A Path Prediction Model based on Multiple Time Series Analysis Tools used to Detect Unintended Lane Departures

John Dahl[1,2,3], Gabriel Rodrigues de Campos[1] and Jonas Fredriksson[3]

***Abstract*— In this paper, a path prediction model is presented and used to detect unintended lane departures caused by erroneous driving behaviours. The prediction model is inspired by the concept of a linear vector autoregressive model that is commonly used for multiple time series analysis. The original concept is extended to allow sparse historic sampling, which is shown to reduce the computational complexity while maintaining the predictive performance. A real world data set is used to derive and validate the proposed model, for which the performance is benchmarked against a kinematic model. The results show that the proposed model can improve the true-positive rate by 18% and reduce the false-positive rate by 34%, with respect to a constant velocity model and for a prediction horizon of 1.75 s.**

I. INTRODUCTION

Automotive Collision Avoidance Systems (CAS) have been a popular research topic for decades, covering applications such as automated braking, Lane-Keeping Aid (LKA) and collision avoidance at intersections. In recent times, automotive manufacturers have industrialized such techniques and they are now important cornerstones in the pursuit of reducing traffic fatalities caused by driver errors. However, despite the trend of falling numbers in traffic fatalities, it remains a fact that driver errors are still a severe issue. According to [1], in 2016 the most common cause of accidents with fatal outcome (30%) within the European countries is single vehicle accidents. Furthermore, these accidents tend to happen in rural areas where the speeds can be considerably high, and where run-off-road and collision with standing still objects can have dramatic outcomes.

An effective way of preventing single vehicle accidents is therefore to ensure that the vehicle does not unintentionally depart from the lane. In a CAS, and in particular for a lane-departure detection system, prediction models are used for Threat Assessment (TA), where future behaviors and potential driving errors are detected. The output of the TA is used to make a decision on whether the driver should be assisted with an automated steering maneuver. However, it is challenging in practice to derive an accurate prediction model that can adapt to various situations, drivers and sensor uncertainties, see e.g., [2], [3] for thorough reviews on prediction models and threat assessment techniques.

A. Related work

Different tools and approaches have been proposed in the literature regarding prediction models for threat-assessment and decision making purposes in safety critical situations.

For instance, some prediction models can, in some cases, predict the future threat level without providing any insight in how the lane departure will occur. For example, the unintended lane departure problem can be seen as a logistic regression problem [4], [5], [6]. In [4] a Support Vector Machine (SVM) is used to make logistic regression on a data set derived from human driving in a simulator. The work is extended in [5], where the SVM is replaced by a Neural Network (NN). A similar approach is used in [6], where an NN is shown to outperform a kinematic based model using a real world driving data set. Despite the high predictive performance of the proposed approaches, their outputs do not reveal the criticallity of the situation, i.e., the model output is the same for a lane departure with low heading angle and a lane departure with a high heading angle towards the lane marker.

Other works derive path prediction models that provide an explicit future trajectory explaining how the vehicle will depart from the lane. A simplistic approach to path prediction is to assume that the motion of a vehicle is solely dependent on its kinematics, i.e., neglecting the driver input over the prediction horizon [7]. Examples of commonly used kinematic models are the Constant Velocity model (CV), Constant Turn Rate and Velocity model (CTRV), [8], and Constant Turn Rate and Acceleration (CTRA) model, [8], [9]. Kinematic models are typically best suited for short prediction horizons, due to their inability to adapt to changes in driver behavior over the prediction horizon. However, [9] has shown that by mixing a CTRA model and a simple maneuver recognition module (consisting of a set of predefined trajectories), the prediction performance can be improved also for longer prediction horizons. In [10] authors proposed a learning-based approach that leverages a personalized driver model, obtained by combining a Gaussian mixture model and a hidden Markov model. This personalized driver model is then used for the design of an online model-based path predictor, which was benchmarked against a basic Time-to-Lane-Crossing (TLC) method. In [11], a Recurrent Neural Network (RNN) in combination with a deep ensemble technique is used to predict the future path with corresponding uncertainty of obstacle vehicles. The same problem is addressed in [12] where the performance of a NN, RNN and a Hybrid NN is benchmarked using a simulation environment. But while

Work supported by Zenuity AB, Vinnova and AI Innovation of Sweden.

