

# Gray-Box Driver Modeling and Prediction: Benefits of Steering Primitives

Jairo Inga, Michael Flad and Gunter Diehm, Sören Hohmann

Institute of Control Systems, Karlsruhe Institute of Technology, Karlsruhe, Germany

jairo.inga@kit.edu; flad@kit.edu; gunter.diehm@kit.edu; soeren.hohmann@kit.edu

**Abstract**—Shared control is a promising approach for designing an Advanced Driver Assistance System, since it unifies the advantages of both manual control and full automation. However, for a true cooperative shared control ADAS the automation has to understand the human and thus a suitable model which describes the driver in the control loop is essential. Our gray-box approach bases on the biological concept that humans realize motion by combining a finite set of motion primitives (we call *movemes*). With the assumption that a driver switches between *movemes* based on perceived information, we propose a Hidden Markov Model which determines the probability of each *moveme* given a certain driving situation. Car turn maneuver experiments show a good approximation of steering trajectories recorded in a driving simulator. A comparison with a black-box model show that the *moveme*-based driver model performs significantly better. In addition, training algorithms are available and the probabilistic approach of the model allows further interpretation of the results.

**Index Terms**—Driver Modeling, Motion Primitives, Hidden Markov Model, Shared Control.

## I. INTRODUCTION

In 2012 3600 people died as a result of traffic accidents on German roadways [1]. The total sum of economic losses related to the traffic accidents for only this one year and country sum up to 30 billion euro. According to [2] 75% of all traffic accidents are primary related to failures of the drivers. Based on this data it seems reasonable to support drivers with a high level of automation (e.g. automatic lane keeping). However, it is widely known from the automation of aircraft cockpits that there is the risk of the "out-of-the-loop" problem [3] with a high level of automation. This means that the driver can no longer supervise the vehicle as he is no longer involved in the vehicle control task. In addition, [4] shows that in a highly automated vehicle the driver cannot adequately resume to manual control in the case of an emergency like the breakdown of the automation system. Instead of designing systems that take over control of a subtask of the vehicle guidance problem, systems could be designed that only warn the driver in critical situations. But in this case it still fully depends on the driver's actions whether a critical situation can be handled or not. It is clear that neither manual vehicle control nor a high level of automation is the ideal solution. To combine the advantages of both human and machine it is therefore advisable to use a shared control structure in which both the driver and an Advanced Driver Assistance System (ADAS) control the vehicle together and in cooperation [5], [6]. Several shared control ADAS concepts have been proposed

in the literature [7], [8]. A cooperation between driver and machine requires to understand the human partner in a shared control loop. If the human behavior is known, an appropriate controller can be designed in order to achieve the desired task. Moreover, a human-specific model would lead to a higher acceptance. Therefore, one of the essential requirements for the design of a cooperative shared control system is a model of the driver [9]. In contrast to virtual driver models, which only form a control framework to steer a vehicle, this model needs to describe the real steering behavior of the driver. Preferably, the neuromuscular behavior of the driver should be modeled. Several biological experiments suggest that humans realize complex motions by combining a set of motion primitives [10], [11]. This results in a finite set of steering primitives a driver uses to control his vehicle. In this specific scope as a gray-box model for the neuromuscular system of the driver, we model these primitives as autonomous dynamic systems and call them *movemes* in line with [12]. A significant benefit of the *movemes* is that they can comparatively easy be identified using the algorithms submitted in [13] [14] and yield guarantees regarding correct identification. In previous work [15] [16] we proposed a driver model for the steering task which utilizes these *movemes* to generate a trajectory of driver steering motion. Thereby a *switching controller* determines the *moveme* sequence the driver applies to control the vehicle. This switching controller was modelled using an Model Predictive Control (MPC) framework. It is shown that the model can adequately describe the driving behavior [16]. However, due to the MPC framework the use of the model in a practical ADAS application is difficult, since the calculation time in a real-time environment is very high [15]. In addition, the driver model yields a deterministic prediction of the driver steering actions which does not allow any further interpretation.

In this paper we propose an alternative approach for the switching controller which describes the driver's selection of *movemes* using a Hidden Markov Model (HMM), being an appropriate model for real world tasks. For a given driving situation our switching model yields the probability that a driver applies a certain *moveme*. As this probability is given for all possible *movemes* the model permits an interpretation of the model confidence. Combined, the gray-box driver model allows to understand the neuromuscular system via the *movemes* and the selection strategy between them via the HMM switching model. In addition to being interpretable the gray-box driver model has, in contrast to data-driven black-

box driver models as bare neural networks, better extrapolation capability. This means the proposed HMM can not only describe the driver steering actions for the training data, it can also be used to explain the steering behavior for data which are not part of the training. This is a useful property since it is not possible to include all conceivable scenarios in the training data.

