Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

Behavior Decision and Path Planning for Cognitive Vehicles using Behavior Networks

Joachim Schröder, Markus Hoffmann, Marius Zöllner and Rüdiger Dillmann
– intended for Special Session "Cooperative Cognitive Automobiles" –

***Abstract*— Behavior decision in human brains is a process which involves different regions of the brain, each one considering specific driving aspects. These regions interact with each other and are able to stimulate or suppress other areas. The executed behavior is a result of a fusion process and depends on the structure of the network and the motivations of individual behaviors.**

At the Institute of Interactive Diagnosis- and Servicesystems in Karlsruhe, a behavior based architecture, called *Behavior Network*, was developed and is used since several years to control walking machines. The used approach of coupling and modularization of behaviors on different layers with reactive or deliberative character can be transferred to cognitive vehicles, since the complextiy of behavior interaction is also given here.

This biologically-oriented method seems adequate to derive behaviors for driving and perception, which are often combinations of several sub-behaviors with different motivations (road-following, lane-keeping, speed control, collision avoidance). The attempt to control the vehicle with human-like behaviors has the advantage of good traceability of the executed manoeuvres, as well as the use of humans as teachers to parameterize behaviors.

This paper describes how to implement a behavior network for road traffic with basic behaviors to execute safe driving manoeuvres, and how to ensure safety in certain situations by using fusion nodes.

I. INTRODUCTION

A. Overall Project Description

This work is part of the collaborative research centre on *Cognitive Automobiles* [4], a project started January 2006 and financed by the German Research Foundation (DFG). The University of Karlsruhe, the TU Munich, the Fraunhofer Gesellschaft (IITB in Karlsruhe) as well as the Universitaet der Bundeswehr Munich are working together with the objective to develop an autonomous vehicle which is able to gather data from the environment, understand traffic scenarios and perform independent or cooperative driving manoeuvres.

This paper describes part of the tasks to be done in the subproject *Cognitive Behavior Decision and Path Planning*, which is responsible for selecting and executing behaviors in order to generate input for the underlying control. An important issue in this subproject is the consideration of feedback mechanisms, which should lay the basis for future machine learning.

B. Behavior Networks

Behavior networks were developed with the goal to obtain a modular and robust control architecture for walking machines [1], [2]. The idea was to combine a classical hierarchical control approach with conclusions of biology. It is known, that certain movements or impulses from humans or animals stimulate always the same neurons, and an activity of a certain region has influence on other areas. This lead to the idea of paying special interest to the interactions of behaviors, and to model them separately. As a result, a behavior unit with several connections for user and status data arose, as can be seen in Figure 1.

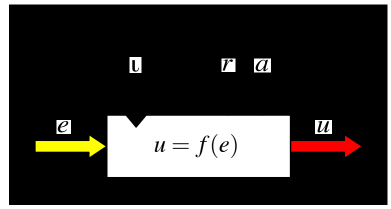

Fig. 1. Single behavior module with interface: Motivation ι to stimulate the generation of output $f(e)$ and Reflection r and Activity a to express satisfaction and effort of the behavior.

The transfer function f contains the basic character of the behavior. In a first step, the result $f(e)$ is computed, and modified in a second step according to the motivation ι. Every behavior has defined an internal goal which it is trying to reach. The reflection r describes to which extent the actual state differs from the desired state. The rating is generated independently from the activity of the behavior. This aspect of *virtual sensors* is a very important issue within the architecture, and different to other behavior-based architectures as [3], [5], [6].

Through the activity a, the behavior shows the actual effort in reaching the defined goal. This information can be used for weighted fusion of different behavior outputs. Standardization of status data connectors ι, r and a to [0;1] allows an abstraction from internal physical values, and qualifies the connections for coupling of behaviors.

C. Short-Term and Long-Term Objectives

At the beginning of the project, interfaces to other software components have to be specified and a first set of basic behaviors needs to be implemented. These basic behaviors are levelled on the lowest hierarchy of the network and represent the interface to the control part. The first behavior executed on the real vehicle should be a higher levelled (so-called strategic) behavior as *Follow Street*. Thus, all other tactical behaviors (*Keep Lane, Set Speed*) which are necessary to accomplish this task need to be specified and implemented.

