https://doi.org/10.2352/ISSN.2470-1173.2019.15.AVM-041

Autonomous highway pilot using Bayesian networks and hidden Markov models

Kurt Pichler, Sandra Haindl, Daniel Reischl, Martin Trinkl; Linz Center of Mechatronics GmbH; Linz, Austria

Abstract

In this paper we propose a data driven model for an autonomous highway pilot. The model is split into two basic parts, an acceleration/deceleration model and a lane change model. For modeling the acceleration, a Bayesian Network is used. For the lane change model, we apply a Hidden Markov Model. The lane change model delivers only discrete lane change events like stay on lane or change to left or right, but no exact trajectories. The model is trained with simulated traffic data, and validated in two different scenarios: in the first scenario, a single model controlled vehicle is embedded into a simulated highway scenario. In the second scenario, all vehicles on a highway are controlled by the model. The proposed model shows reasonable driving behavior in both test scenarios.

Introduction

Private transport has become more and more important over the last decades since people put emphasis on individual mobility. Together with the economic growth this leads to a rapidly increasing number of traffic participants. It is not possible to match the increased number of vehicles with appropriate expansion of infrastructure. This results in a rising traffic density which demands increased attention from the driver.

It is known that human driving errors are the main reason for traffic accidents [1–3]. Therefore it is reasonable to try to eliminate the main risk factor, the human behavior, in today's traffic. Thus, an autonomous driving system is an important contribution to increase traffic safety significantly.

Even though increased safety is the main advantage of an autonomous driving system, there are also more benefits. Autonomous drivers can be more economical by anticipatory driving and employing a smoother driving style than most human drivers do. Furthermore, by communicating with other vehicles and the infrastructure, autonomous driving will be able to increase traffic throughput in the future.

However, the aim of this contribution is to create a highway pilot that participates in normal highway traffic by predicting the driving maneuvers acceleration/deceleration and lane change. These maneuvers can be seen as the driver's intention on the tactical level of the scheme introduced by [4]. The intention of the driver cannot be observed directly, it has to be inferred from other observable signals. There are many potential signals that can be used to infer the driver's intention, for instance the ego vehicles environment, pedal positions, steering wheel actions, viewing direction of the driver, and many more. The effort for measuring these signals is different. Depending on the type of the signal, some are more difficult and less robust to determine than others. In this paper, we assume that only the most basic signals are available, this means the ego vehicle's actual and desired velocity, the positions and velocities of the surrounding vehicles and some basic information about the highway such as the number of lanes.

A wide range of papers have been published dealing with the task of autonomous driving. A survey over different approaches and a general theory of driver behavior is provided in [5–7]. The driver's actual intention is recognized in [8, 9] using Hidden Markov Models (HMMs). A recognition and prediction method for certain driving events and maneuvers using HMMs is proposed in [10, 11]. The intended action of a driver is modeled as a sequency of mental states using Markov Dynamic Models in [12]. Optimal policy determination in a certain prediction horizon is proposed in [13] by assuming a markov process for the policy states. This approach is extended in [14] by predicting other drivers behavior using change point detection and Viterbi algorithm. Collision avoidance by steering is considered in [15] using a stochastic switched ARX model. A broad range of different Bayesian Network (BN) models for driver behavior modeling and prediction is proposed in [16–20]. They all use different variables and net structures. The acceleration and deceleration behavior of a driver is modeled as a Dirichlet process mixture model in [21]. A various range of features is extracted from vehicle's sensor signals in [22] and classified using Relevance Vector Machines. The prediction of different models in different nodes is combined in a competition function to increase the accuracy of the decision [23]. Lane change is predicted in a 2 seconds horizon using the velocities and accelerations of the vehicles ahead and locally-weighted projection regression for classification in [24]. Lane change prediction in urban streets for drivers of different aggressiveness levels using a legistic regression model is proposed in [25], while using support vector machines for lane change detection is proposed in [26]. Risk function orientated autonomous overtaking integrating safety indicators as time headway and time to collision is discussed in [27]. A control algorithm for autonomous overtaking using stochastic model predictive control is presented in [28]. It relies on suitable prediction for longitudinal and lateral speeds of the surrounding vehicles.

