functions as a trigger to chase it away [9].

From this textual description, one can derive an automaton similar to Figure 3. Modelling the volume of the hole dug so far by  $v$  the automaton starts in the state “Build” where the volume of the hole is increased steadily. Once the hole has reached the desired size (the automaton *has* to make the threshold specific), courting behaviour begins. Our model assumes the existence of a nearest female, the  $d$  variable tracking its distance to the nest. During courting, the distance is lowered until it reaches zero, at which point the female is chased away.

### *Feeding Behaviour*

Figure 4 shows the hybrid automaton representation of a preliminary foraging model. The model consists of three higher order behaviours – eat, idle and search – with a second behavioural level nested within search behaviour. The main factor in this model is a steadily increasing variable  $x$ , representing the reinforcement value of food. After it has reached a certain threshold value, the animal will start foraging, starting with a global search, which focuses on potential food sources (for example a certain type of tree, which signals the availability of edible fruit). After a potential food source has been encountered, the animal switches to a focal search mode, where

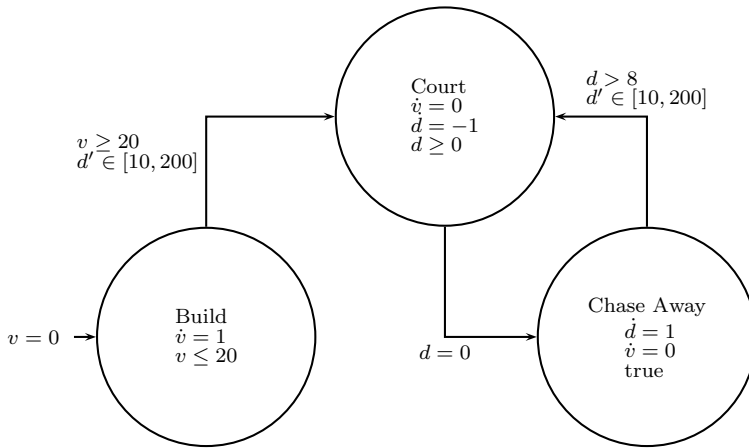


Fig. 3. A tentative model for the courting behaviour of the three-spined stickleback

it looks for specific food items (for example a certain type of fruit) until a certain amount of time ( $t$ ) has passed. In case that no food has been found, the animal switches back to global search. Note that the model does not distinguish between different potential food sources – thus it seems possible for an animal to return to the same source several times. This kind of behaviour may occur in animals with rather low memory capacities.

If, however, food has been found (we model the amount by  $y$ ), the animal will start consuming it, thereby lowering the reinforcement value of food until either the animal is satiated or the food source depleted. The specific numeric values of the model will be different for different species.

### *Easter Egg Search*

Similar to the distinction between global and focal search mode in animal foraging, humans also sometimes switch between different strategies while executing a single task. Consider the yearly task of finding easter eggs hidden within some well defined area. The switching of strategies can be described by the hybrid automaton of Figure 5. Initially, one scans the area, trying to look for locations which appear to promise good concealment. Having identified such a location, one moves there and tries to find an egg therein (which might not indeed exist, as modelled by *egg*). The actions exhibited during the two phases of search are clearly different. Note that during the global search mode the occurrence of an easter egg has no influence on the searcher's behaviour. Empirically this means that one does not notice an egg if one is looking for potential hiding places. Since easter egg search has rarely been investigated systematically until the present day, this prediction remains hypothetical – indicating the potential of hybrid modelling to generate empirically testable hypotheses.

Most humans also note the decline in search success once most eggs have been found and finally abandon the search. We model this by keeping track of the time  $t$  during which no egg was found.