In summary, the goal of the paper is to propose a gray-box driver model which bases on movemes and a novel HMM approach to model the moveme selection. In section II the moveme model is introduced. The model which describes the selections of the movemes is given in section III. In section IV the driver model is validated using real steering data recorded on a driving simulator. Furthermore, the gray-box driver model is compared with a black-box model.

## II. STEERING MOVEMES

The movemes as basic building blocks of drivers' movements are represented as autonomous subsystems that are selected sequentially and that generate the movement trajectories. In general, any kind of system description can be used to formally describe such autonomous subsystems. However, w.l.o.g. we describe the movemes as piecewise affine autoregressive systems without exogenous input of the form

$$w_k = \sum_{j=1}^{n_a(\lambda_k)} a_j(\lambda_k) w_{k-j} + c(\lambda_k), \quad (1)$$

according to [13]. These generate  $T$  samples of an output trajectory  $\{w_k\}_{k=1}^T$ ,  $w_k \in \mathbb{R}$  whereas  $\lambda_k \in \mathcal{M}$ ,  $\mathcal{M} = \{1 \dots N\}$  denotes the currently active moveme at time  $k$  and  $n_a(\lambda_k)$  the order of the autoregressive subsystem. The  $N$  available movemes are thus defined by their parameterizations  $\underline{\vartheta}_\lambda = [a_1(\lambda) \dots a_{n_a}(\lambda) c(\lambda)]^T$ . The measured output is given by  $y_k = w_k + \epsilon_k$ , where  $\epsilon_k$  represents zero-mean white noise.

In [13], [14] a multi-step prediction error-based algorithm for the identification of movemes from the measured output data has been presented and validated experimentally, even for noisy measurements. Thereby the order  $n_a(\lambda)$  is assumed to be uniform for all subsystems. It is shown that applying this identification procedure to a given measured movement trajectory yields a bias-free estimate of the repertoire of used movemes  $\{\hat{\vartheta}_1 \dots \hat{\vartheta}_N\}$ . In this paper we use measurements of the driver's steering wheel angle  $y$  to identify *steering movemes*.

However, in a complete driver model which is able to predict driver movements, it is necessary to generate a proper selection of movemes, hence the state sequence  $\{\lambda_k\}_{k=1}^T$ , from the information of the road and the environment, e.g. road geometry or the relative dynamics of other vehicles. This information which naturally can be perceived by a human driver also serves as inputs  $\underline{u}$  to the switching mechanism in Fig. 1 on that we focus in this paper.

## III. GRAY-BOX LEARNING OF MOVEME SELECTION

A well-known probabilistic approach to model sequential data is the Hidden Markov Model (HMM). HMMs have been

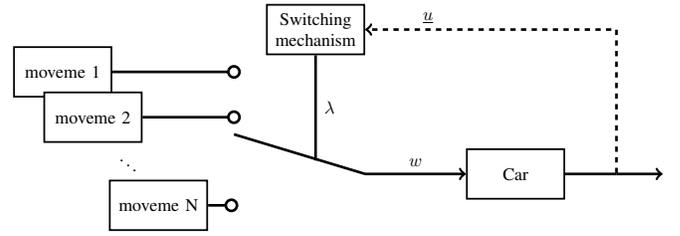


Fig. 1. Driver steering movement model based on a moveme repertoire and a switching mechanism which selects the current moveme  $\lambda$  based on car and environment information  $\underline{u}$ . The active moveme passes a desired steering angle  $w$  on the car

successfully applied in several application fields like speech recognition [17], gesture recognition [18] and bioinformatics [19]. We desire to define the movemes as the possible discrete states of the model and the steering angle as emitted output in order to use a HMM approach. However, a difficulty in utilizing a standard HMM lies in the fact that the moveme at the current time step  $\lambda_t$  only depends on the last moveme  $\lambda_{t-1}$ . Moreover, in the problem at hand, the switching probability must also depend on exogenous inputs  $\underline{u}$ . A HMM extension that includes inputs is given by the Input-Output Hidden Markov Model (IOHMM) [20]. However, even this model is not applicable to our problem due to the fact that it does not allow dynamic subsystems. Therefore, a new HMM model is introduced in this section, which allows input-dependent switching of the movemes.