More strategic behaviors as e.g. *Turn into Street* will follow to be able to perform a wide variety of driving manoeuvres and to complete the network.

At a later stage of the project, driving manoeuvres will be evaluated with the help of behavior network reflecions, that describe each one how satisfied the particular behavior was during the process. By combining this internal information with real sensory data, the authors think that it will be possible to draw useful conclusions regarding *Safety* (for own car or other paricipants) *Efficiency* or *Comfort*. This information can be either used to help in the development process, or for machine learning and adaption of behaviors.

II. System Architecture

A. Classification within project

As mentioned in the introduction, the classification of the behavior network is in-between the scene interpretion and the vehicle control, as illustrated in Figure 2. Looking at the overall process, objects in the traffic scene like lane markings, obstacles or other participants are detected with vision systems and LIDAR, classified and the result will be handed to the situation interpretion. This module generates information about object relations and adds logic features, considering the own intention according to the navigation system. After this step, it will be possible e.g. to distinguish between a lane with oncoming traffic or a lane which should be followed towards the destination.

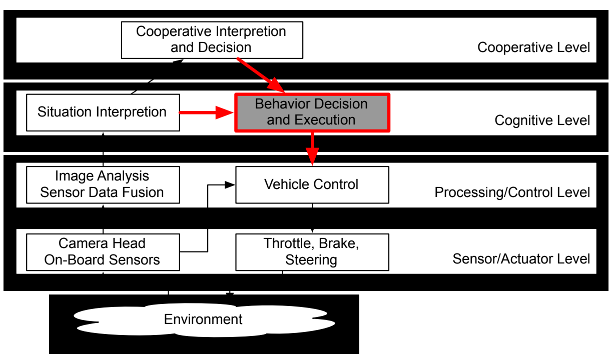

Fig. 2. Overview of the System Architecture

The results of this process are being transferred to the behavior module, which is split into a decisive and an executive part. Strategic behaviors on the cognitive level know how to perform a driving manoeuvre and have to decide about motivation of underlying tactical behaviors, fitting best for the actual situation. Behaviors of tactical or reactive character are executed according to their included scheme and have only limited knowledge about the overall situation, but are directly supplied by the database with newest sensor informations. A detailed description of the behavior network and implemented behaviors will follow in Section III-C.

The lower set of reactive behaviors is generating input values for the control part, which is carried out in conventional way with single or dual track model and separate controllers for steering, throttle and brake. To perform cooperative manoeuvres in case of emergency, an interface to a cooperative decision unit provides the ability to overwrite individual decisions.

B. Interfaces

In order to evaluate functioning of each software component, it must be possible to generate input data individually, and to have a suitable way of displaying the result, e.g. through linguistic expressions or geometric representations.

This issue played a role for definition of interfaces between the behavior module and other software components, as illustrated in Figure 2. Those are:

1) Situation Interpretion: Delivers a significant description of the environment. Perception data is logically split and expanded upon other attributes and relations (e.g. lane is usable, occupied with objects A and B). A strategic behavior will be selected according to the shape of the planned path and information about admissible velocity and street category (as in a navigation system). This sets the focus, but will not result in immediate action. This interface is work in progress at the moment and will be the subject of another publication [7].
2) Vehicle Control: The interface to the vehicle control needs to generate input for the control to move the vehicle, but has to make sure that it leaves room for dynamical improvements, which are not modeled in detail in the behavior network. A driving corridor seems suitable for this task to commit the driving intention of the behavior network. It resembles a free street to the control part, which is able to optimize according to certain criteria as e.g. lowest lateral acceleration. See Section III-B for more details.
3) Cooperative Decision: The active behaviors will be, as well as the interpretation data, transferred to a cooperative decision module. When performing cooperative driving manoeuvres, the module is able to assign a compulsive behavior for the vehicle. This could be helpful in critical situations or to obtain more efficient traffic flow, and will be implemented at advanced state.

III. Behavior Network Design and Implementation

A. Motivation and Intention

Human driving decisions come off through permanent thinking ahead and balancing of options. Where a driver is able to tell the most important reasons for the decision, he or she considered subconsciously further aspects that had an influence on the result. Reduction on few or even a single driving aspect occurs only in case of emergency, where certain behaviors with low latency create fast reactions, as e.g. *Avoid Obstacle*.

