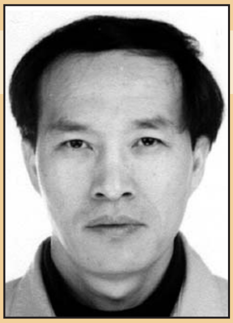

Editor: Fei-Yue Wang
University of Arizona and
Chinese Academy of Science
feiyue@sie.arizona.edu

Making Driver Modeling Attractive

Antonio Pellecchia, Christian Igel, Johann Edelbrunner, and Gregor Schöner,
Institut für Neuroinformatik, Ruhr Universität Bochum

Automobile industry and academic researchers expend considerable effort to develop driver assistance systems. DASs are aware of certain driving situations and support drivers through information, warnings, or even intervention. Many DAS applications are safety oriented, such as lane departure warning systems. Some are comfort oriented, such as automated parking assistants. Still others are designed to alleviate the driver's attentional load—for example, traffic signal detectors or the already deployed intelligent cruise control systems. These systems must adapt to varying traffic situations to achieve the goal of enhanced safety and comfort. Moreover, DAS human-machine interfaces must support careful communication with potentially taxed drivers.

Driver models support DAS design in several ways. First, they let researchers analyze system performance in a variety of driving situations. They also help optimize parameter values for certain performance measures,[1] such as the degree to which a desired speed is maintained. More sophisticated yet, driver models can form the basis for classifying driver types, such as aggressive versus defensive. DAS applications can use these classifications to activate different parameter value sets that adjust warning thresholds appropriately for different driver types.

Driver models might also help classify traffic scenarios, such as fluid versus congested traffic. Analyzing the scenario and the driver's behavior, DASs could autonomously adapt parameter settings to best respond to the situation. Eventually, onboard operation of driver models could enhance scene analysis and interpretation by predicting the behavior of not only the car in which they are installed but also other cars in the vicinity.[2] As the driver model more fully accounts for the traffic scene, DASs increase their power to compare observed with predicted behaviors,[3,4] and thereby detect anomalies.

We have applied attractor dynamics to modeling driving behaviors. Originally developed to generate behavior in autonomous robots, attractor dynamics encode behavioral policies for modeling a driver with meaningful parameters that support optimization by direct policy search. We used a powerful evolutionary algorithm to vary the parameters in order to generate three driver models that capture the behavioral patterns of different driver types.

Editor's Perspective

This article is based on the best student poster paper presented at the 2004 IEEE Intelligent Vehicle Symposium in Italy. As a judge for this competition, I was "attracted" by the innovative approach of using attractor dynamics and evolutionary algorithms to encode behavioral policies for modeling drivers. This approach and the use of applied cognitive methods can play an important role in computational experiments for driving safety studies and, more generally, in transportation systems research.

—Fei-Yue Wang

Tactical decision complexities

Our work focuses on modeling the tactical level of driving decisions, such as when to brake and whether to accelerate and pass another vehicle. Such decisions are based on local, instantaneously available environmental information about the road and other cars in the same or an adjacent lane.

The tactical level poses considerable modeling difficulty. First, tactical driver modeling involves both continuously valued variables, such as acceleration, and discretely valued decision variables, such as whether to change lanes or not. The ensemble of relevant local driving scenarios quickly acquires considerable combinatorial complexity with only moderate increases in the planning horizon. Second, ground truth data—that is, measurements of actual driver behavior under realistic conditions—is difficult to obtain, precisely because real driving situations can't manipulate local traffic conditions.

1541-1672/05/$20.00 © 2005 IEEE
Published by the IEEE Computer Society

Essentially, tactical driver modeling must satisfy multiple constraints under the additional constraint of uncertain, time-varying information. Constraints include traffic rules, desired speeds, safe distances from other cars, and coherent behavior—for example, avoiding unnecessary lane changes.

We don't address the strategic level of driver modeling, which involves understanding how drivers achieve goals, such as taking a particular exit. We address only to a limited extent the control level at which DASs generate a control signal's time course, such as level of acceleration. However, the optimization methods we discuss are potentially applicable at both the strategic and the control levels.