### A. Model Structure

As mentioned in the last section, the observed output  $y_t$  depends on the current moveme  $\lambda_t$  at time  $t$  as well as on its last  $n_a$  values  $\{y_k\}_{k=t-n_a}^{t-1}$ . With a standard HMM the current moveme  $\lambda_t$  would depend only on the last moveme  $\lambda_{t-1}$ . If we assume that  $\lambda_{t-1} = i$ ,  $i \in \mathcal{M}$ , the next moveme is determined with a markov step. The corresponding transition probabilities from moveme  $i$  to other movemes is given by the  $i$ -th row of the transition probability matrix  $\Phi = \{\varphi_{ij}\}_{i,j=1 \dots N}$  with

$$\varphi_{ij} = P(\lambda_k = j \mid \lambda_{k-1} = i) \quad (2)$$

and

$$\sum_{j=1}^N \varphi_{ij} = 1 \quad \forall i. \quad (3)$$

For the switching mechanism we require that an input vector  $\underline{u}_t$  also has an influence on the current state. The desired variable dependencies are depicted in Fig. 2. In this example, we have autoregressive subsystems of second order. Consequently, the output  $y_t$  depends on the two last values  $y_{t-1}$  and  $y_{t-2}$  and on the current moveme  $\lambda_t$ . In turn,  $\lambda_t$  is determined by the last moveme  $\lambda_{t-1}$  and by the actual input vector  $\underline{u}_t$ .

The resulting model structure is illustrated in Fig. 3. The transition matrix  $\Phi$  is calculated out of the input vector  $\underline{u}$  by means of an artificial neural network with the parameter set  $\Theta$ , i.e.  $\Phi = f_{NN}(\Theta, \underline{u})$ . Since the input sequence  $\mathcal{U} = \{\underline{u}_k\}_{k=1}^T$  is known, we can calculate the current transition matrix  $\Phi_t$  at

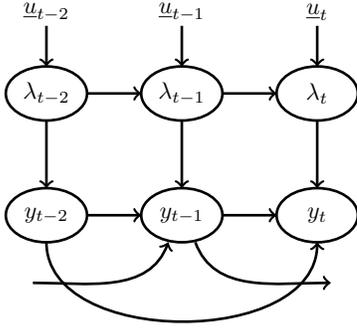


Fig. 2. Representation of the variable dependencies for a model with AR-subsystems of second order with a Dynamic Bayesian Network

each time  $t$ . As a consequence, the transition matrix becomes time-variant and its components are given by

$$\varphi_{ij,t} = P(\lambda_t = j \mid \lambda_{t-1} = i, \underline{u}_t). \quad (4)$$

We now introduce the state probability distribution  $\zeta_t = [\zeta_{1,t} \ \cdots \ \zeta_{N,t}]$  with

$$\zeta_{j,t} = P(\lambda_t = j \mid \{\underline{u}_k\}_{k=1}^t). \quad (5)$$

The variable  $\zeta_{j,t}$  gives the probability of being in a certain state  $j$  at time step  $t$ , given the input sequence until  $t$ . The probability distribution  $\zeta_t$  is calculated with

$$\zeta_t = \zeta_{t-1} \Phi(\underline{u}_t) \quad (6)$$

with

$$\sum_{j=1}^N \zeta_{j,t} = 1 \quad \forall t. \quad (7)$$

The probability distribution therefore fulfills the markov property.

The next step is to use the information of the state distribution to select a moveme. Several approaches are possible, however, we decide to select the moveme with the currently highest probability. We perform the moveme selection by means of an operator  $\mathcal{A}$ , that is

$$\lambda_t = \mathcal{A}(\zeta_t) = m^* \in \mathcal{M} \mid \zeta_{m^*,t} > \zeta_{m,t} \quad \forall m \neq m^* \quad (8)$$

The model output is then determined from the autoregressive system equation (1) as described in Sec. II. The model parameters which have to be trained correspond to the parameter set  $\Theta$  of the network  $f_{NN}(\Theta, \underline{u})$ , which consists of the weights of all neuron connections. The learning algorithm is described in the following subsection.