Apparently, human driving behavior is affected by cooperative decisions, whose can also overwrite each other depending on character and importance, but usually complement one another. While the group of E.D. Dickmanns achieved their impressive results with a competing architecture [8], a cooperative architecture was chosen in this approach to represent human-like driving. The behavior decision and execution for a cognitive vehicle should be related closely to those of humans, due to the following reasons:

- Human driving behavior has proven itself and is able to handle very complex and unknown situations. Negative aspects of human driving as carelessness or recklessness are not considered and do not reduce favorable qualities.
- Complete modeling of the environment might not be possible due to complexity and diversity of situations, humans are able to compensate this fact with reactive behavior patterns.
- Without complete modeling of the environment, there exists no mathematical proof if a certain decision was the most appropriate or not. In fact, humans will have to act as teachers and judges to rate a manoeuvre. Similar behavior models facilitate this process.

B. Driving Corridor

A driving corridor was chosen as interface to the underlying control and represents a free street, which can be used for trajectory planning and dynamical optimizations. The corridor representation should be independent from lane models used in perception, so that polygon lines seem to fit best to describe corridor boundaries. Even though dynamical details are considered in the control part, the behavior network is responsible of delivering a drivable corridor, and needs to take into consideration some limitations. Therefore, the corridor should feature the following properties, illustrated in Figure 3:

- The corridor is described by a set of point pairs, each point pair is orthogonal to the corridor direction (connection between center of previous and actual point pair). Thus, the distance between two points of a pair is the width w_i of the corridor at this point.
- Each center of a point pair is connected with the previous and next midpoints, the distance d between those points is the same throughout the corridor.
- The final output of the behaviors results in a force F_i for each point pair, which causes a rotation

$$\varphi_{i_t} = \frac{F_i}{D_i}$$

  around the center of the previous pair, according to the spring constant D_i of this joint. To consider dynamic constraints of the car already in the corridor shape, the spring constant was chosen as

$$D_i = b \cdot \frac{v^2}{(i \cdot d)^2}$$

  where v is the vehicle velocity, $i \cdot d$ the distance from the vehicle and b a tuning parameter.
- Significant changes from previous to new corridor are only permitted in a certain distance from the car, this is ensured by limiting the angle φ_{i_t} according to the last value so that $|\varphi_{i_t} - \varphi_{i_{t-1}}| \leq \varphi_{max}$, where

$$\varphi_{max} = \frac{c}{(i \cdot d)^2}$$

  and c is a tuning parameter. This value is also restricted to the maximum angle, given through the minimum turning circle of the car.
- Additional parameters v_d and a_d contain the desired speed and acceleration to be adjusted if possible. A specific point pair P_F marks the front of the corridor, which moves with velocity v_F and cannot be crossed.

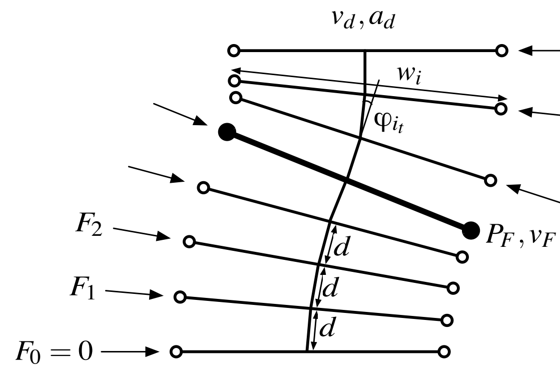

Fig. 3. Influences of behavior outputs on corridor point pairs to form a driving corridor

A useful extension would be the acceptance of different distances between center points to decrease computational expenses at higher speeds. By generating multiple corridors with different boundaries, distinction between *desired, acceptable* and *emergency* corridors would be possible and increase independency of the control.

C. Implemented Behaviors

During behavior development, the main focus has been set on the identification of the different actions a human vehicle driver performs. In a second step these actions have been taken to derive the behaviors for the network. Figure-4 gives an overview of the implemented behaviors. Three layers have been identified which represent the different time horizons of the behaviors. The *strategic layer* includes deliberative, long-term behaviors which use a high amount of knowledge to operate. Behaviors containing driving actions are placed on a *tactical layer*. They coordinate movements with a time scope of two to ten seconds and use reactive behaviors to generate movements.