[1] Zenuity, Gothenburg, Sweden. name.surname@zenuity.com
[2] AI Innovation of Sweden, Gothenburg, Sweden.
[3] Electrical Engineering Department, Chalmers University of Technology, Gothenburg, Sweden. name.surname@chalmers.se

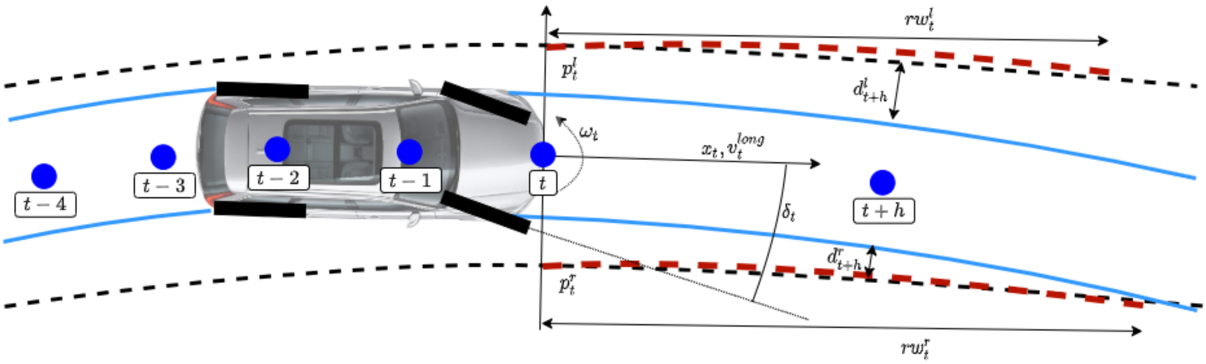

Fig. 1. A vehicle departing from the lane. The blue solid lines indicates the trajectory of the vehicle relative the lane markers, depicted as black dashed lines. The red dashed lines are approximations of the lane markers, which are derived from the front camera system. The blue filled circles indicate the center of the vehicle for different time instances. The goal is to develop a prediction model that can predict the distances to the left and right lane markers, h steps ahead, given the information available at current time instance t.

such learning-based models achieve high predictive performance, they suffer in terms of real-time performance due to the numerous computations for each forward pass.

B. Contribution statement

The goal of this work is to derive a computationally effective prediction model that can accurately predict unintended lane departures using in-vehicle sensor data. The prediction model is inspired by tools commonly used in multiple time series analysis [13]. Moreover, the prediction model is developed for prediction horizons up to 1.75 seconds and is evaluated using a real world data set including several passenger vehicles and drivers. The main contributions are:

- a multiple linear regression (MLR) prediction model, based on multiple time series, that can predict the distance to the left and right side lane-marker h-steps ahead;
- a novel MLR model formulation leveraging Sparse Historic Sampling (SHS);
- preliminary results showing that SHS can reduce the number of coefficients in the MLR model while maintaining the predictive performance;
- results and discussions that show that the MLR models outperformed, in general, the kinematic model for all prediction horizons.

C. Outline

The remainder of the paper is organized as follows: Section II introduces the real world data set, while Section III describes the concept of multiple time series analysis. Section IV covers the threat assessment and decision making logics, while the experimental results are presented in Section V. Finally, Section VI presents our conclusions and future perspectives.

II. Real world driving data set

This work makes use of a vast data set, property of Zenuity AB, as a basis for model evaluation. It has been collected by professional drivers driving a fleet of test-vehicles under different weather conditions, road types and countries such as Sweden, Germany and China.

A. Input signals definition

The test-vehicles are equipped with a forward looking vision system, including software to estimate the geometry of the lane markings relative to the ego-vehicle, as illustrated in Fig. 1. Here p_t^l and p_t^r denote the approximate lane-marker polynomials on the left and right side, respectively. They can generically be written as:

$$p_t^\diamond(x_t) = a_{0,t}^\diamond + a_{1,t}^\diamond x_t + a_{2,t}^\diamond x_t^2 + a_{3,t}^\diamond x_t^3, \qquad (1)$$

where x_t is the longitudinal distance, t the time instance, and the $\diamond$ symbol a wildcard indicating the side of the vehicle. In addition to the polynomials, the vision system also estimates the range of view $rw_t^\diamond$, which is the longitudinal distance for which the polynomials are valid. Furthermore, the yaw rate ω_t, front wheel angle δ_t and longitudinal velocity v_t^{long} are obtained from the ego-vehicle's CAN-bus. Each signal is sampled with a sampling frequency of $f_s = 1/T_s = 40$ Hz.