### B. Learning Algorithm

Available data to train the model are the input sequence  $\mathcal{U} = \{\underline{u}_k\}_{k=1}^T$  and the observed output sequence  $\mathcal{Y} = \{y_k\}_{k=1}^T$ . The parameter set to be trained is located in the network  $f_{NN}(\Theta, \underline{u})$ . Nonetheless, the time-variant transition matrix  $\Phi$ , i.e. the output of the neural network is not known a-priori. Hence, typical training methods for neural networks like the backpropagation algorithm cannot be applied, since we would

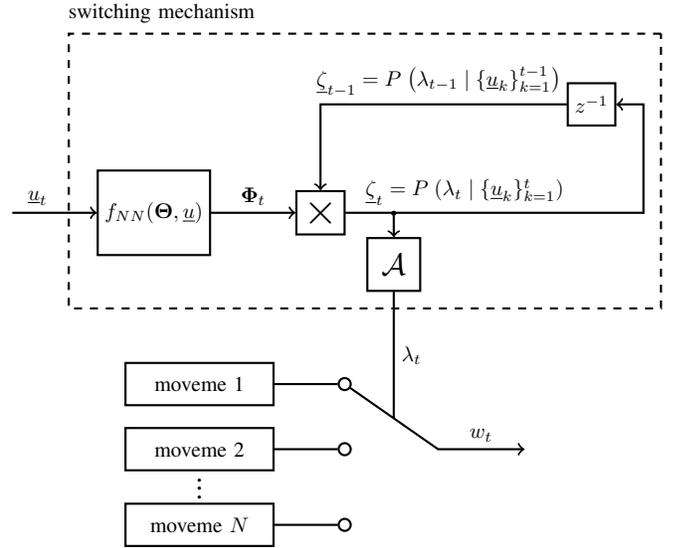


Fig. 3. Gray-box moveme driver model

need fixed input-output pairs  $\{\underline{u}_t, \Phi_t\}$  for all time steps for the net. To circumvent this problem, we train the model parameter set  $\Theta$  with a maximum likelihood estimation (MLE), i.e. we search for the model parameter set  $\hat{\Theta}$  that maximizes the likelihood of the observed output sequence  $\mathcal{Y}$ , given the input data sequence  $\mathcal{U}$ . The likelihood is defined as

$$\mathcal{L}(\Theta) = P(\mathcal{Y} \mid \mathcal{U}; \Theta) \quad (9)$$

So the MLE seeks

$$\hat{\Theta} = \arg \max_{\Theta} \mathcal{L}(\Theta). \quad (10)$$

The MLE is done by means of an expectation-maximization (EM) algorithm. The EM-algorithm is a well-known iterative procedure to find the maximum (log)-likelihood of a statistical model. Its convergence has been proven in [21].

## IV. EXPERIMENTAL RESULTS

In this section we show the applicability of the proposed gray-box driver model based on movemes using real measurement data. In addition, we give a first benchmark of our results with the performance of a black-box model built on a dynamic artificial neural network solely.

### A. Experimental Setup

We measure real steering trajectories in a static driving simulator equipped with an active steering wheel and the CarMaker realtime simulation environment by IPG Automotive GmbH. The maneuver we regard is a left-turn. The road consists of a straight road, followed by a 90 degrees curve with a constant radius of 120 meters and another straight road at the end of the curve. The measurements of the steering wheel angle  $y$  are done by an incremental encoder of 40000 increments per full rotation at a sampling frequency of 10 Hz. This choice is due to preceding analysis which showed that human steering movements show relevant frequencies up to 3.5

TABLE I  
MEASURED VALUES IN THE TEST MANEUVER

Input	Unit	Description
$u_{1,k}$	$\frac{1}{m}$	(Road curvature at time $k + 5$ ) $\cdot 100$ , where $k + 5$ corresponds to 0.5 seconds in future
$u_{2,k}$	$m$	Deviation distance of the front axle center to the lane center
$u_{3,k}$	$rad$	Angle between the vehicle direction and a preview point on the lane center 15m ahead

Hz. In addition, we measure the following three values of the road and environment information which we will use as model inputs  $\underline{u}$ . The measured quantities are the road curvature 0.5 seconds ahead, the deviation distance from the lane center and the angle between the vehicle direction and a preview point on the lane center 15 meters ahead. The road curvature  $u_1$  is scaled by a factor of 100 to improve numerical accuracy. The input quantities are noted in Table I.

Four test runs at a constant speed of 50 km/h were performed for one subject. The test person was a graduate student of 23 years with average driving skills. He was asked to hold the steering wheel in a quarter-to-three hand position and to drive the road like in a normal situation. Afterwards, the movemes of the resulting four steering trajectories M1-M4 were determined with the identification algorithm in [14]. The identification results yield a total of 34 different movemes. Due to the fact that the movemes come from the same maneuver and hence some of them have similar parametrization, it seems reasonable to cluster the movemes in groups. The clustering is done by means of a k-means algorithm, in which we choose to have 10 moveme groups. The number of movemes in each group is given in Table II. The cluster centroids define the parameters of the 10 new movemes which we now set as the moveme repertoire for this maneuver. The artificial neural network we choose for the function  $f_{NN}(\Theta, \underline{u})$  of the gray-box model is a standard multilayer perceptron (MLP) with one hidden layer with 30 neurons.