In this paper, we propose a dual model to realize a highway pilot: the acceleration and deceleration is predicted by a BN model [29,30], while the lane change is predicted by a HMM [31]. Even though similar models are proposed in literature before, all of them mentioned above use more or different input signals than we have available. The acceleration model only takes the velocity of the ego vehicle and the preceding vehicle into account. The lane change model considers the traffic in the lane of the ego vehicle as well as the traffic in the lanes on the left and right side of the ego vehicle.

Problem Statement

An automated highway pilot faces several challenges. The main task is to integrate into the moving traffic and to avoid collisions. However, it has limited knowledge of the surrounding environment and the other vehicles.

In our particular application scenario, the highway pilot knows the number of lanes in driving direction and its actual lane. Furthermore it has a desired velocity value, i.e. the velocity that it would like to drive on a free road. The highway pilot receives measurement data from its on-board sensors. These data contain information about the 5 surrounding vehicles of the ego vehicle:

- Vehicle in front in the same lane
- Vehicle in front in the right lane
- Vehicle in front in the left lane
- Vehicle behind in the right lane
- Vehicle behind in the left lane

The sensors deliver the distance of the ego vehicle to the surrounding vehicles and the actual velocity of those vehicles. The distance values are in relation to the ego vehicle, i.e. the vehicles in front of the ego vehicle have positive distances, the vehicles behind of the ego vehicle have negative distances. The velocity value represents the absolute driving velocity of each vehicle. For simplicity reasons, we assume that all sensors work and deliver correct values at all time instances.

The task of the highway pilot is to predict a certain acceleration and lane change to be performed within the following time step. We also assume that lane changes and desired accelerations are performed and completed in one time step of 0.5s.

Models

In highway driving scenarios, there are numerous driving situations and events. It is simply impossible to enumerate all sorts of situation a driver might face during a highway ride. Therefore a rule based highway pilot will not be able to cover the whole range of occurring situations. Hence we decided to use data driven models to predict the desired acceleration and lane change of the ego vehicle. The parameters of the models are trained using data of a simulated highway scenario.

We are well aware of the fact that satisfying autonomous driving results have been achieved by employing deep learning approaches in combination with high performance computation hardware [32]. However, since our hardware resources are restricted to ordinary personal computers, we are looking for models with less computational effort.

According to the introduction section, researchers report that especially HMMs and BNs are capably to fit the needs of autonomous driving. After testing some approaches we decided to train separate models for acceleration and lane change. For acceleration a BN performed best, while for changing lanes a HMM delivered better results than other tested models. The details of the models are described in the following subsections. The selection of the proposed input structures was performed using well known feature selection filter and wrapper approaches [33]. The models are applied in a time resolution of 0.5s, i. e. each 0.5 seconds the models deliver a new prediction for the following time step based on the actual sensor values.

We intentionally don't use physical units in the description of the model. The models we use are data driven and the model inputs are normalized before applying the model. We only have to make sure that the training and test data use the same physical units.

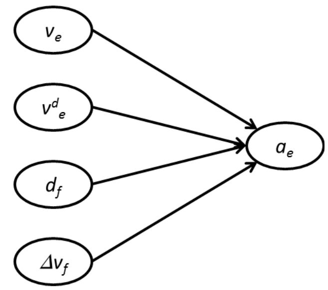


Figure 1. *Structure of the BN.*

Acceleration Model

The acceleration model considers only the traffic in the actual lane of the ego vehicle, i. e. it evaluates the ego vehicle itself and its preceding vehicle. A simple BN using the following inputs performed best amongst all evaluated input and net structures

- Ego vehicle velocity v_e
- Ego vehicle desired velocity v_e^d
- Distance to vehicle in front d_f
- Velocity of vehicle in front v_f

From the ego vehicle's velocity v_e and the velocity of the vehicle in front v_f, the velocity difference $\Delta v_f = v_e - v_f$ is computed and used as a model input instead of just using v_f. The net structure that is used is shown in Fig. 1 with a_e denoting the acceleration of the ego vehicle as the output of the model.

The conditional probability tables (CPT) of the BN are trained based on training data using maximum a posteriori (MAP) estimation [34]. For that purpose the continuous input and output signals are quantized in certain bins. The bins are defined according to the training data.