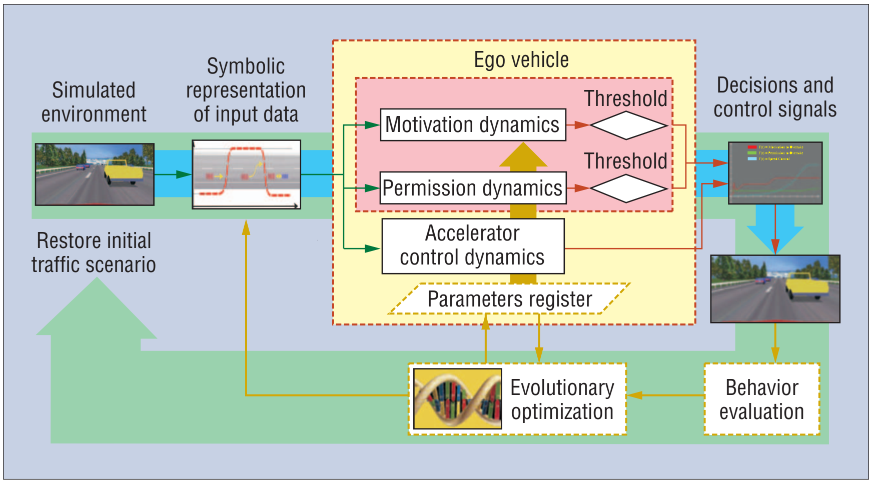

Figure 1. Driver-modeling system structure. The ego vehicle's driver model consists of three levels—motivation, permission, and accelerator control—each instantiated by an attractor dynamics.

Methods

Our approach to driver modeling represents control behaviors directly by their associated continuous control variables, such as driving speed. It represents discrete behavioral decisions through neuronal activation variables, which are somewhat like probabilities. The attractor dynamics act as a low-pass filter, enabling linkage to noisy and time-varying inputs, while also stabilizing decisions. We optimize the model parameters based on simulations.

This approach views driver modeling as a reinforcement-learning problem, solved through evolutionary direct policy search. We evolve models of different driver classes by varying the learning goals, such as desired speed or desired *time-to-contact*—that is, the time interval for taking an action before a potential collision would occur.

Figure 1 illustrates our driver-modeling system's general structure. The large green arrow symbolizes a system training cycle. Yellow directed segments illustrate the information flow during evolutionary optimization. Red segments indicate decision and control information. The blue arrow symbolizes the driver model's interaction with the simulated environment in the absence of adaptation.

The modeled behaviors are cruise control within a lane (control behavior) and a lane change (discrete behavioral decision). The decision to change lanes is based on two activation variables, motivation and permission (pink box). Motivation to pass another vehicle achieves high activation levels, for instance, when cruising speed is too low. Permission activation levels will increase with increases in the contiguous free space in the left lane. Cruise control is achieved by generating planned acceleration values.

The system combines the three types of behavioral variables—motivation, permission, and acceleration—and makes a decision based on which acceleration and lane-change commands are transmitted to the vehicle model.

Simulator

To enable learning and optimization, the system must generate and evaluate a wide variety of driving behaviors. This is possible only in simulation, of course. At Ruhr Universität Bochum's Institut für Neuroinformatik, we've developed a highway traffic simulator.[5] The simulator is a modular platform that provides four basic functions:

- a road structure model;
- a traffic model, including a simplified behavioral model for all vehicles except the vehicle whose behavior is being optimized, which we call the *ego vehicle*;
- models of all vehicles' sensors; and
- a model of the ego vehicle's driving behavior, which the learning algorithm optimizes.

For optimization, we fix a lane-based road model and a traffic scenario that includes a given number of cars and a speed distribution. Each simulator run generates a driving episode by initiating all behavioral models with a new random sampling of the traffic model. The sensor model continuously obtains readings for the distance and relative velocity of vehicles in the ego vehicle's vicinity. The simulator assigns these to one of six cells around each vehicle.