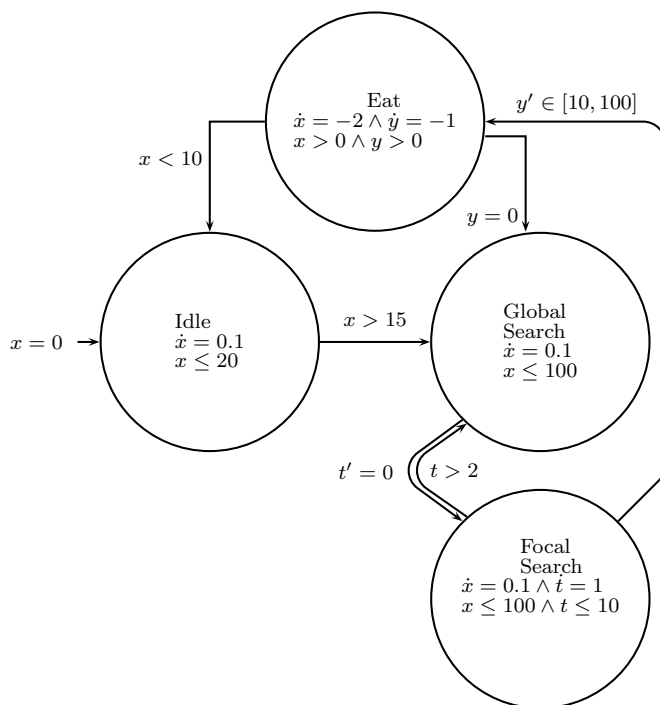


Fig. 4. A tentative model for foraging behaviour

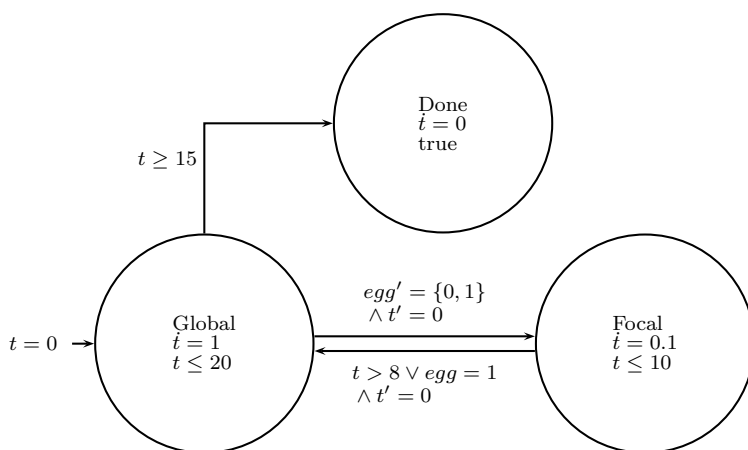


Fig. 5. A tentative model for easter egg searching

## 4 Hybrid Behaviour Simulation of Human Drivers

As already noted, we will try to use hybrid automata modelling to answer questions about life and death, which is to say, behaviour in traffic. For a person who has just learned to drive a car, conscious cognitive processes determine most of the driving behaviour. As experience is accumulated, however, the behaviour tends to become automated to a considerable degree. Thus, steering, braking and even lane changes happen without the driver's consciousness being necessary for controlling it. Both

for the design of advanced driver assistance systems and the design of the driving environment it would be useful to implement this fact in a driver model.

We took this problem as our testbed for the applicability of hybrid automata to the modelling of human behaviour. As a first step of the research, we restrict our attention to the situation of driving onto a freeway.

Models of the behaviour of human drivers are usually situated in the domain of cognitive psychology. So-called cognitive driver models treat the driver's behaviour as a result of some complex information processing sequence, which can be divided into several stages of perceiving, evaluating, goal-setting and deciding (for example [11,10]). In the context of the IMoST Project [1], a formalisation of such models by hybrid automata started, but turned out to be quite complex.

Cognitive models, however, suffer from some severe problems: Firstly they model processes that are intrinsically unobservable. And secondly, they tend to become highly complicated, even when modelling simple situations. We therefore ground our model on observable behaviour, namely trajectory and speed, restricting ourselves to only a single internal “hidden variable” – the reinforcement value. Our results will show that, for our setting, it is indeed possible to model such complex behaviour faithfully by a rather simple model. We are confident that the approach scales for even much more complex situations, e.g. situations considering several interacting drivers.