These different time scopes can be observed also on human drivers who plan their route through a city and plan how to pass a crossing, but still show reactive behavior for example to brake if a child runs on the street.

Besides behavior elements, fusion nodes are necessary to merge outputs of different behaviors. The outputs can be weighted with the help of activity values from each behavior. In the following section, important network behaviors will be described in detail. The behavior descriptions are ordered in the way they were implemented, from reactive to strategic:

- **Keep Lane** This behavior pulls the driving corridor towards the current lane. To control the vehicle more precisely, it is possible to set a relative in-lane position as input value. To calculate the output of this behavior,

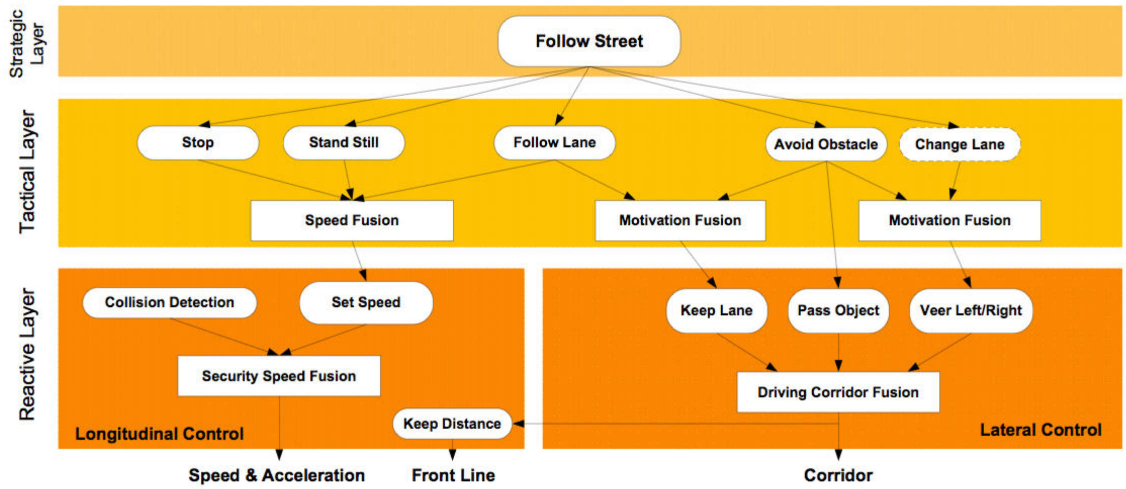

Fig. 4. Implemented set of basic behaviors to execute *Follow Street*

every point pair of the driving corridor is considered. The examination is based on two values, the distance l_i the point pair is displaced from the requested in-lane position and the angle α_i between the corridor direction and the lane direction at this point pair. In more detail, the force F_i is calculated for each corridor point pair P_i by the following formula in which the capital letters are constants and determine the behavior characteristics:

$$F_i = (A \cdot l_i^2 + B \cdot \alpha_i^2) \cdot \iota$$

The reflection r is computed as the sum of displacements

$$r = \frac{C}{\sum_{i=0}^{n} l_i}$$

The activity a results as $a = 1.0 - r$ to express the necessary amount of work.

- **Pass Object** To keep the safety distance between the vehicle and objects that are going to be passed, this behavior has been introduced. If the distance drops under a certain limit an output is generated which pushes the driving corridor away from the object to increase the lateral distance. The rate of corridor modification can be influenced by the number m of point pairs that are changed and the force F_j by which these are pushed. Both values can be seen as results and are calculated as:

$$m = \frac{D}{d_o}$$

$$F_j = E \cdot \iota$$

where d_o is the distance between the vehicle and the object, v_d the relative speed $v - v_o$ with $v_d \geq 0$. D and E are constants. The activity and reflection formulas are defined as:

$$r = \frac{d_o}{G \cdot v_d} \qquad a = 1.0 - r$$

- **Veer Left/Right** This behavior is competitive to *Keep Lane* and pulls the corridor out of the current lane. The direction to which the corridor should be moved and the offset from the actual lane serve as behavior input values.
- **Keep Distance** The task of this behavior is to set the front line and its velocity attribute so that the control part is able to brake if the distance drops under a determined range. Only objects within the corridor are considered in this behavior.
- **Set Speed** This behavior sets the desired vehicle speed in the from of a corridor attribute. According to its motivation and the difference in desired and actual speed, it sets the desired acceleration value a_d according to:

$$a_d = H \cdot \iota$$

where H is a tuning parameter. In this behavior, the activity and reflection are not closely related to each other and calculated as:

$$a = \iota \qquad r = \frac{K}{|v_d - v| + L}$$

with parameters K and L.

- **Follow Lane** This behavior belongs to the tactical layer which means that it uses beahviors of the lower level (reactive level) to perform a driving action. In order to guide the vehicle on a given lane, it is necessary to motivate behaviors for lateral control (*Keep Lane*) and longitudinal control (*Set Speed*).
- **Avoid Obstacle** In an urban environment, obstacle avoidance is essential, which is the task of this behavior. To do so, it uses reactive behaviors *Keep Lane*, *Pass Object*, *Veer Left/Right* and *Set Speed*. Since *Keep Lane* and *Veer Left/Right* are competitive behaviors it is important that changes of motivation values are coordinated. E.g. if the vehicle has to use the left lane of a road to pass an obstacle, the motivation of *Veer*

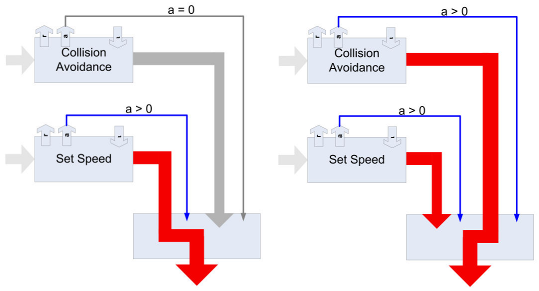

Fig. 5. Fusion nodes as instrument for priorization: in case of a *Collision Avoidance* activity $a > 0$, the network structure is adapted to suppress the output of *Set Speed*.

Left needs to be increased and the one of *Keep Lane* decreased. If the motivation of *Keep Lane* was not changed, leaving the lane would be hardly possible. This behavior is, in opposite to the reactive ones it uses, in charge of controlling the entire avoidance process.

- **Follow Street** This behavior is situated on the highest level of the hierarchy, the strategic layer. It contains all knowledge about how to perform the entire manoeuvre, and how to generate actions in coordinating underlying tactical behaviors. The procedure of a certain manoeuvre is structured in a finite state machine. Figure-4 shows that this behavior uses *Follow Lane* and *Avoid Obstacle* to generate movement (*Change Lane* is not implemented yet). Motivation between those behaviors is distributed according to the existance of an obstacle. In case of obstacles, fusion nodes within the network ensure that *Avoid Obstacle* has a higher control priority of lower level behaviors than *Follow Lane*, since it is more important to avoid an obstacle than to strictly follow the given lane.
The strategic behavior *Follow Street* is in charge of supervising the entire process and gets feedback through behavior reflections to indicate behavior progress.

D. Safety

Due to the complexity of the behavior network and the missing formal description of the environment, it is very difficult to prove general safety in mathematical sense. With the ability to prioritize behaviors through fusion nodes, safety can be ensured for a set of standard situations.

Selection of such a representative set of situations for autonomous vehicles is a challenging task and might be established in addition to known crash-tests in automotive industry. A clear distribution of responsibilities within the network is important to make use of so-called *safety fusion nodes*. Therewith, network structure reduces in critical situations to essential behaviors. Figure 5 illustrates, how the behavior *Collision Avoidance* overwrites the target velocity, specified by the behavior *Keep Lane* (or other behaviors with influence on longitudinal movement), to stop the vehicle in case of emergency.

Each safety fusion node is a function of multiple user data inputs and activities $u_{sf}(u_1, u_2...u_n, a_1, a_2..a_n)$ and decides according to the *Winner takes all* principle, which behavior generates the output.