Given the CPTs and the quantized input signals, the acceleration can be predicted. Even with incomplete data a prediction could be made using for instance Gibbs sampling for the missing inputs. In a certain time step, the model delivers a probability value

$$P\left(a_{e,i}|v_e, v_e^d, d_f, \Delta v_f\right),\ i = 1,\dots,n \quad (1)$$

for each acceleration bin denoted by $1,\dots,n$ in ascending order, with v_e, v_e^d, d_f and Δv_f denoting here the quantized input values. The probabilities $p_i = P\left(a_{e,i}|v_e, v_e^d, d_f, \Delta v_f\right)$ fulfill the conditions $p_i \in [0,1]$ and $\sum_{i=1}^{n} p_i = 1$.

The straightforward way of choosing the bin with the highest probability results in a deterministic driver that performs exactly the same driving maneuvers every time. However, the more realistic scenario is a probabilistic driver that chooses between some possible driving maneuvers in a probabilistic way. To realize this, we compute the cumulative sums of the output probabilities p_i by $c_i = \sum_{j=1}^{i} p_j$ for $i = 1,\dots,n$. Then a random number $r \in [0,1]$ is generated and the highest acceleration bin i with $r <= c_i$ is chosen as the model output.

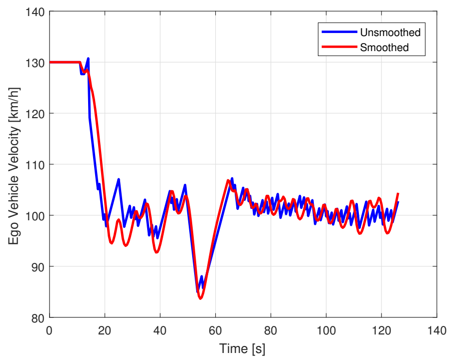


Figure 2. *Example of an acceleration profile before and after smoothing.*

The quantization of the input and output signals entails that the model delivers also a quantized output. Since it is not desirable to have a quantized acceleration and velocity profile, the acceleration values are smoothed in a post processing step to obtain a smoother profile. The smoothing is done using Whittaker's smoother [35–37]. A comparison between a typical velocity profile in our tests before and after acceleration smoothing can be seen in Fig. 2.

Lane Change Model

The HMM for lane change considers both the traffic situation on ego vehicles lane and the traffic situation in the two neighboring lanes (left and right of the ego vehicle's lane). The inputs for the lane change model are computed from the raw sensor data. These inputs are the observations of the HMM, while the lane change is the hidden state to be predicted. The raw sensor data needed to compute the model inputs are

- Ego vehicle velocity v_e
- Ego vehicle desired velocity v_e^d
- Distance to vehicle in front d_f
- Velocity of vehicle in front v_f
- Distance to vehicle in front in the right lane $d_{f,r}$
- Velocity of vehicle in front in the right lane $v_{f,r}$
- Distance to vehicle behind in the right lane $d_{b,r}$
- Velocity of vehicle behind in the right lane $v_{b,r}$
- Distance to vehicle in front in the left lane $d_{f,l}$
- Velocity of vehicle in front in the left lane $v_{f,l}$
- Distance to vehicle behind in the left lane $d_{b,l}$
- Velocity of vehicle behind in the left lane $v_{b,l}$

From these data, certain input features for the model are computed. The difference of the ego vehicles velocity to its desired velocity is computed as $\Delta v_e^d = v_e - v_e^d$. The velocity difference of the ego vehicle to its preceding vehicle is computed as $\Delta v_f = v_e - v_f$.

For the vehicle in the right lane and in the left lane of the ego vehicle, the estimated time to collision (ETTC) is computed. Therefore, we again compute the velocity differences to the surrounding vehicles as

$$\begin{aligned}\Delta v_{f,r} &= v_e - v_{f,r}\\ \Delta v_{b,r} &= v_e - v_{b,r}\\ \Delta v_{f,l} &= v_e - v_{f,l}\\ \Delta v_{b,l} &= v_e - v_{b,l}\end{aligned} \tag{2}$$