### B. Training Procedure and Results

The training is first carried out with the input  $\mathcal{U}$  and output sequence  $\mathcal{Y}$  of 2 steering trajectories M1 and M2. The input sequence and the training results are shown in Fig. 4. The training results include the observed steering trajectory and the output trajectory of the model. The data is overall well fit by the model. The state probabilities for all movemes that generate the model output are also illustrated in the figure. In many time intervals there is one most probable moveme. Only in some situations (e.g. around 10s and 40s) there is not a clearly dominant moveme. Remember that the output is generated by the selected moveme as determined in (8). Even at those times, the generated output still approximates the data

TABLE II  
NUMBER OF MOVEMES PER GROUP AFTER CLUSTERING PROCEDURE,  
MEAN AND STANDARD DEVIATION

Cluster	1	2	3	4	5	6	7	8	9	10	$\mu$	$\sigma$
Movemes	2	9	4	2	5	1	2	2	3	4	3.4	2.32

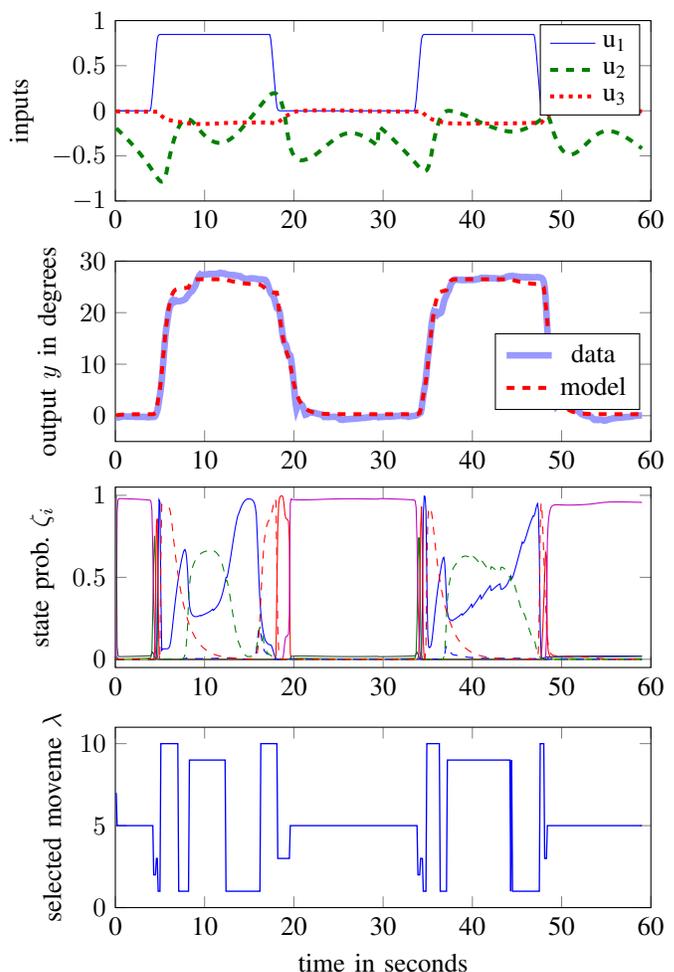


Fig. 4. Training results with gray-box moveme model from M1 and M2 (concatenated). The third subplot shows the probability of all movemes over time, where each color and linestyle represents a different moveme index.

well. Afterwards, the trained model is tested with the inputs of another measurement M3. The inputs and the model output compared to the measured steering trajectory are depicted in Fig. 5. The model generates an output that approximates the measured steering trajectory adequately. In order to give enough evidence for the correct functioning of the model, a 2x2 cross-validation is carried out with the 4 maneuvers M1-M4. This means that the model is trained with 2 maneuvers and tested with the other 2 afterwards. The procedure is done for all possible combinations that we call CV1 to CV6 as shown in Table III.