For the *i*th cell, $i \in \{1, \ldots, 6\}$, the risk function

$$\rho_i = \omega_v \cdot vel + \omega_{\Delta v} \cdot \Delta vel_i + \frac{\omega_x}{dist_i} \tag{1}$$

combines a car's velocity, *vel*, relative velocity, Δ*vel*, and distance, *dist*, from the vehicle to a car in the cell into a single scalar, each weighted with a dedicated parameter ω_v, ω_x, $\omega_{\Delta v}$, and ω_{vel}, which is varied during optimization. The simplified behavioral model is purely reactive to the risk value thresholds recorded for all cells around that vehicle.

Driver model

The ego vehicle's driver model consists of three levels, shown in Figure 1, each instantiated by an attractor dynamics of a relevant variable *F*.[6] The following dynamical system defines all three levels:

$$\tau \frac{dF(t)}{dt} = -s \cdot F(t) + Y_{\text{off}} \cdot \text{sign}[F(t)] + \sum_i w_i \cdot input_i(t) - bias \tag{2}$$

where in each case, *F*(*t*) is the acceleration or activation level of the decision-making state encoded in the motivation or permission level. The first term generates stability, as illustrated in Figure 2. The negative slope, determined by the model parameter *s*, makes *attractors* of the fixed points, or zeros of

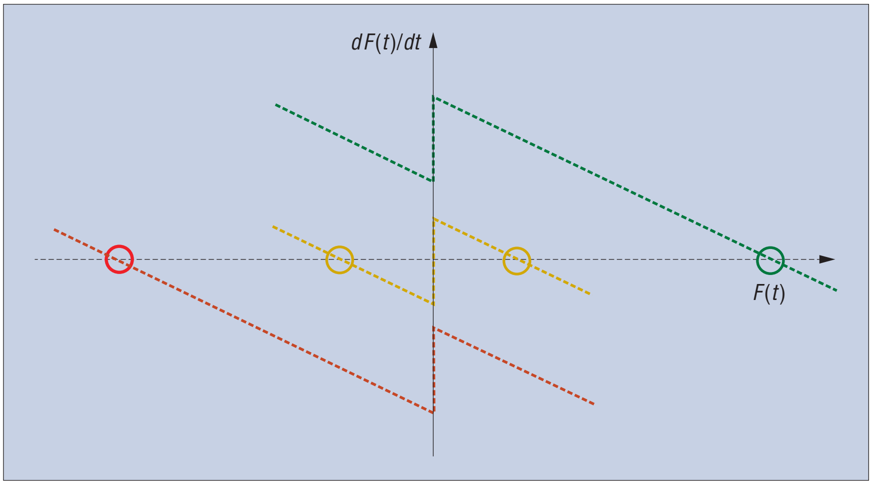

Figure 2. Behavioral dynamics. Equation 2 is plotted for three levels of total input: red, yellow, and green for low, intermediate, and high levels, respectively. For neuronal variables, negative values represent "no" decisions; positive values represent "yes" decisions.

the rate of change. The attractors are time-constant solutions to which neighboring values are attracted. For neuronal variables, negative values represent "no" decisions and positive values represent "yes" decisions. The sign function, scaled by the model parameter Y_{off}, generates three cases depending on the total system input. A single attractor exists at low and high input levels, while the system is bistable at intermediate levels, which helps to stabilize decisions against fluctuating sensory input.

Other parameters are the time constant, τ, and the relative weights, w_i, of different inputs such as the opening size on the left lane or the deviation from desired speed (four inputs each for motivation and permission, one input for acceleration). The model parameter *bias* sets the default values of the state variables in the absence of inputs.

Acceleration dynamics receive input from the risk function, ρ, which is designed to attain the desired speed while respecting a minimal time-to-contact. Input from the motivation and permission layers for a lane change generates additional acceleration just before a passing maneuver. A sigmoidal function transforms the acceleration state variable into actual acceleration values handed to the vehicle model. The sigmoidal function limits the acceleration range to admissible values and smoothes acceleration.