and the resulting ETTCs as

$$\begin{aligned}\mathrm{ETTC}_{f,r} &= \begin{cases}\frac{d_{f,r}}{\Delta v_{f,r}} & \text{if} \quad \Delta v_{f,r} > 0\\ \infty & \text{if} \quad \Delta v_{f,r} \leq 0\end{cases}\\ \mathrm{ETTC}_{b,r} &= \begin{cases}\frac{d_{b,r}}{\Delta v_{b,r}} & \text{if} \quad \Delta v_{b,r} < 0\\ \infty & \text{if} \quad \Delta v_{b,r} \geq 0\end{cases}\\ \mathrm{ETTC}_{f,l} &= \begin{cases}\frac{d_{f,l}}{\Delta v_{f,l}} & \text{if} \quad \Delta v_{f,l} > 0\\ \infty & \text{if} \quad \Delta v_{f,l} \leq 0\end{cases}\\ \mathrm{ETTC}_{b,l} &= \begin{cases}\frac{d_{b,l}}{\Delta v_{b,l}} & \text{if} \quad \Delta v_{b,l} < 0\\ \infty & \text{if} \quad \Delta v_{b,l} \geq 0\end{cases}.\end{aligned} \tag{3}$$

To get just one single value for each lane, we finally compute the minimum of the two ETTCs on each lane according to $\mathrm{ETTC}_r = \min\left(\mathrm{ETTC}_{f,r}, \mathrm{ETTC}_{b,r}\right)$ for the right lane of the ego vehicle and $\mathrm{ETTC}_l = \min\left(\mathrm{ETTC}_{f,l}, \mathrm{ETTC}_{b,l}\right)$ for the left lane of the ego vehicle. In the case that there are no vehicles at certain positions in relation to the ego vehicles, the according input features are set to default values.

Finally, the features Δv_e^d, d_f, Δv_f, ETTC_r, and ETTC_l are used as model inputs, i.e. as observations of the HMM. The transition and the emission probabilities of the HMM are trained based on training data using a maximum likelihood estimator [38]. For that purpose, the continuous input and output signals are quantized in certain bins. The bins are defined according to the training data.

Given the inputs, the transition and the emission probabilities, the HMM predicts the lane change value $c \in \{-1, 0, 1\}$ with the meaning

$$c = \begin{cases} -1 & \ldots \quad \text{change to right lane}\\ 0 & \ldots \quad \text{stay on lane}\\ 1 & \ldots \quad \text{change to left lane.}\end{cases} \tag{4}$$

As in the acceleration model, the lane change model delivers a probability value $P(c)$ for each possible lane change $c \in \{-1, 0, 1\}$ with $\sum_{c \in \{-1,0,1\}} P(c) = 1$.

The final selection of the lane change maneuver is realized analogously to the acceleration model by computing the cumulative sums of the probabilities and generating a random number. However, to avoid totally random lane changes, a majority vote of predicted lane changes is performed over the last few time steps. Additionally, to consider safety issues, the distances to the vehicles in the target lane is observed. If one of those distances is below a certain safety distance, the lane change will not be performed. Furthermore, a separate lane change model is trained and applied if the ego vehicle is in the far right or far left lane. Those models exclude lane changes to the not existing neighboring lane.

Training and Test Setup

The models were trained using simulated data of the commercial traffic simulation system PTV VISSIM [39]. Four data sets on a highway with four lanes were produced, two of them on a 2 km highway section and two of them on a 4 km highway section. Altogether approximately 2 hours of traffic data with 10190 vehicles were simulated. The data contain only flowing traffic with velocities between 80 and 160 km/h. From all vehicles, the features were extracted and used as training data for the proposed models.

For testing the models, two different approaches were implemented. In the first one, a single instance of vehicle controlled by the proposed models was embedded in a simulated 4-lane VISSIM highway scenario using a Matlab interface. In the second one, a Matlab highway scenario was created containing only vehicles controlled by the proposed models. The vehicles are created randomly at the beginning of a highway section with a random desired speed, initial speed and lane. Then the vehicles drive along this highway section applying the models in every time step. On each lane, a vehicle was created with a probability of 20% each second (i.e. 1 vehicle every 5 seconds) and with a minimal time difference of 2 seconds in the same lane. Moreover, the desired velocity v_d of each vehicle was uniformly distributed between 90 km/h and 140 km/h, and the initial speed was uniformly distributed between 90 km/h and v_d km/h. Each vehicle starts at position 0 m on the highway and drives until it exceeds the 3 km limit, then it is deleted.