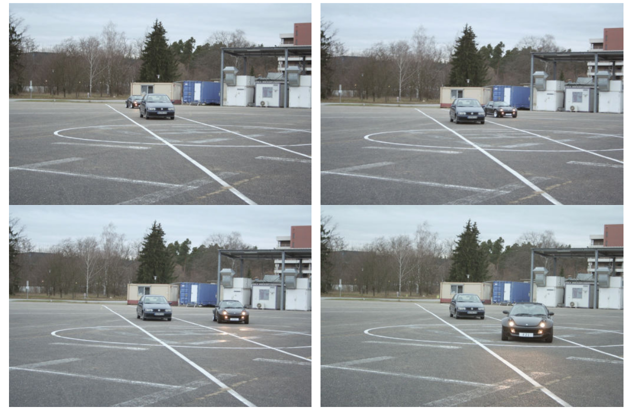

Fig. 6. Testing strategical behavior *Follow Street* with essential underlying behaviors.

Safety fusion nodes differ from regular fusion nodes only in the respect that they work without weights, which would not be acceptable to guarantee safety, and have only one responsible behavior on top. Insertion of safety nodes takes place only after integration of other network components, starting from the bottom.

E. Experiments

To validate the results obtained from simulation, experiments were carried out on a real testing vehicle. The strategic behavior *Follow Street* was started manually, and controlled underlying tactical and reactive behaviors. The course consisted of a straight lane with obstacle, both detected by the stereo camera. (see Figure 6). The interpretation module extended the scenario upon a virtual second lane, placed left of the real lane, and a virtual obstacle at the end of the lane which ranged over both lanes and could not be passed.

Passing on the second lane was only possible since it was marked as free from the interpretation module. Otherwise, the tactical behavior *Stop* would have been motivated.

In Figure 7, the results of the manouvre can be seen in terms of motivation and reflection values of the most involved behaviors. The experiment was carried out with a speed of about $8\frac{m}{s}$, which is reasonable for such a narrow passing scenario. The obtained movement was as expected from simulation, and the first parameter setup seemed to work fine for the beginning.

IV. CONCLUSION AND OUTLOOK

This paper introduced a behavior-based approach to control an autonomous vehicle. Human-like driving behaviors were implemented in a hierachical structure. This provides the possibility to compare not just the result but also used behaviors with a human driver's description of his/her operations.

It was shown, how safety aspects can be considered by using reactive behaviors and fusion nodes for prioritization of behaviors. In contrary to other behavior-based approaches, the interior values were made accessable and can be used for