Now we train a black-box model consisting of a NARX artificial neural network (NARX-ANN) with a hidden layer and 30 neurons [22]. We choose a NARX-ANN since it also considers output feedback and hence can be better compared

TABLE III  
MANEUVER COMBINATIONS FOR CROSS-VALIDATION

	CV1	CV2	CV3	CV4	CV5	CV6
Trained with	M1,M2	M1,M3	M1,M4	M2,M3	M2,M4	M3,M4

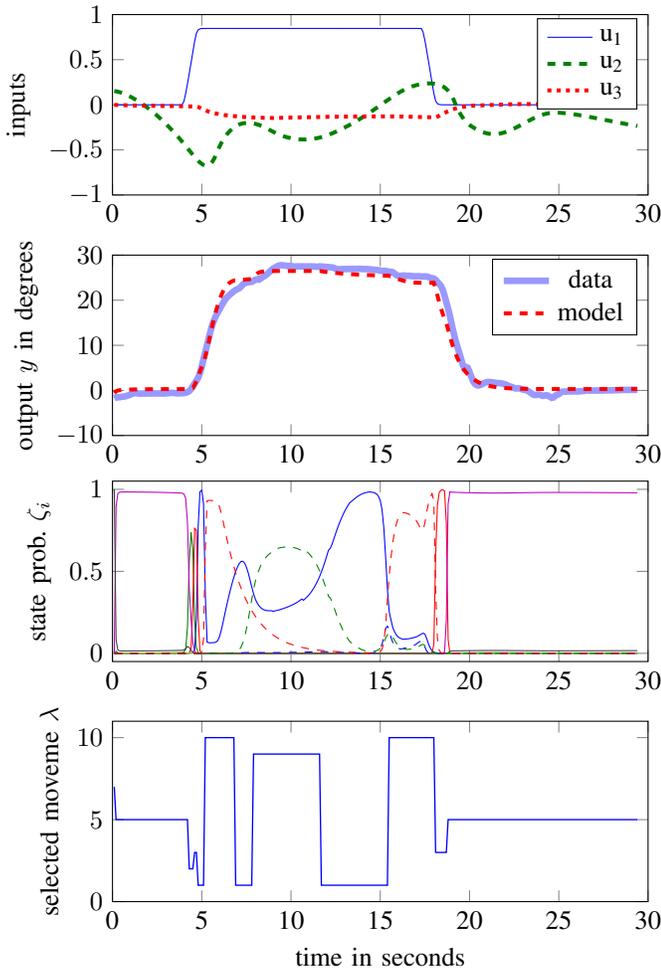


Fig. 5. Test results with gray-box moveme model from M3 after training with M1 and M2

to our model than conventional static MLPs. The inputs of this neural network are the 3 inputs from before plus the last two values  $y_{k-1}$  and  $y_{k-2}$  of the output data at each time step  $k$ . The closed-loop training results are illustrated in Fig. 6. This means the model output is fed to the NARX-ANN input instead of the measured target values. It can be seen that the data is overall well fit by the model. However, some oscillations can also be observed at the beginning. In the same manner as before, the trained model is tested with the 3 measured inputs of maneuver M3. The results can be seen in Fig. 7. It can be stated that the model performance decreases as soon as different input combinations are fed into the model. For example,  $u_3$  is negative in the first time steps of the training data whereas it is positive in the test data. Nevertheless, the same cross-validation as before is done with the black-box model. The results of the cross-validation of the moveme-based gray-box model and the ANN as black-box model are summarized in Table IV and in Fig. 8. Note the log-scale of the y-axis. The root mean squared error (RMSE) in degrees between the measured steering trajectory and the model output is noted for the training with 2 maneuvers. The mean RMSE