Permission continuously represents the extent to which conditions for initiating a passing maneuver are fulfilled. Input to the permission dynamics encodes the existence of a space-time gap on the lane to the left of the ego vehicle that makes lane change possible. Analogously, motivation continuously represents the extent to which conditions exist that make a passing maneuver desirable. Input to the motivation dynamics encodes the distance to the car in the cell in front of the ego vehicle, the time-to-contact, as well as the deviation from the desired speed.

Both variables must be positive to initiate a passing maneuver. Because zero separates the two possible attractors in the bistable regime, the two variables do not change sign at a high frequency. When the behavioral model detects a sign change to positive, the simulator generates a trajectory that transitions the vehicle to the left lane.

Evolutionary adaptation

Our representation of behavioral policy through dynamical systems supports gradual optimization by direct policy search. To do

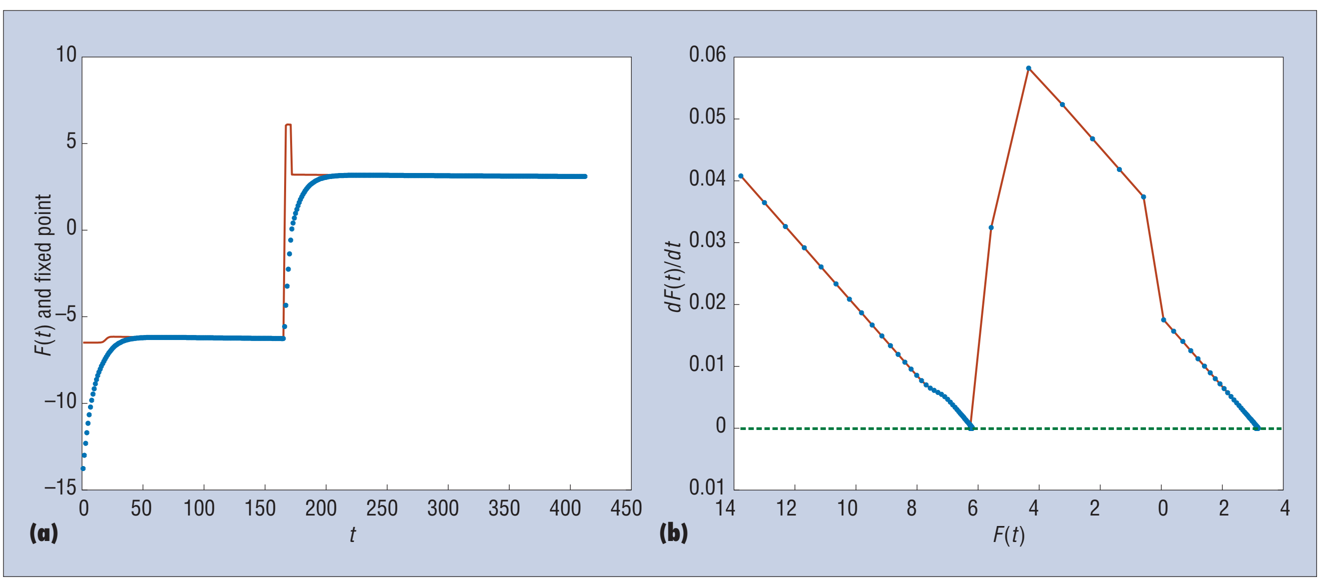

Figure 3. Behavioral dynamics solutions. (a) An activation variable's time course during simulated driving of the evolved driver model is plotted in blue, together with the value of the associated attractor (red). (b) The associated rate of change is plotted against the activation variable.