Results

Exemplary results of the two test scenarios described in the previous section are provided here.

Single controlled vehicle in VISSIM environment

In the first test scenario, the controlled vehicle drives on the 4 km highway section with 4 lanes in the VISSIM environment. It is placed initially on a starting position of 400 m on lane 2, with an initial velocity of 110 km/h and a desired velocity of 120 km/h.

Fig. 3 shows an exemplary result of this test scenario. The upper subplot shows the local environment of the ego vehicle at the end of the highway, the driving direction is from left to right. The rectangles represent the vehicles (ego vehicle in red). The second subplot shows the velocity of the ego vehicle throughout the whole test run, while the third and fourth subplots show the distance to the preceding vehicle and the ego vehicle's lane respectively.

In the beginning, the ego vehicle accelerates to its desired velocity and is significantly faster than the (changing) preceding vehicles far in front. When the ego vehicle approaches the preceding vehicles, it assimilates the velocity to keep the distance. After a short change to lane 3 (between 1280 m and 1420 m) the vehicle stays on lane 2 again and keeps the distance to its preceding vehicle by assimilating the velocity. As another vehicle pulls into the ego vehicles lane at 1900 m, the ego vehicle has to decelerate quickly. After changing to lane 3 at 1970 m and lane 4 at 2300 m, the ego vehicle accelerates again towards its desired velocity, only adjusting the velocity to keep the distance to the preceding vehicle at about 40 m to 50 m.

Although it is not possible to show all relevant driving information here (for instance the distances and velocities of all surrounding vehicles), the driving behavior appears quite reasonable. Some desirable lane changes were prevented by other vehicles in the target lane. Furthermore, after approaching the preceding vehicles the ego vehicle was not able to reach its desired velocity any more due to the traffic, even by changing to the far left lane.

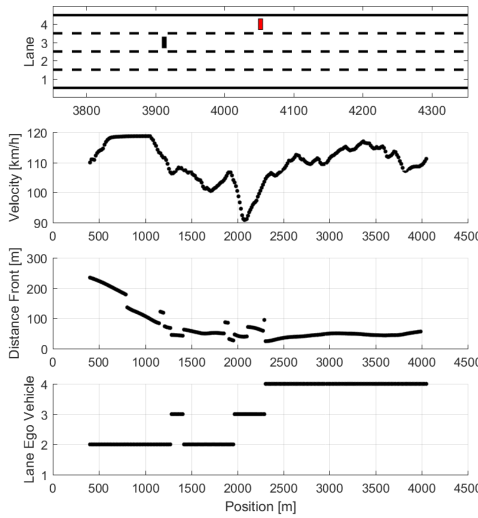


__Figure 3.__ Exemplary result of a single controlled driver embedded in a VISSIM environment.

All controlled vehicles in Matlab environment

An exemplary driving maneuver is shown in Fig. 4. The vertical rectangles represent the vehicles on the highway, while the horizontal lines represent their trace in the last 5 seconds of the simulation. We simulated a 3 km highway section with 4 lanes, the driving direction is from left to right.

The left hand side shows the scenario at simulation time 106 seconds. The blue vehicle on lane 4 had an initial velocity of 102 km/h, a desired velocity of 115 km/h and is currently driving on position 1663 m with a velocity of 112 km/h. The white vehicle on lane 3 had an initial velocity of 101 km/h, a desired velocity of 102 km/h and is currently driving on position 1683 m with a velocity of 102 km/h. The blue vehicle just changed from lane 3 to lane 4 to overtake the white vehicle.

The right hand side shows the scenario at simulation time 130.5 seconds. The white vehicle changed from lane 3 to lane 2 and is driving now at position 2385 m with a velocity of 103 km/h. The blue vehicle changed back to lane 3 and is driving now at position 2436 m with a velocity of 115 km/h. Hence the blue vehicle performed an overtaking maneuver and went back to its original lane when the lane was clear again.