The approach put forward in this paper is radically different from traditional, cognitive models of driving behaviour. As a starting point we take the assumption that most driving behaviour is a result of implicit feedback-learning processes such as operant and classical conditioning. It appears sensible, therefore, to postulate some kind of reinforcement value of relevant parameters of the situation in question. By reinforcement value we mean the amount to which an outcome is more preferable to possible alternatives (“to collide with another car”, for example, is less preferable than “to reach travelling speed”). This reinforcement value does not need to be represented consciously in the organism. Neither does it refer to a set goal, towards which the behaviour of the organism is driven. The reinforcement value in a given situation is a (purely theoretical) function of the anticipated consequences of a possible behaviour. It may depend on biological determinants (genetic predispositions), as well as on the organism's learning history. In the context of acquired behaviours such as driving, the reinforcement value can be taken to be the result of the consequences of past driving behaviour. Abrupt changes in trajectory, for example, are associated with a negative reinforcement value – not due to an evaluative process occurring in the driver – but rather because of (potentially dangerous) past consequences of abrupt trajectory change. By means of the mentioned feedback learning processes the behaviour eventually approximates an optimal solution for a given situation. Thus, our main hypothesis is that drivers tend to behave in such a way that the reinforcement value in a given situation is maximized.

To reflect the optimising process of the organism in our models, we include into the hybrid automata “unknown” functions ( $f$  and  $g$  in our model), and a reinforcement value variable ( $q$  in our model).



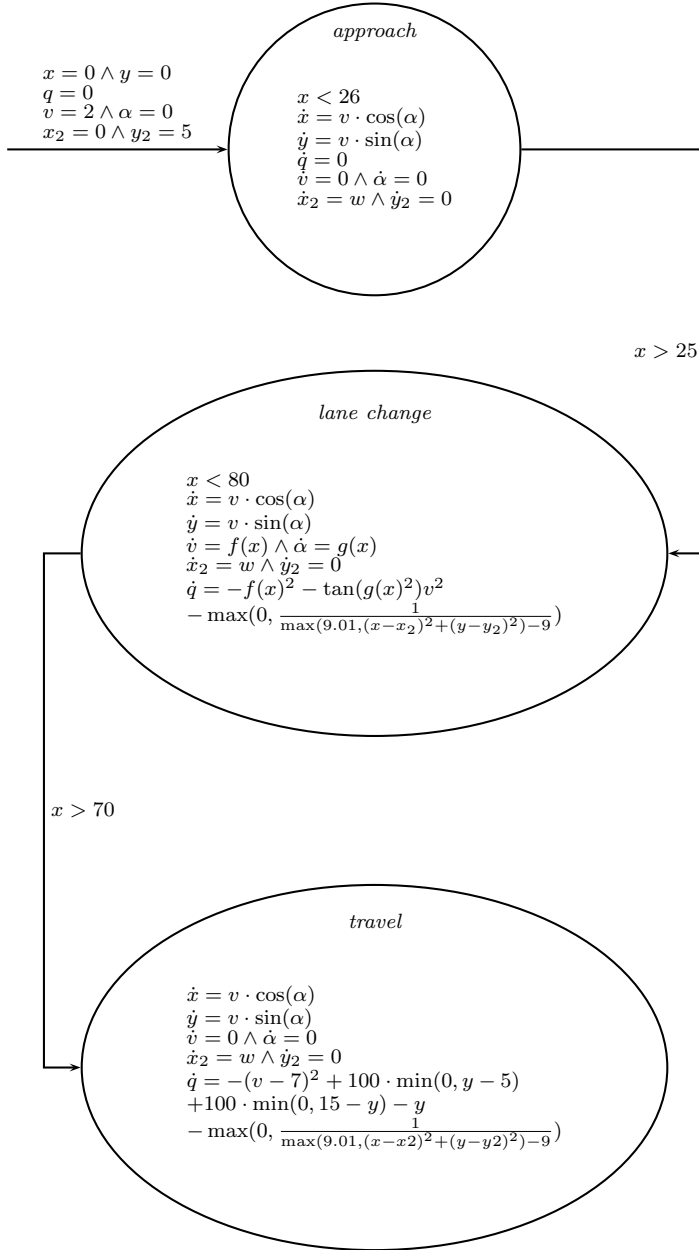


Fig. 6. A model of a human driver moving onto a freeway with another vehicle already on the freeway.  $x, y$ : position,  $v$ : velocity,  $\alpha$ : angle to freeway direction,  $x_2, y_2$ : position of other vehicle,  $w$ : velocity of other vehicle,  $f \in [0, 2]^{\mathbb{R}}$ : acceleration (optimised function),  $g \in [-0.1, 0.1]^{\mathbb{R}}$ : steering (optimised function),  $q$ : reinforcement value (measured at  $x = 140$ )

Figure 6 shows a hybrid model of a driver entering the freeway. We model the driver and the car as one unit, disregarding intermediate processes like moving the foot to press a pedal.