B. Data selection

The quality of the signals in the data set varies with respect to different factors such as weather conditions or worn lane markers. Since the threat assessment method proposed later in this paper is data-driven, it is paramount that the data is consistent and representative of the chosen scenario, i.e., that it properly reflects lane departure situations.

Let the *extracted data set* be a subset of the real world data set, where each data sample fulfills the following criteria:

- the lane width is not wider than 4 m;
- the curve radius of the road is larger than 250 m;
- the longitudinal velocity is greater than 60 km/h;
- no turn indicator is used when departing from lane;
- no intentional lane change is performed within a 4 s period after a lane departure;

Since the extracted data set should reflect unintended lane departures on rural and highway roads, these criteria are effectively excluding situations such as regular lane-changes, queuing, forks, merges and roundabouts.

C. Data annotation

The annotation of the extracted data set is straightforward since each signal is a time series. Given a time instant t and a prediction horizon of H seconds, the future lateral distance to the lane marker is simply $d^{\diamond}_{t,h} = p^{\diamond}_{t+h}(0) = a^{\diamond}_{0,t+h}$, where $h = H/T_s$. The annotated data is divided into a non-event data set $|\mathcal{B}| = 3000$ and an event data set $|\mathcal{C}| = 12645$, where the notation $|\ |$ indicates the total number of time series sequences in the set. The non-event set contains time series sequences of length 11 s with only normal driving in lane, i.e., where no lane departures occur, and is used to evaluate the false-positive performance. The event set contains short snippets of time series where each snippet ends with a lane departure, i.e., the shortest distance among $d^{\diamond}_{t,h}$ is less or equal to 0. The snippet's duration is a design parameter affecting the balance between normal driving and driving leading to a lane departure. The snippet's length used in this work is proportional to the prediction horizon, and is given by $4H$ s. The event set is divided into an estimation set $|\mathcal{C}^e| = 10645$ used for model coefficient estimation, a calibration set $|\mathcal{C}^c| = 1000$ used for performance calibration and a test set $|\mathcal{C}^t| = 1000$ used for testing the model performance. Note that $\mathcal{C}^e, \mathcal{C}^c$ and $\mathcal{C}^t$ are mutually exclusive, and that all data samples have been standardized.

III. Multiple Time Series Analysis

Define a time series as a sequence of states Y_t ordered by time instance $t \in \mathbb{N}$, given as:

$$Y_t = [y_{t,1}, y_{t,2}, \ldots, y_{t,K}]^\intercal, \tag{2}$$

where $y_{t,k},\ k \in [1, K]$, denotes the k^{th} signal in the time series and $\intercal$ the matrix transpose operator. Notation-wise, the time series is called univariate if $K = 1$ and multivariate if $K > 1$, i.e., a vector valued time series. The behavior of the time series is determined by the underlying process that is observed. In the typical case, this process is an unknown function that exhibits random properties. However, tools for multiple time series analysis can be used to estimate a function that mimics the properties of the process, which in turn can be used to make time series predictions.

A classical approach to derive such a model is to assume that the observed process is Vector Auto-Regressive (VAR), implying that any state of the series can entirely be described as a function of the previous states, see [13] and [14]. If the function is linear, a VAR process of order p can be defined as:

$$Y_t = \sum_{i=1}^{p} A_i Y_{t-i} + W_t, \tag{3}$$

where the dependency on the p previous states is determined by the weight-coefficients in the matrices:

$$A_i = \begin{bmatrix} \theta^{(i)}_{1,1} & \cdots & \theta^{(i)}_{1,K} \\ \vdots & \ddots & \vdots \\ \theta^{(i)}_{K,1} & \cdots & \theta^{(i)}_{K,K} \end{bmatrix}. \tag{4}$$

The process is driven by white noise $W_t \in \mathbb{R}^K$, with $\mathbb{E}[W_t] = 0$, $\mathbb{E}[W_t W_t^\intercal] = \Sigma \in \mathbb{R}^{K \times K}$ and $\mathbb{E}[W_t W_s^\intercal] = 0$ for $t \neq s$.