Another indicator for the performance of the model is the average number of cars and the average velocity on each lane. With the obligation to drive right, it is assumed that there are more vehicles in the right lanes than in the left lanes. Furthermore, the faster vehicles will be more likely to drive in the left lanes than in the right lanes. Even though the vehicles are created with the same probability and the same distribution of initial and desired speed on each lane, the model yields exactly the expected behavior. For the 3 km highway section, the average number of vehicles on each lane and their average velocity after a 12 hour test run can be seen in Table 1. It shows that the average number of vehicles on a lane decreases from right to left, while the average velocity is increasing from right to left. Furthermore, three rear end collisions occurred in this test run. This was the case when two vehicles changed to the same lane from two sides in the same time step and the distance between them was not big enough to avoid

Statistics of the simulation scenario

Lane	Average number of vehicles on lane	Average velocity of the vehicles on lane [km/h]
1	24.44	103.06
2	17.74	112.94
3	11.65	115.91
4	6.10	117.29

the collision. Since the model has no information what other vehicles plan (especially vehicles 2 lanes away), this situation could not be avoided by the proposed model.

Conclusions

We proposed a simple model to predict the driving events acceleration/deceleration and lane change for autonomous driving. Having in mind the simplicity of the model, it performs well in doing reasonable driving maneuvers.

However, there are still some improvements to be done. The autonomous vehicle rarely keeps a constant velocity, there are many minor acceleration and deceleration events. The acceleration model and the lane change model work independent of each other. A collaboration of the two models might result in more smooth driving actions like overtaking. Another task that might be improved is including the velocity information in the target lane in the lane change model. Even though the feature selection did not select it as a significant feature, it seems to be intuitively important, also from human driving experience. Furthermore, the model has to be tested in more details including pre designed critical driving scenarios.

Acknowledgments

This work has been partially conducted within the ENABLE-S3 project that has received funding from the Ecsel Joint Undertaking under grant agreement no 692455. This joint undertaking receives support from the European Unions Horizon 2020 Research and Innovation Programme and Austria, Denmark, Germany, Finland, Czech Republic, Italy, Spain, Portugal, Poland, Ireland, Belgium, France, Netherlands, United Kingdom, Slovakia, and Norway.

References

[1] S. H. Chen, J. S. Pan, and K. Lu, “Driving behavior analysis based on vehicle OBD information and AdaBoost algorithms,” Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, 2015

[2] D. L. Hendricks, J. C. Fell, and M. Freeman, “The relative frequency of unsafe driving acts in serious traffic crashes,” Report no. DOT-HS-809-206, 2001

[3] Statistisches Bundesamt, “Verkehr - Vekehrsunfälle,” Fachserie 8, Reihe 7, Wiesbaden, Germany, 2017

[4] J. A. Michon, “Human behavior and traffic safety,” Plenum Press, New York, 1985

[5] M. K. Kumar, and V. K. Prasad, “Driver behavior analysis and prediction models: a survey,” International Journal of Computer Science and Information Technology, vol. 6, no. 4, pp. 3328–3333, 2015

[6] W. Wang, J. Xi, and H. Chen, “Modeling and recognizing driver behavior based on driving data: a survey,” Hindawi Publishing Corporation, Mathematical Problems in Engineering, 2014

[7] R. Fuller, “Towards a general theory of driver behavior,” Elsevier Accident Analysis & Prevention, vol. 37, no. 3, pp. 461–472, 2005

[8] H. Berndt, J. Emmert, and K. Dietmayer, “Continuous driver intention recognition with hidden Markov models,” Proceedings of the 11th IEEE Conference on Intelligent Transportation Systems, pp. 1189–1194, 2008

[9] L. He, C. F. Zong, and C. Wang, “Driving intention recognition and behavior prediction based on a double-layer hidden markov model,” Journal of Zhejiang University-SCIENCE C (Computers & Electronics), vol. 13, no. 3, pp. 208–217, 2012

[10] D. Mitrovic, “Reliable method for driving events recognition,” IEEE Transactions in Intelligent Transportation Systems, vol. 6, no. 2, pp. 198–205, 2005

[11] K. Torkkola, S. Venkatesan, and H. Liu, “Sensor sequence modeling for driving,” Proceedings of the 18th International Florida Artificial Intelligence Research Society Conference - Recent Advances in Artifical Intelligence, pp. 721–726, 2005

[12] A. Liu, and D. Salvucci, “Modeling and prediction of human driver behavior,” Proceedings of the 9th International Conference on Human-Computer Interaction, 2001