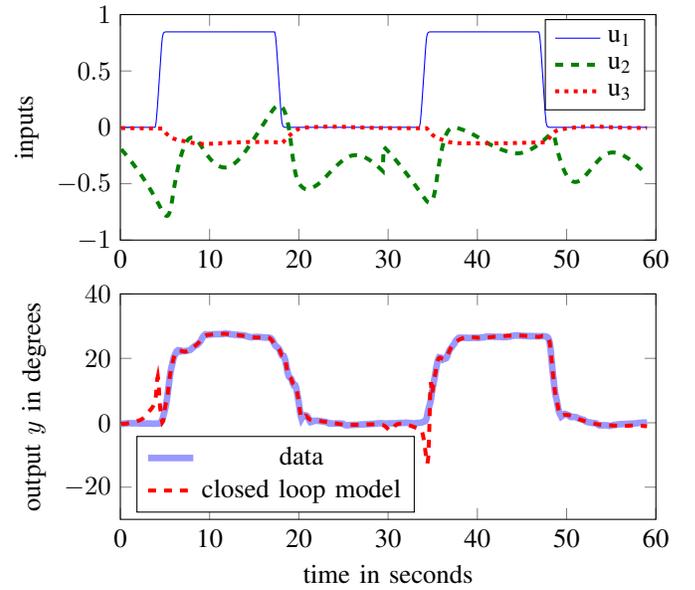


Fig. 6. Training results with NARX-ANN from M1 and M2

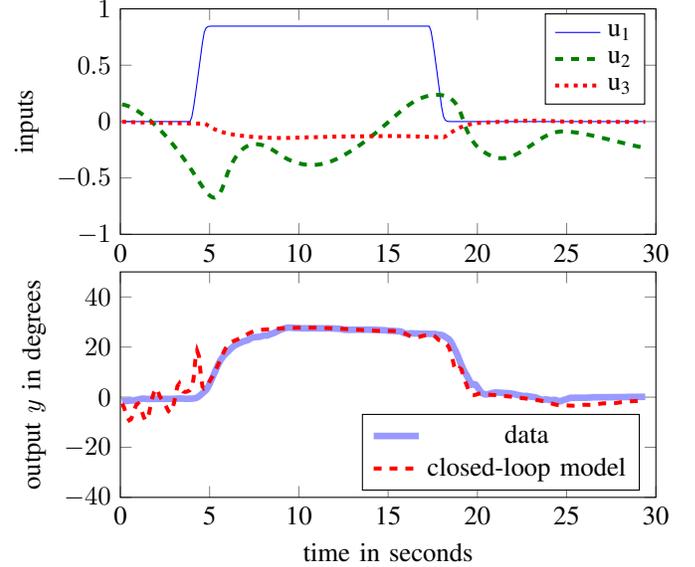


Fig. 7. Test results with NARX-ANN from M3 after training with M1 and M2

of the 2 testruns with the other 2 maneuvers done afterwards are also included.

The moveme-based gray-box model approximates the output trajectories better than the black-box model. Especially in the test with non-training data the presented gray-box model shows a significantly better extrapolation capability.

## V. CONCLUSION

In this paper we propose a gray-box driver steering model. The proposed concept is to use movemes to reproduce the driver's steering actions and model the selection of them using a HMM. In the experiment the gray-box moveme model can replicate the driver adequately and has a comparable

TABLE IV

ROOT MEAN SQUARED ERROR (RMSE) OF STEERING TRAJECTORIES AFTER TRAINING AND MEAN RMSE DURING TEST FOR MOVEME MODEL AND ANN

	CV1	CV2	CV3	CV4	CV5	CV6	$\emptyset$
<b>training</b>	1.08	1.33	1.41	4.79	0.97	1.54	1.85
<b>test (mean)</b>	1.47	1.72	1.58	1.54	1.21	8.32	2.64
<b>training ANN</b>	1.83	4.87	4.8	1.76	0.69	0.31	2.38
<b>test ANN (mean)</b>	2.92	4.71	9.98	2.45	1.49	26.1	7.94

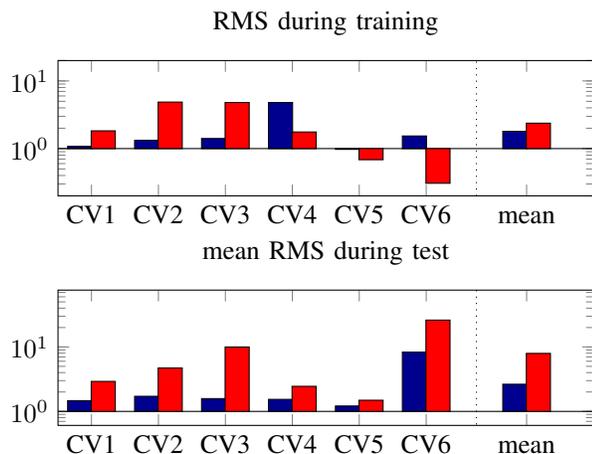


Fig. 8. Root mean squared error (RMSE) of steering trajectories after training and test for moveme model and ANN. Values in lower graph represent mean values of both testruns. The blue bars represent the results with the gray-box moveme model and the red bars represent the black-box ANN model.

performance to the reference ANN black-box model.