A. The 1-step prediction model

As mentioned earlier in this section, the function describing the underlying process is typically unknown, i.e., the order p and the coefficients in A_i are unknown. A common way to find a model that can accurately represent the underlying process it to estimate a prediction model, based on an initial guess on the model's order d and the time series observations, and evaluate how well the model predicts the future state of the time series. The best model is found by sweeping the value of the order d and compare the result for each corresponding model.

Lets introduce a 1-step prediction model of order d given as:

$$\hat{Y}_{t+1} = \sum_{i=0}^{d-1} \hat{A}_i Y_{t-i}, \tag{5}$$

where $\hat{A}_i$ are the estimates of the coefficient matrix and $\hat{Y}_{t+1}$ is the prediction for time instance $t + 1$ given observations up to time instance t.

B. The h-step prediction model

An h-step predictor can be derived by using the 1-step predictor (5), *recursively*, h times with a fixed t [15]. Hence, after d prediction steps, the $d + 1$ prediction step is solely a function of previous prediction steps, since observations of the time series are only available up to time instance t. The recursive model predicts every time instance up to $t+h$ and can be interpreted as a proper future path. An alternative approach is to use the *direct* h-steps predictor expressed as:

$$\hat{Y}_{t+h} = \sum_{i=0}^{d-1} \hat{A}_i Y_{t-i}, \tag{6}$$

where the $t + h$ prediction is computed directly and purely on observed samples. Note that the numerical values of $\hat{A}_i$ are different in the two models (5) and (6). A limitation with the direct model is that it only computes one future sample, but in practise it is not a limitation for the application under consideration in this paper, since the prediction model is used within a receding horizon framework. Moreover, the direct prediction model is more efficient in terms of real-time computation and, in terms of predictive performance, both prediction models have shown similar prediction errors for various data sets, see [15], [16]. It is worth mentioning that such literature considers different application domains other than lane-keeping assistance, and therefore no comparison to such works is provided in this paper.

C. The h-step prediction model based on multiple linear regression

In the above prediction model, all historic samples of the time series are sampled in consecutive order, with a uniform sample rate determined by the sample time T_s. However, the time series might be over-sampled, which means that the difference between two consecutive samples is very small. Hence, it might be hard to distinguish the contributions from the signals and the embedded noise. If the signals in the time series are sufficiently filtered, it is possible to down-sample the time series without loosing any valuable information, as long as the Nyquist sampling theorem is fulfilled. Hence, the down-sampling increases the difference in magnitude between two consecutive samples, which reduces the risk of fitting the noise while estimating the model coefficients [17].

In this work, a modified direct prediction model is used, which introduces sparse sampling of historic data. The sparse sampling technique down-samples the time series in real time by using a sample offset pattern, while maintaining the original prediction frequency f_s. Define an offset pattern as:

$$\Gamma = \{\gamma_0, \gamma_2, \ \cdots \ , \gamma_{d-1}\}, \tag{7}$$

where γ corresponds to a time offset expressed in number of samples and d the number of coefficient matrices. Note that the offsets can be chosen freely[1] as long as $\gamma_i \geq 0$ and γ_i remains unique in Γ.

Finally, the MLR direct prediction model with SHS can now be expressed as:

$$\hat{Y}_{t+h} = \sum_{i=0}^{d-1} \hat{A}_i Y_{t-\gamma_i}. \tag{8}$$

D. Model coefficient estimation

This section describes how the prediction model's coefficient matrices A_i, as defined above, are estimated in closed form using a sequence of sampled data. Rewrite (8) in a matrix notation such that:

$$\hat{Y}_{t+h} = BZ_t, \tag{9}$$

where,

$$B = [\hat{A}_0, \dots, \hat{A}_{d-1}] \in \mathbb{R}^{K \times Kd}, \tag{10}$$

$$Z_t = [Y_{t-\gamma_0}^{\intercal}, \dots, Y_{t-\gamma_{d-1}}^{\intercal}]^{\intercal} \in \mathbb{R}^{Kd \times 1}. \tag{11}$$