[13] A. G. Cunningham, E. Galceran, R. M. Eustice, and E. Olson, “MPDM: multipolicy decision making in dynamic, uncertain environments for autonomous driving,” IEEE International Conference on Robotics and Automation, 2015

[14] E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson, “Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment,” Autonomous Robots, vol. 41, no. 6, pp. 1367–1382, 2017

[15] T. Suzuki, S. Sekizawa, S. Inagaki, S. Hayakawa, N. Tsuchida, T. Tsuda, and H. Fujinami, “Modeling and recognition of human driving behavior based on stochastic switched ARX model,” Proceedings of Joint IEEE Internation Conference on Decision and Control and European Control Conference, pp. 5095–5100, 2005

[16] D. Sadigh, K. Driggs-Campbell, A. Prugelli, W. Li, V. Shia, R. Bajcsy, A. L. Sangiovanni-Vincentelli, S. S. Sastry, and S. A. Seshia, “Data-driven probabilistic modeling and verification of human driver behavior,” Formal Verification and Modeling in Human-Machine Systems: Papers from the AAAI Spring Symposium, pp. 56–61, 2013

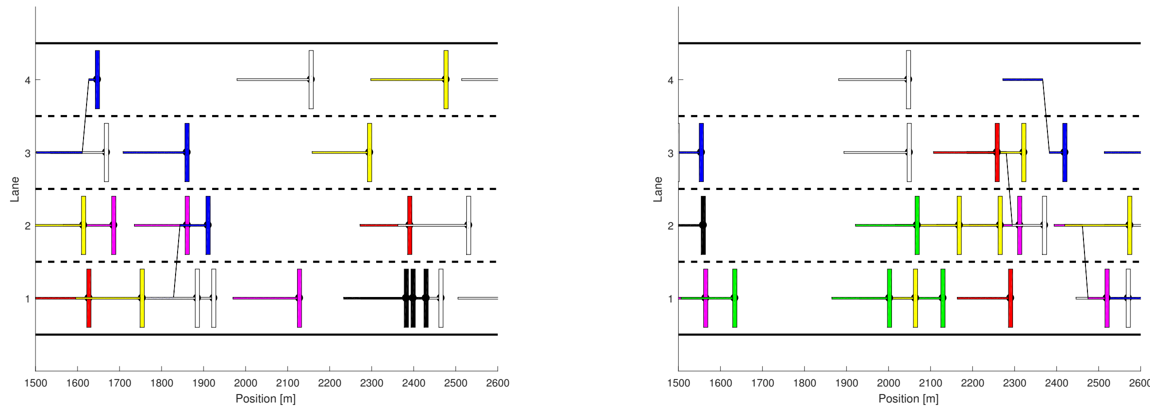
[17] T. Kumagai, Y. Sakaguschi, M. Okuwa, M. Akamatsu, “Prediction of driving behavior through probabilistic inference,” Proceedings of the 8th International Conference of Engineering Applications of Neural Networks, pp. 117–123, 2003

[18] J. Forbes, T. Huang, K. Kanazawa, and S. Russell, “The BATmobile: towards a bayesian automated taxi,” Proceedings of the 14th International joint Conference on Artificial Intelligence, vol. 2, pp. 1878–1885, 1995

[19] T. Gindele, S. Brechtel, and R. Dillmann, “A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments,” 13th International IEEE Annual Conference on Intelligent Transportation Systems, pp. 16225–1631, 2010

[20] J. Maye, R. Triebel, L. Spinello, and R. Siegwart, “Bayesian online learning of driving behavior,” IEEE International Conference on Robotics and Automation, 2011

[21] P. Angkititrakul, C. Miyajima, and K. Takeda, “Analysis and predic-



(a) Scenario at time 106 seconds

(b) Scenario at time 130.5 seconds

Figure 4. *Exemplary driving event when all drivers are controlled by the proposed model.*

tion of deceleration behaviour during car following using stochastic driver-behavior model," 15th IEEE Conference on Intelligent Transportation Systems, pp. 1221–1226, 2012

[22] A. Doshi, B. T. Morris, and M. M. Trivedi, "On-road prediction of drivers intent with multimodal sensory cues," IEEE Pervasive Computing, vol. 10, no. 3, pp. 22–34, 2011