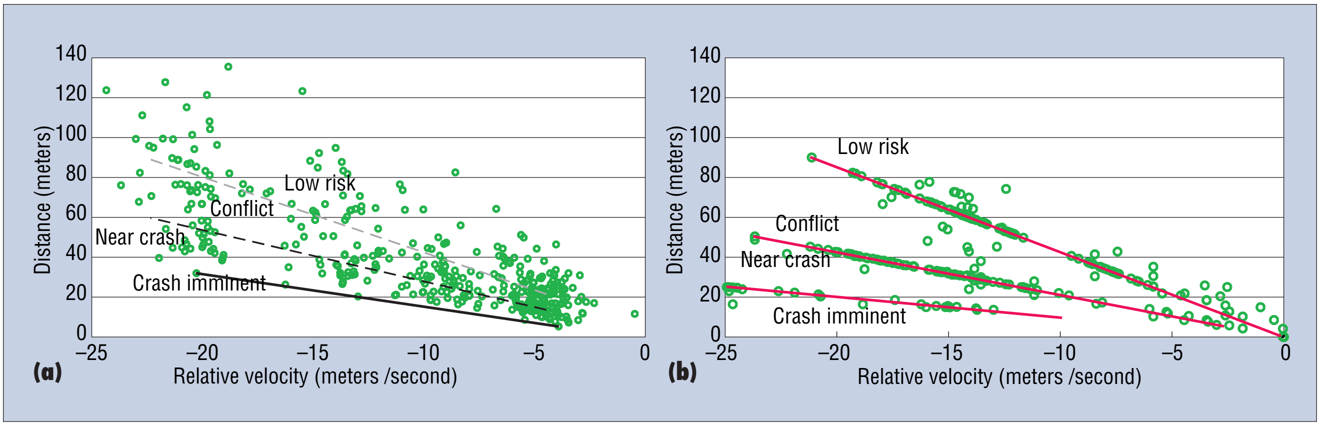

Figure 4. Comparison of driver model to human drivers. (a) The values of relative speed and distance at which test drivers, in a real highway scenario, decided to initiate a passing maneuver; (b) the same measures, obtained in a simulated environment, as a result of training our agent's decision system.

this, we employ an evolutionary algorithm,[7] where the fitness reflects the driver model's performance. The parameters subject to optimization include weights, time constants, amplitudes and biases from Equation 2 of the three dynamics employed, and the slopes and offsets of the sigmoids used as input filters.

We optimized the parameters by using the powerful Covariance Matrix Adaptation Evolutionary Strategy.[8] The CMA-ES automatically adapts the mutation distribution's covariance matrix and thereby considers correlations and scaling of the objective parameters. We trained the acceleration dynamics for autonomous cruise control—that is, for maintaining a desired speed while respecting the minimal time to contact. We trained the permission and motivation dynamics to reproduce measured, real driver behaviors and to generate correct driving behavior. We constructed logical conditions to describe the fitness for these three levels. Such conditions detected violations of the desiderata—for instance, permission to pass when the space-time gap on the left lane was insufficient or motivation to pass when the speed was close to desired.

Results

The evolved driving behavior fulfills the constraints while achieving a good approximation of the desired speed and avoiding high risk values.

Analysis of the dynamical systems that emerge from the optimization reveals that they do, in fact, closely track the attractor states, as Figure 3 shows. Instabilities induce behavioral changes, such as switching from deceleration to acceleration, which always pass through bistable regimes by construction, thus stabilizing the decision. In Figure 3a, an activation variable's time course during the evolved driver model's simulated driving is plotted in blue, together with the value of the associated attractor (red).The variable tracks the attractor when the attractor location jumps from negative to positive values near 170 time units. In Figure 3b, the associated rate of change is plotted against the variable. This illustrates the approach to the attractor at negative values and then the fast relaxation to the shifted attractor at positive values. The blue points are samples obtained from the simulator, connected by the red line.

Figure 4 compares the driver model to the behavior of real human drivers. David Smith and his colleagues asked participants to drive on a test track and then in a driving simulator using one of three strategies: cautious, normal, or aggressive.[9] They observed the points at which drivers changed lanes in the context of passing maneuvers and plotted the distance to the front car against the relative velocity. All three conditions showed events scattered along a straight line that corresponds to a particular time-to-contact. With increasing risk taking, the line's slope decreased, corresponding to a decreasing time-to-contact at lane change. By imposing different desired speeds and different thresholds in the various contributions to the fitness functions, we were able to model the three kinds of

Antonio Pellecchia is a doctoral student at the Institut für Neuroinformatik at Ruhr-Universität Bochum. His research interests focus on autonomous agents. He is a member of the IEEE Intelligent Transportation Systems as well as the Systems, Man, and Cybernetics Societies. Contact him at antonio.pellecchia@neuroinformatik.rub.de.