Now assume there exist N observations of $(\hat{Y}_{t+h}, Z_t)$:

$$Y = [Y_{t+h} \dots Y_{t+N+h}] \in \mathbb{R}^{K \times N},$$
$$Z = [Z_t, \dots, Z_{t+N}] \in \mathbb{R}^{Kd \times N}.$$

The model coefficients is now found by solving the Least Square (LS) problem:

$$\arg\min_B \frac{1}{N} ||Y - BZ||^2, \tag{12}$$

which has the closed form solution:

$$\hat{B} = YZ^{\intercal}(ZZ^{\intercal})^{-1}. \tag{13}$$

Observe that the LS problem consists of K equations that can be solved independently. Let Y_t^k denote the k:th row in Y_t. This yields:

$$Y^k = [Y_{t+h}^k, \dots, Y_{t+N+h}^k], \tag{14}$$

and compute the coefficients of the subsystem k as:

$$\hat{b}^k = Y^k Z^{\intercal}(ZZ^{\intercal})^{-1}, \tag{15}$$

where $\hat{b}^k = \hat{B}^k$.

IV. Threat Assessment and Decision Making

This section covers the implementation details for the proposed MLR path prediction model, as well as a kinematic path prediction model used for benchmarking purposes. In addition, details on the decision making algorithm as well as the methodology for performance calibration are provided.

A. The direct MLR path prediction model

Consider a path prediction model based on (8) obtained by using a multiple time series consisting of 13 signals. The available signals at every time instance t are:

$$Y_t = [a_0^l, a_0^r, a_1^l, a_1^r, a_2^l, a_2^r, a_3^l, a_3^r, \omega, \delta, v, rw^l, rw^r]^{\intercal},$$

where the t notation for the signals has been omitted for the sake of clarity. The goal is to predict the future distance $d_{t+h}^{\diamond}$ with respect to the left and right hand side lane markers h - steps ahead. The prediction model has then two outputs that are linearly dependent on the signals in Y_t, and can be written as:

$$\hat{\mathcal{T}}_{t+h} = \begin{bmatrix} \hat{d}_{t+h}^l \\ \hat{d}_{t+h}^r \end{bmatrix} = \begin{bmatrix} \hat{a}_{0,t+h}^l \\ \hat{a}_{0,t+h}^r \end{bmatrix} = \sum_{i=0}^{d-1} \begin{bmatrix} \hat{A}_i^0 \\ \hat{A}_i^1 \end{bmatrix} Y_{t-\gamma_i}. \tag{16}$$

Numerical estimates of the coefficient row-vectors $\hat{A}_i^0$ and $\hat{A}_i^1$ can easily be obtained by using (15) twice.

B. Kinematic model

Kinematic models are popular since they are easy to implement and fairly accurate under ideal conditions. However, this type of models fall short in presence of dynamic driver behaviours and are sensitive to measurement noise, as they rely on signals from only one time instance. In this work, a Constant Velocity (CV) model is used as a baseline for the performance benchmark. It is worth mentioning that other kinematic models, such as the CTRV model, are also frequently used. However, empirical testing on the considered data set, that mainly represents non-curvy roads, has shown poor performance of the CTRV model for the lane-keeping application. For this reason, the CTRV model is not a part of this work.

1) Constant velocity model: The CV model computes the future distance $d_{t+h}^{\diamond}$ based on the assumption that the lateral velocity remains constant over the prediction horizon, see Alg. 1. Such a model has its strengths in situations where the road is fairly straight and the longitudinal speed varies slowly, such as for the data set considered in this paper.

[1] The definition in (7) allows for sparse sampling of the historic data. Thus, old samples are not even forced to be considered in a consecutive temporal order, which gives the designer the freedom to choose any desired input pattern.