[23] S. Bonnin, T. H. Weisswange, F. Kummert, and J. Schmüdderich, "Accurate behavior prediction on highways based on a systematic combination of classifiers," IEEE Intelligent Vehicles Symposium, pp. 242–249, 2013

[24] M. Garcia Ortiz, F. Kummert, and J. Schmüdderich, "Prediction of driver behaviour on a limited sensory setting," 15th International IEEE Conference on Intelligent Transportation Systems, pp. 638–643, 2012

[25] D. Sun, L. Elefteriadou, "Research and implementation of lane-changing model based on driver behavior," Transportation Research Record: Journal of the Transportation Research Board, no. 2161, pp. 1–10, 2010

[26] H. M. Mandalia, and D. D. Salvucci, "Using support vector machines for lane-change detection," Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 49, no. 22, pp. 195–1969, 2005

[27] D. Moser, Z. Ramezani, D. Gagliardi, J. Zhou, and L. Del Re, "Risk Functions Oriented Autonomous Overtaking," Asian Control Conference, 2017

[28] N. Nguyen, D. Moser, P. Schrangl, L. Del Re, and S. Jones, "Autonomous overtaking using stochastic model predictive control," Asian Control Conference, 2017

[29] S. J. Russell, and P. Norvig, "Artificial intelligence: a modern approach," Upper Saddle River, New Jersey: Prentice Hall, 2003

[30] F. V. Jensen, "Bayesian networks and decision graphs," Information Sciences and Statistics series (2nd ed.), New York: Springer-Verlag, 2007

[31] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, 1989

[32] B. Kisacanin, "Deep learning for autonomous vehicles," IEEE 47th International Symposium on Multiple-Valued Logic, 2017

[33] I. Guyon, and A. Elisseeff, "An introduction to variable and feature selection," Journal of Machine Learning Research, vol. 3, pp. 1157–1182, 2003

[34] R. E. Neapolitan, "Learning bayesian networks," Prentice Hall, 2003

[35] E. T. Whittaker, "On a new method of graduation," Proceedings of the Edinburgh Mathematical Society, vol. 41, num. 1, pp. 63–75, 1923

[36] P. H. Eilers, "A perfect smoother," Analytical Chemistry, vol. 75, num. 14, pp. 3631–3636, 2003

[37] M. Vilela, C. C. H. Borges, S. Vinga, A. T. R. Vasconcelos, H. Santos, E. O. Voit, and S. S. Almeida, "Automated smoother for numerical decoupling of dynamic models," BMC Bioinformatics, vol. 8, pp. 305–312, 2007

[38] R. Durbin, S. Eddy, A. Krogh, and G. Mitchinson, "Probabilistic models of proteins and nucleic acids," Cambridge University Press, 1998

[39] M. Fellendorf, P. Voritsch, "Microscopic Traffic Flow Simulator VISSIM", in J. Barcelo (ed.) Fundamentals of Traffic Simulation, International Series in Operations Research & Management Science 145, Springer Science+Business Media, LLC 2010

Author Biography

Kurt Pichler received his diploma in mathematics (2005) and his PhD in technical sciences (2014) from the Johannes Kepler University (JKU) Linz. He was with the Institute for Design and Control of Mechatronical Systems of the JKU from 2006 to 2007. Since 2007, he has been with the business area Sensors & Communication of the Linz Center of Mechatronics (LCM) GmbH. His research interests include feature engineering, machine learning, and artificial intelligence.

Sandra Haindl received her MSc in Automation Engineering from the University of Applied Sciences Upper Austria (2015). Since then she has worked in the Research and Development Department at Linz Centre of Mechatronics, Linz. Her work has focused on the development of mechatronic systems and data evaluation.

Daniel Reischl received his diploma in mechatronics (2010) and master in legal and business aspects in technics (2014) from the JKU Linz. Since 2010 he has been with the business area Mechanics & Control of the LCM GmbH and is leading the team adaptronics since 2015. His research interests include (vibration) control, piezoelectric systems, multibody simulation and robotics.

Martin Trinkl is student at the JKU Linz. Since 2014 he has been with the business area Mechanics & Control of the LCM GmbH.