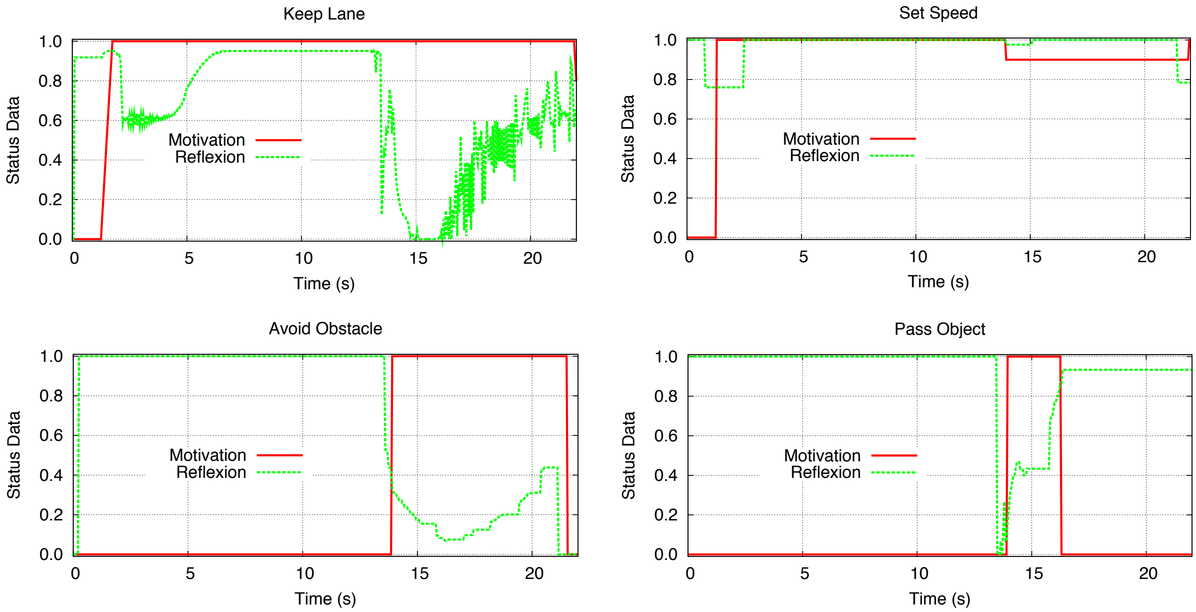

Fig. 7. Diagrams above show motivation and reflection values of the most important behaviors, necessary to perform the overall behavior *Follow Lane*. Activation is not shown since it is $a = 1 - r$ at the moment for all behaviors. ***Keep Lane*** is motivated right from the beginning. The minimum after $3s$ is a re-alignment after giving a new in-line position, the minimum after $15s$ occurs when the focus changes immediately to the left lane in order to pass the other car. ***Set Speed*** has only relevant discontent (and with it activity) in case of acceleration or deceleration. ***Avoid Obstacle*** has low reflection when the first obstacle occurs after $14s$, which then increases after passing the car and turns completely 0 after the second obstacle blocking the street. Behavior ***Pass object*** has the task to keep distance to obstacles and forming the corridor accordingly. The low reflection when passing produces activity and increases again after successfull passing. The behavior is not affected from the second obstacle, since this one cannot be passed.

behavior coupling or network feedback. With a first set of behaviors, it was possible to perform an autonomous driving manoeuvre *Follow Street* on a testing course. The chosen interface to the controller module in the form of a driving corridor seems adequate to express the driving intention.

Due to the nature of the behavior network structure, correct tuning of the parameters is essential. Determination through experiments results in a set of working parameters, but has no claim to be perfect for various situations. A learning process (in a restriced automotive sense) would help to adapt the initial parameters within a given range.

While this approach of behavior output fusion seems to work for the presented scenario, it is not possible to predict the resulting corridor. One way for doing this, and to consider clear geometric constraints of the environment, would be the usage of potential or risc maps to express the desire of each behavior.

The oncoming tasks will be to implement further behaviors on strategic level, such as *Turn into Street, Use Exit* or *Overtake Car*. With more complex scenarios, it will be necessary to have an intelligent decision module in between the strategic level (manoeuvres) and tactical level (actions). After implementation of a wide set of behaviors, usage of internal network data (reflections and activities) is going to be used for manoeuvre rating, and to apply optimization criteria. Storage in a semantic map would help to re-use this information for the next mission, other cars or the developers.

V. ACKNOWLEDGMENTS

The authors gratefully acknowledge support of this work by the Deutsche Forschungsgemeinschaft (German Research Foundation) within the Transregional Collaborative Research Centre 28 'Cognitive Automobiles'.

REFERENCES

[1] J. Albiez, T. Luksch, W. Ilg, and K. Berns. Reactive reflex based posture control for a four-legged walking machine. In *Proceedings of CLAWAR*, Karlsruhe, Germany, September 2001.

[2] Jan Albiez and Rüdiger Dillmann. Behaviour networks for walking machines - a design method. Technical report, Forschungszentrum Informatik (FZI), Karlsruhe, Germany, 2004.

[3] Ronald C. Arkin. *Behavior-based Robotics*. The MIT Press, Cambridge, Massachusetts, 1998.

[4] SFB/TR28 Cognitive Automobiles. http://www.kognimobil.org, January 2007.

[5] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 1986.

[6] Rodney A. Brooks. Intelligence without representation. Technical report, MIT Artificial Intelligence Laboratory, Cambridge, 1987.

[7] Daniel Jagszent. Implementierung von fahrverhalten zur durchführung eines abbiegevorganges. Master's thesis, University of Karlsruhe (TH), 2007.

[8] M. Pellkofer and E.D. Dickmanns. Behavior decision in autonomous vehicles. In *Proceedings of IEEE Intelligent Vehicle Symposium*, pages 495 – 500, Vol.2, June 2002.