Algorithm 1 Predict the distance to the lane-marker h steps ahead given a constant velocity.

```
procedure CV($v_t^{long}, p_t^{\diamond}, h, T_s$)
    $\psi = \frac{d}{dx} p_t^{\diamond}(0)$
    $v_t^{\diamond,lat} = \sin\psi v_t^{long}$
    $d_{t+h}^{\diamond} = p_t^{\diamond}(0) + v_t^{\diamond,lat} h T_s$
    Return $d_{t+h}^{\diamond}$
```

TABLE I

OFFSET PATTERNS USED FOR SPARSE SAMPLING IN HISTORIC DATA

Label	offset pattern	Depth [s]
Γ_1	{0, 1, 2}	0.075
Γ_2	{0, 1, 2, 3, 4, 5, . . ., 19}	0.5
Γ_3	{0, 1, 2, 3, 4, 5, . . ., 39}	1
Γ_4	{0, 1, 2, 3, 4, 5, . . ., 79}	2
Γ_5	{0, 5, 39}	1

C. Decision making algorithm

Threat assessment and decision-making for a LKA system consists of a prediction model that can foresee the future risk of departing from the lane. Should the safety of the vehicle and its occupants be at risk, an automatic steering intervention is to be triggered, leading the vehicle towards the center of the ego lane. For the sake of simplicity of the language, these automatic steering interventions will be denoted, throughout the remaining of this paper, as *activations*.

In this scope, a path prediction model is used to predict lane departures, which can be detected by checking whether $\hat{d}_{t+h}^{l} \leq \tau$ or $\hat{d}_{t+h}^{r} \leq \tau$ is true, where the threshold τ is a design parameter. Morevoer, for the performance argumentation presented later in this paper, the test data set $\mathcal{C}^t$ is used to compute the number of True-Positive (TP) activations, i.e., how many activations are triggered when needed, and the mean triggering time is denoted by $\bar{t}_h$ for a given τ. A TP is encountered if the prediction model detects a true lane departure up to $2H$ seconds before the departure is taking place. The False-Positive (FP) rate is computed using the non-event set $\mathbb{B}$ given τ, where a FP is encountered if an unwanted activation is triggered, i.e., any activation in the non-event set $\mathbb{B}$. Generally speaking, a well functioning system has a high TP-rate (TPR) and a low FP-rate (FPR) while maintaining a triggering timing close to the prediction horizon $\bar{t}_h \approx H$.

D. Calibration of the mean triggering time

A fair comparison between prediction models in terms of TP rate and FP rate can only be achieved if they are tuned to the same mean triggering time. Notice that, for a perfect prediction model, the mean triggering time, obtained for $\tau = 0$, should be equal to H s. Unfortunately, in practice, there are no such guarantees with respect to the mean triggering time. However, the mean triggering time can be calibrated to $\bar{t}_h = H$ by adjusting the threshold τ. The calibration is individually performed for every model using a small calibration data set $\mathcal{C}^c$, and comparable TP rate and FP rate can be computed using a test set $\mathcal{C}^t$ with the calibrated value of τ^*.

V. IMPLEMENTATION AND RESULTS

The MLR prediction model in (16) is implemented for prediction horizons $H = \{0.5, 0.75, 1, 1.25, 1.5, 1.75\}$ s and for 5 different offset patterns defined as in (7).

See Tab. I for an overview of the simulation setup. The *depth* indicates the oldest sample, in seconds, used in each offset pattern. The first four offset patterns use historic samples in an uniform sequence, while Γ_5 is sampled with a logarithmic spacing, sharing the same number of coefficients as Γ_1 but spanning the same historic depth as the Γ_3. The underlying idea with the logarithmic set is that it should be able to balance the importance of the data given how old the data is. Hence, it is biased to sample more recent information in relation to older samples, which might be interpreted as mimicking the vanishing importance of older information in the brain of the driver.

In the following section, the performance of each model will be assessed with respect to two aspects: i) the MSE error, representing how accurate predictions are with respect to the ground-truth path; ii) the efficiency of lane departure predictions.

The MSE of the different models can be seen in Tab. II, from where it can be concluded that a historic depth of $0.5 - 1$ s is sufficient to provide increased performance. Furthermore, it can be observed that the logarithmic model is the best choice for most prediction horizons. It is worth noting that a low MSE can be the result of good predictions in normal driving conditions even if predictions might diverge in critical situations, i.e., in cases of lane departures that happen more seldomly.

From an application point of view, the prediction efficiency relies on the combination of MSE error and the timing aspect mentioned before. The mean triggering time for the models can be seen in Tab. II for $\tau = 0$, where many models deviate from the designed target H. Hence, models have to be calibrated before a fair comparison in TP rate and FP rate performance can be derived.