The model is based on the assumption that the driver starts at a given velocity  $v$  and has a desired travelling speed on the freeway. Moving onto the freeway he

tries to avoid high forces due to acceleration or trajectory change, to stay as far to the right as possible and, more than anything, avoid collisions with other vehicles.

The three states of the automaton represent three different phases of the manoeuvre. In *approach*, the car can not yet move off the acceleration lane and simply approaches the situation with a constant velocity. In *lane change*, the actual acceleration and lane changing takes place. In *travel* the car again keeps velocity and angle to the road constant. We use this state to evaluate whether the state at the end of the lane change behaviour is sufficiently stable to keep the car on the freeway for a meaningful amount of time.

The car has a position  $(x, y)$ , a current velocity  $v$ , and an angle to the lane  $\alpha$ . Those four variables are related in the obvious way. The variables  $v$  and  $\alpha$  can be controlled by the driver via the functions  $f$  and  $g$ , representing acceleration and steering respectively. It is those two functions we optimise towards maximal reinforcement value. This reinforcement value is represented by the value  $q$  into which we integrate all negative experiences possible while driving onto the freeway. Finally, we model another (dumb) car driving on the right lane of the freeway via the position  $(x_2, y_2)$  and its velocity  $w$  to make the situation more interesting.

The terms contributing to  $q$  in *lane change* are  $-f(x)^2$  for acceleration forces and  $-\tan(g(x)^2)v^2$  for angular forces as we hypothesise that drivers try to minimize forces affecting them. To ensure that the reinforcement value gets (strongly) negative when the car's position coincides with another car's position, we include  $-\max(0, \frac{1}{\max(9.01, (x-x_2)^2 + (y-y_2)^2) - 9})$ . In *travel* we keep this crash-related term and add  $-(v - 7)^2$  as a term for deviations from the desired target velocity, and  $100 \cdot \min(0, y - 5)$  resp.  $100 \cdot \min(0, 15 - y)$  to negatively value time spent outside the road on the right resp. left side, and finally  $-y$  as a small incentive to drive on the rightmost lane. Our model has a number of rather arbitrarily chosen parameters, representing the desired target velocity (7), the size of a car (9), the width of a lane (5), and the length of the acceleration lane (50). To achieve realistic results, these numbers would need to be matched to the geometry of the situation. As we are (in this paper) only interested in showing the feasibility of the approach however, we just chose a set of numbers which seemed roughly right and went with it.

In order to get visual feedback about our models, we wrote a numerical simulator for our hybrid automata models. It can either approximate the complete statespace of the automaton or execute (much faster) monte-carlo approximations to estimate the expected value of the reinforcement value. The monte-carlo approach in particular enables us to use genetic algorithms, which need to execute a model many times, to search for those values of the unknown functions maximising the expected reinforcement value.

The simulator takes a model in a Phaver-like [4] language and calculates an approximate solution of the differential equations using Euler's method and single precision floating point values. When optimising, the functions to be optimised are represented as a 100-element float vector specifying the function values at evenly spaced points in the interesting region. A linear interpolation is done for values

between these points. Using these function representations as genomes, a single cross-over genetic algorithm with a population size of 64 is run until the user stops the optimisation process. To evaluate the quality of a genome, ten monte-carlo runs of the simulation are performed and the resulting reinforcement are summed. We are well aware of various mathematical flaws in such a simplistic approach. As our results are nonetheless repeatable and numerically stable, we expect the application of standard numerical techniques to only improve the situation.

Figure 7 shows simulation results of the model, illustrating the behaviour of the driver after having optimised for the maximal possible reinforcement value. The upper panel shows how a car moves on the freeway with a comparably slow vehicle already on the road. In the simulation pictured in the second panel we raised the speed of car 2 just a little bit and, as a consequence, observed an abrupt qualitative shift in the behaviour of car 1 behaviour. Instead of accelerating and entering in front of car 2 it moves more slowly, enters *after* car 2 and overtakes it right after having entered. Note that in no part of the model we specified a behaviour like “overtaking” or “filtering into the stream of traffic”. The observed behaviour emerged naturally from the postulated reinforcement values and the optimization process.