Christian Igel is junior professor for optimization of adaptive systems at the Institut für Neuroinformatik at Ruhr-Universität Bochum. His research focuses on biological and technical aspects of neural and evolutionary information processing. He received his PhD from the Technical Faculty of the University of Bielefeld, Germany. He's a senior member of the IEEE. Contact him at christian.igel@neuroinformatik.rub.de.

Johann Edelbrunner is a doctoral student at the Institut für Neuroinformatik at Ruhr-Universität Bochum and is CEO of NISYS. His research interests include image processing, object recognition, behavior planning, and statistical signal processing. He received his MS in physics from the University of Graz, Austria. Contact him at johann.edelbrunner@neuroinformatik.rub. de.

Gregor Schöner is a professor and director of the Institut für Neuroinformatik at Ruhr-Universität Bochum. He received his PhD in theoretical physics from Universität Stuttgart. Contact him at gregor.schoener@neuroinformatik.rub.de.

drivers. In other words, we obtained driving styles by defining different targets for the evolutionary optimization of the driver model.

Our driving simulator is a powerful tool for modeling driving behavior, both in setting and testing different driving strategies and in varying optimization targets. The behavioral model makes it possible to represent different driver types. DAS developers can apply such experimental settings to driver models to optimize their systems.

In our lab, we've developed a sophisticated computer vision system as the perceptual module of vision-based DASs.[10] This module uses a large number of parameters including thresholds and filters. To tune the parameters for optimal performance, we are using driver models in our simulator and integrating the visual-sensor module through a visual-sensor model. We've employed evolutionary computation to adapt the vision system parameters, optimizing fitness functions based on the ground truth available within the simulator. We envision support for driver classifications based on on-board data obtained from the vision system or other sensors and on recorded driver actions. ■

References

1. R. Sukthankar, S. Baluja, and J. Hancock, "Multiple Adaptive Agents for Tactical Driving," *Applied Intelligence*, vol. 9, no. 1, 1998, pp. 7–23.
2. D. Ehmanns and A. Hochstädter, "Driver-Model of Lane Change Maneuvers," paper presented at 7th World Congress on Intelligent Transportation Systems, 2000.
3. I. Dagli, M. Brost, and G. Breuel, "Action Recognition and Prediction for Driver Assistance Systems Using Dynamic Belief Networks," LNAI 2592, Springer-Verlag, 2002, pp. 179–194.
4. A. Pentland and A. Liu, "Modeling and Prediction of Human Behavior," *Neural Computation*, vol. 11, 1999, pp. 229–242.
5. A. Pellecchia and J. Edelbrunner, "Modeling Real Driving Behaviors through Attractor Dynamics for Motivation," *Proc. 2004 IEEE Intelligent Vehicles Symp.*, IEEE Press, 2004, pp. 121–126.
6. G. Schöner and M. Dose, "A Dynamical Systems Approach to Task-Level System Integration Used to Plan and Control Autonomous Vehicle Motion," *Robotics and Autonomous Systems*, vol. 10, 1992, pp. 253–267.
7. C. Igel, "Neuroevolution for Reinforcement Learning Using Evolution Strategies," *Proc. IEEE Congress Evolutionary Computation*, IEEE Press, 2003, pp. 2588–2595.
8. N. Hansen and A. Ostermeier, "Completely Derandomized Self-Adaptation in Evolution Strategies," *Evolutionary Computation*, vol. 9, no. 2, 2001, pp. 159–195.
9. D.L. Smith et al., *Feasibility of Modeling Lane-Change Performance*, tech. report 2003-01-0280, Nat'l Highway Transportation Safety Administration, 2003.
10. T. Bücher et al., "Image Processing and Behavior Planning for Intelligent Vehicles," *IEEE Trans. Industrial Electronics*, vol. 50, no. 1, 2003, pp. 62–75.