Anticipatory Driving for a Robot-Car based on a Two-Level Control and Supervised Learning

Irene Markelić, Tomas Kulvicius, and Florentin Wörgötter

Bernstein Center for Computational Neuroscience, University of G¨ottingen,
Bunsenstrasse. 10, 37073 G¨ottingen, Germany
{irene,tomas,worgott}@bccn.de
http://www.bccn-goettingen.de

Key words: anticipatory behavior, reactive behavior, learning from observation

Introduction: In this paper we describe how we achieve anticipatory vehicle control by imitating human behavior. The main novelty is the combination of a reactive and a planning unit for improved overall performance and enabling self-learning of the system. Both controllers are based on the assumption that similar situations call for similar control actions and are obtained via supervised learning. This work deals exclusively with the problem of road following, i.e. there are no other participants on the road and no further complications like intersections, or else. Also we consider only steering control, keeping the velocity of the system fixed.
The approaches for vehicle control can be distinguished into three different classes: The first involves a physical modeling of a system where many sensors contribute to the creation of a world model in 2-d (top view) or even three dimensions. Usually GPS and/or SLAM algorithms are used for determining position and orientation of the vehicle from which then a path and the requried control signals can be calculated, comp. [1]. The second class is based on machine learning, especially imitation/supervised learning, or reward based learning. In the case of supervised learning, the underlying assumption is that the vehicle is a complex system that is difficult to model since knowledge of various required parameters is often not obtainable. Instead, a good controller, i.e. a human, is already available. Thus, the goal is to find a transfer function that relates the input to the human to its generated control output, see e.g. [2]. The third class is an area from psychophysics where human strategies are analyzed and tried to be understood by modeling, as such it is not focused on the application, comp. [3]. This taxonomy allows us to position this work in the second class of approaches to vehicle control. In the following, we will motivate our two-level architecture, elaborate our experimental setup, explain the training procedure and present the results along with a short discussion of the pros and cons of the applied technique.
Experimental Setup: For the benefit of greater flexibility, experiments are carried out on a car-like robot (VolksBot [4]) equipped with cameras, instead of a real car, comp. Fig. 1A. The laboratory setup simulates a street environment, where the driver can control the robot from a special station, see Fig. 1B. Here,

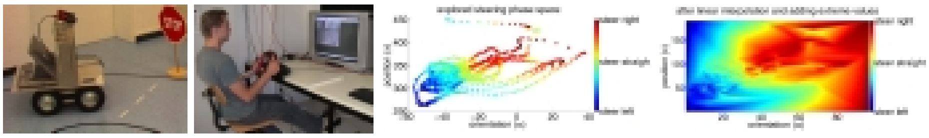

Fig. 1. From left to right: A) A car-like robot. B) Control station. C) Steering signal plotted against α and x. D) Filled steering signal space after linear interpolation.

he can see "through the robot's eyes" by means of a TV on which we display the robot's camera output and he can use the robot's actuators by means of a steering wheel and pedal set. The communication between human control output and robot sensory input is realized via a client-server architecture. The server, a laptop placed on the robot, is connected to its motors and cameras. In a cycle-wise fashion it acquires a camera image, sends it to the TV, and then waits for a control input from the client, a desktop computer connected to the steering wheel and pedal set. The client sends a message containing the control signal generated by the human to the server, which in turn sends it to the motors of the robot. Thus, for every incoming image there is one control output. The "street" consists simply of black tape on the floor.

Fig. 2. Example of the planner operating mode. The observed street is compared to the database entries and the assigned steering sequence of the best match is depicted on the right.

The Controller: The motivation for using a two-level control architecture stems from [3]. Here, we learn a simple reactive controller (RC), and a planning unit, short planner, from the human supervisor, where each controller alone has certain properties that are complementing each other when combined.
The planner generates a sequence of steering actions as response to an observed road trajectory. The benefit of having such a plan, is that the system can use it as a fallback strategy, in the case of an open-loop situation, e.g. when the street cannot be detected. Also it relieves the system from having to process all sensory input in every cycle, since it can use the forecasted actions to a certain degree, saving processing time and leading to faster ouput. A further advantage is, that the generated output can effectively be smoothed, which is difficult when only one control output per timestep is produced. The generated plan is not required to be very exact, but roughly capturing the future course. Also it is not necessary