## 5 Conclusion

The aim of this paper was to provide an outline of the potential of hybrid models in the domain of behavioural psychology and empirical science in general. We argued for the application of hybrid automata to model complex behaviour of living organisms. In addition to some general examples of hybrid systems within the context of animal behaviour we constructed a preliminary model of a human driver entering a freeway. The results of the simulation so far correspond well to the behaviour one would expect of a real driver. On the qualitative level, the model predicts “filtering into the traffic” as well as “overtaking” without ever having specified any such behaviours. Apparently, there seems to be no need for a cognitive process initiating these behaviours. Instead, a simple feedback learning mechanism like operant conditioning is sufficient to explain even complex driving manoeuvres. We take this as an illustrative example for the potential of a purely behavioural model of human drivers in terms of a hybrid system. We are currently working towards an empirical validation of the model in order to derive quantitative predictions for real life situations.

Ultimately, we would like to move on to modelling more complex situations involving substantially more than two cars. For this purpose we will extend the scope of hybrid automata to enable hierarchical modelling and multi-agent systems.

As shown in Section 3.1 animal behaviour is often structured in a hierarchical manner (compare Figure 2). Unfortunately, hybrid automata do not currently support hierarchical modelling – probably because it does not enhance formal expressiveness. It would, however, reduce cognitive load both on the modeller and on those trying to understand the model. As other hybrid approaches like hybrid