The benefits of the moveme gray-box model are the following:

- The existence of an applicable approach to determine the model parameters by one algorithm for the identification of the movemes and a second for the training of the HMM. The convergence of both algorithms is guaranteed.
- The gray-box moveme model can be used to predict future steering behavior of the driver for a short time horizon by simply propagating the current moveme. Moreover, if road preview is available, a prediction for a longer time horizon is possible by running the HMM switching model.
- The probabilities of the movemes allow the statement of a model confidence. In addition, further interpretation of the selected moveme is possible since the selected model inputs are regarded as possible sensory information of the driver.
- A real-time implementation of the model is simple as the calculation time needed to execute the model is very small compared with an MPC approach.

In future work we plan to use this model in a shared control ADAS based on a game theoretic framework. Aside from the driving context, our model approach can as well be applied to other domains, for example aerospace applications and human-robot interaction like handling heavy weights with robot assistance in an industrial environment.

## REFERENCES

- [1] "Accident history on german roadways for 2012," Federal Statistical Office, Tech. Rep., 2013.
- [2] P. M. Salmon, M. A. Regan, and I. Johnston, "Human error and road transport: Phase one - a framework for an error tolerant road transport system," Monash University Accident Research Centre, Tech. Rep., 2005.
- [3] E. Wiener and R. Curry, "Flight-deck automation: promises and problems," *Ergonomics*, vol. 23 (10), pp. 995–1011, 1980.
- [4] F. Flemisch, J. Kelsch, C. Löper, A. Schieben, J. Schindler, and M. Heesen, "Cooperative control and active interfaces for vehicle assistance and automation," in *FISITA World automotive Congress; Munich*, 2008.
- [5] M. S. Young, N. A. Stanton, and D. Harris, "Driving automation: Learning from aviation about design philosophies," *International Journal of Vehicle Design*, vol. 45 (3), pp. 323–338, 2007.
- [6] M. Mulder, D. Abbink, and E. Boer, "Sharing control with haptics: Seamless driver support from manual to automatic control," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 54 (5), pp. 786–798, 2012.
- [7] P. Griffiths and R. Gillespie, "Shared control between human and machine: haptic display of automation during manual control of vehicle heading," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS '04. Proceedings. 12th International Symposium on*, March 2004, pp. 358–366.
- [8] M. Mulder, D. A. Abbink, and E. R. Boer, "The effect of haptic guidance on curve negotiation behavior of young experienced drivers," in *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2008, pp. 804–809.
- [9] M. Flad, J. Otten, S. Schwab, and S. Hohmann, "Necessary and sufficient conditions for the design of cooperative shared control," in *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2014, pp. 1253–1259.
- [10] F. Mussa-Ivaldi and S. Solla, "Neural primitives for motion control," *IEEE Journal of Oceanic Engineering*, vol. 29 (3), pp. 640 – 650, 2004.
- [11] C. B. Hart and S. F. Giszter, "A neural basis for motor primitives in the spinal cord," *The Journal of Neuroscience*, vol. 27, pp. 1322–1336, 2010.
- [12] D. Del Vecchio, R. M. Murray, and P. Perona, "Decomposition of human motion into dynamics based primitives with application to drawing tasks," *Automatica*, vol. 39, pp. 2085–2098, 2003.
- [13] G. Diehm, S. Maier, M. Flad, and S. Hohmann, "Online identification of individual driver steering behaviour and experimental results," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2013, pp. 221–227.
- [14] —, "An identification method for individual driver steering behaviour modelled by switched affine systems," in *IEEE 52nd Annual Conference on Decision and Control (CDC)*, 2013, pp. 3547–3553.
- [15] M. Flad, C. Trautmann, G. Diehm, and S. Hohmann, "Individual driver modeling via optimal selection of steering primitives," in *19th IFAC World Congress*, 2014, pp. 6276–6282.
- [16] —, "Experimental validation of a driver steering model based on switching of driver specific primitives," in *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2013, pp. 214–220.
- [17] 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.
- [18] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden markov models," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1992, pp. 379–385.
- [19] J. Stanke, "Gene prediction with a hmm," Ph.D. dissertation, Georg-August-Universität Göttingen, 2003.
- [20] Y. Bengio and P. Frasconi, "Input-output hmms for sequence processing," in *IEEE Transactions on Neural Networks*, vol. 7, no. 5, 1996, pp. 1231 – 1249.
- [21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-likelihood from incomplete data via the em algorithm," *Royal Statistical Society B*, vol. 39, pp. 1–38, 1977.
- [22] S. Chen, S. A. Billings, and P. M. Grant, "Non-linear system identification using neural networks," *International Journal of Control*, vol. 51, no. 6, pp. 1191–1214, Jan. 1990.