to return a plan every cycle.
For finetuning we rely on the RC. Every cycle it generates exactly one control signal which is reliable, and can correct the planner if necessary. In contrast to the planner, the RC does not take all the observed future road course into account, but only a short part of the street, thus, it does not require as much processing time.
Training the Controller: Both controller map parameters from visual input to a steering signal output. The input parameters are derived from the extracted street from the camera images. For our use, knowledge of one lane boundary is sufficient, and without any further reason we chose the right one. Since it is not crucial for the rest of the paper we do not explain the extraction process here, only noting that the result of it is a vector of pixels containing the right lane boundary. The output steering signal is a numerical value in the range [-128, 128], where negative values correspond to left, zero to straight, and positive to right steering. For learning the RC, information about the near road trajectory is required, and about the position and orientation of the vehicle with respect to the street lane. The lateral offset of the car from the lane tells about the vehicle's position. Therefore, we use the x-coordinate of the right lane boundary at the bottom of the image. To learn about the robot's orientation, we calculate the angle formed by the tangent of the street lane boundary at the bottom of the image and the horizontal. We call these parameters x and α. We then train the robot by navigating it through the track in our lab. After 5 laps we plot the steering value from the human against the two parameters as shown in Fig. 1C and then linearly interpolate it, the result is shown in Fig. 1D.

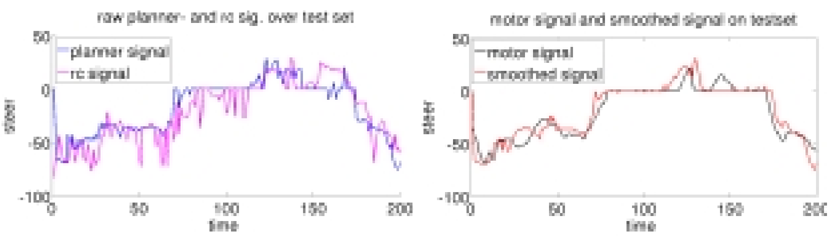

Fig. 3. A) Signals from planner and RC over testset. B) Combined signal compared to human steering signal, here denoted as motor signal.

The planner, for generating a sequence of actions, must integrate more information of the upcoming road course. We achieve this, by building a database in wich we store street-action pairs, based on the human actions. For data reduction, the extracted street is first polygonized and the resulting corner points are written to a vector which is used as street descriptor. On these vectors, we define a metric, based on the weighted euclidean distance between the corner points. The weights are chosen such, that the bottom parts of the street, which describe the near future to the vehicle, are forced to be similar. The corner points that remain closer to the end of the time horizon are allowed to differ more. The controller than works as follows: For each observed situation it searches the database built from previous experience. If a tolerable match is available it returns the assigned steering sequence to that match. In Fig. 2 we show an example of such a retrieval.

Results: The performance of the planner is shown in Fig. 3A, where we plotted its ouput in comparison to that of the RC on a data testset. The "real" signal from the human on this testset is given in Fig. 3B. As can be seen the RC and the planner capture the human signal relatively well, where the RC signal oscillates around it, resulting in a jerky control of the vehicle. The planner, however, is smoother, but sometimes, e.g. between timestep 160 and 170, deviates from the human signal in a non-acceptable way. As claimed before, theses two controller outputs can be combined to improve overall performance. This combination is done by weighting their influence in such a way, that, in case of a bad match found in the database of the planner, we rely more on the RC signal, and in case of a very good match provided from the planner we rely more on the latter. The result is smoother and closer to the human control as shown in Fig. 3B. Furthermore, open-loop control can be achieved. In case of a well constructed database, the planner output can be used for several timesteps, not requiring a continuous processing of the entire image. Another advantage is, that the robot can now follow tracks that it has never seen before. This was tested by training it only in one direction of the track, and then turning it around. The incoming street observations are mostly not contained in the database, but the robot can still stay on track by making heavier use of the RC signal. This, although not implemented yet, can be used for self-learning. This can be done the following way. To a street observation, that is not contained in the database yet, we assign the sequence of actions that is subsequently generated by the RC. This signal can be smoothed, and, together with the street observation, added as a new entry to the databse.

Discussion: In the realm of driving the explained two-level strategy appears to be a promising way from plain reactive/reflex to planned/anticipatory behavior. It is to find out, if it can be extended to other applications, e.g. legged locomotion. A potential disadvantage is the memory intensity of the planning unit, which might be improved by usage of other methods, more advanced than plain storage and retrieval. Also, it is desirable to introduce some kind of generalization skill to the planner.

References

1. Thrun, S. et al.: Stanley: The Robot that won the DARPA Grand Challenge. Journal of Field Robotics, vol. 23(9), pp. 661-692, (2006)
2. Pomerleau, D.A.: Neural Network Vision for Robot Driving. The Handbook of Brain Theory and Neural Networks. M. Arbib, MIT Press (1995)
3. Donges, E.: A Two-Level Model of Driver Steering Behavior. Human Factors, vol. 20, pp. 691-707, (1978)
4. VolksBot, `http://www.volksbot.de`