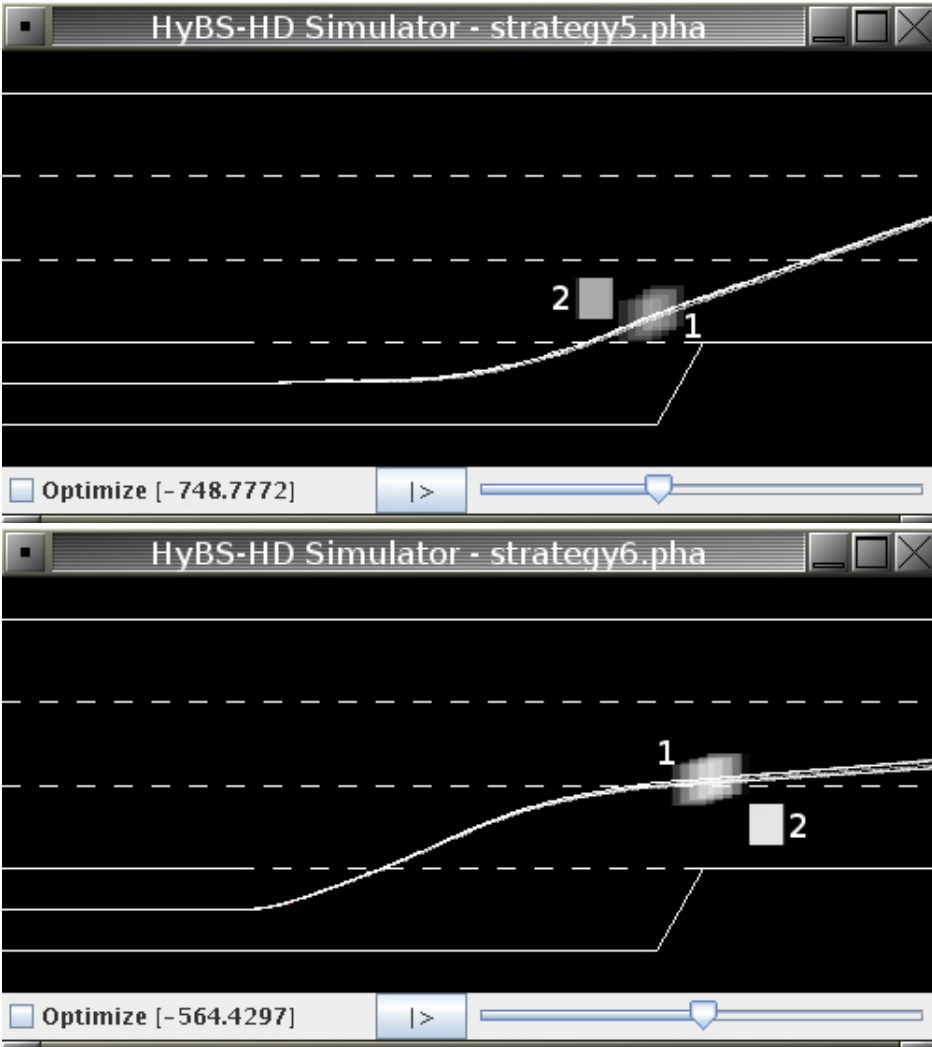


Fig. 7. Two simulation results with differing  $w$  (3.2 and 3.3 respectively). The blurring represents non-determinism of the model.

statecharts [8] already include hierarchical elements, we expect an integration of hierarchical elements into hybrid automata to be largely unproblematic. One interesting problem will be prioritisation of state transition between different levels of the hierarchy. While priorities are relatively clear-cut when modelling computer systems, organisms tend to have various kinds of fuzzy mechanisms for the initiation of a behavioural shift. In principle these mechanisms could be modelled by sufficiently sophisticated conditions on all transitions – as could hierarchy – but if we strive for simple models, a priority mechanism based on continuous variables might be better suited.

Many of our arguments in favour of hybrid automata also support using hybrid statecharts. So far however, we had no use of some of the statechart features, and decided to stay with the simpler formalism, the models of which could – if need

arises – be converted to hybrid statecharts easily enough.

When trying to model multi-agent systems, an explicit description of all interactions between all possible pairs of agents, as we did in the automaton in Figure 6, becomes unwieldy. Rather, it must become possible to describe interactions between agents without explicitly naming them. To address communication partners in such a scheme, agents must also be enabled to “search for” other agents using conditions based on real-valued variables, e.g. distance.

Once these extensions have been implemented in our modelling framework, it should be possible to extend the scope of our approach to a great variety of real life situations, not only in the domain of driver simulation.

## References

- [1] <http://imost.informatik.uni-oldenburg.de/index.html>.
- [2] Anderson, J., *ACT: A Simple Theory of Complex Cognition* John R. Anderson, Cognitive modeling (2002), p. 49.
- [3] Canale, R., *An analysis of models describing predator-prey interaction*, Biotechnology and Bioengineering **12** (1970), pp. 353–378.
- [4] Frehse, G., *Phaver: Algorithmic verification of hybrid systems past hytech* (2005), pp. 258–273.
- [5] Henzinger, T. A., *The theory of hybrid automata*, in: *LICS '96: Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science* (1996), p. 278.
- [6] Henzinger, T. A., P.-H. Ho and H. Wong-toi, *Hytech: A model checker for hybrid systems*, Software Tools for Technology Transfer **1** (1997), pp. 460–463.
- [7] Immelmann, K., K. Scherer and C. Vogel, “Psychobiologie: Grundlagen des Verhaltens,” Fischer, 1988.
- [8] Kesten, Y. and A. Pnueli, *Timed and hybrid statecharts and their textual representation*, in: J. Vytupil, editor, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, Lecture Notes in Computer Science **571**, Springer Berlin / Heidelberg, 1991 pp. 591–620, 10.1007/3-540-55092-5\_32. URL [http://dx.doi.org/10.1007/3-540-55092-5\\_32](http://dx.doi.org/10.1007/3-540-55092-5_32)
- [9] Krebs, J. and N. Davies, “An introduction to behavioural ecology,” Wiley-Blackwell, 1993.
- [10] Möbus, C., M. Eilers, H. Garbe and M. Zilinski, *Probabilistic, and empirical grounded modeling of agents in partial cooperative (traffic) scenarios*, in: *Conference Proceedings, HCI 2009, Digital Human Modeling, San Diego*, Lecture Notes in Computer Science (LNCS) and Lecture Notes in Artificial Intelligence (LNAI), Conference Proceedings, HCI 2009 (2009), pp. 423–432.
- [11] Salvucci, D., E. Boer and A. Liu, *Toward an integrated model of driver behavior in cognitive architecture*, Transportation Research Record: Journal of the Transportation Research Board **1779** (2001), pp. 9–16.