From Fig. 2 it can be seen that all calibrated MLR models, except for Γ_4 at $H = 0.5$ s, outperformed the calibrated CV model in terms of TP rates. The Γ_1 model, using only $2 \times 3 \times 13$ coefficients, works well for 0.5 s prediction horizon but its relative performance drops for longer prediction horizons. The Γ_4 model suffers from overfitting for a prediction horizon of 0.5 s, probably due to the high number of coefficients ($2 \times 80 \times 13$). The logarithmic sampled model Γ_5, that shares the same memory depth as the sequentially sampled model Γ_3, yields similar TP rate performances. For the prediction horizon 1.75 s, the MLR model is approximately 18% better than the CV.

The difference in performance is more significant when analysing the FP rate, see Fig. 3. The MLR models, except for the Γ_1, outperform the kinematic models for all prediction horizons, where improvements of up to 34% can be seen for 1.75 s prediction horizon. The Γ_1 model is under-performing

TABLE II

THE MEAN TRIGGERING TIME ($\tau = 0$) AND MSE EVALUATED USING THE TEST SET $\mathcal{C}^t$.

	0.5		**0.75**		**1.0**		**1.25**		**1.5**		**1.75**	
	$\bar{t}_{0.5}$	MSE	$\bar{t}_{0.75}$	MSE	$\bar{t}_{1.0}$	MSE	$\bar{t}_{1.25}$	MSE	$\bar{t}_{1.5}$	MSE	$\bar{t}_{1.75}$	MSE
Γ_1	0.45	0.002	0.70	0.003	0.97	0.006	1.20	0.010	1.44	0.014	1.61	0.021
Γ_2	0.41	**0.001**	0.71	**0.002**	0.91	**0.004**	1.26	**0.007**	1.43	**0.011**	1.1.65	**0.017**
Γ_3	0.39	**0.001**	0.64	**0.002**	0.92	**0.004**	1.13	**0.007**	1.49	0.012	1.55	**0.017**
Γ_4	0.44	0.002	0.72	0.003	0.98	0.005	1.13	**0.007**	1.48	0.012	1.63	0.018
Γ_5	0.43	**0.001**	0.69	**0.002**	0.94	**0.004**	1.21	**0.007**	1.40	**0.011**	1.60	**0.017**
CV	0.45	0.002	0.74	0.005	1.03	0.010	1.31	0.017	1.62	0.029	2.28	0.051

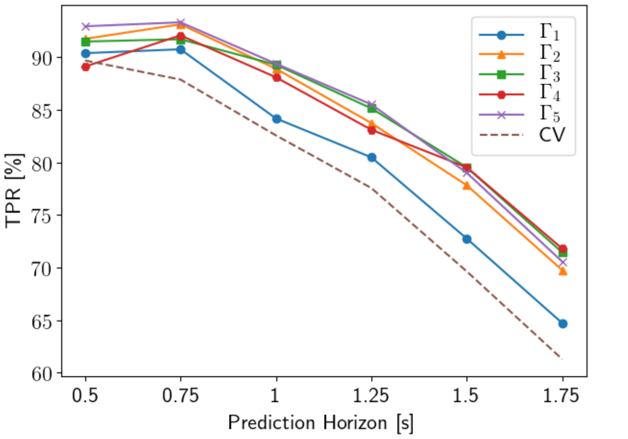

Fig. 2. TP performance for different prediction horizons and offset patterns.

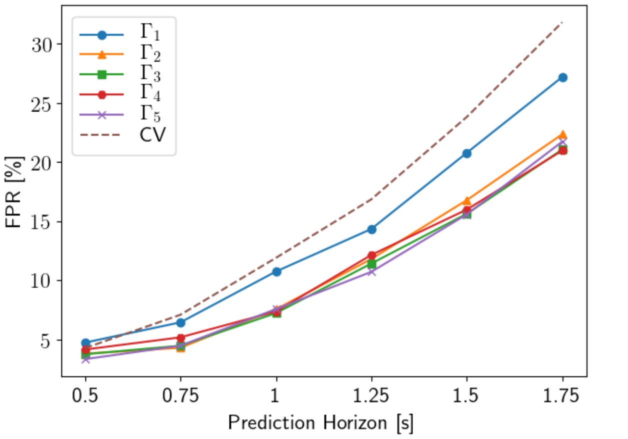

Fig. 3. FP performance for different prediction horizons and offset patterns.

with respect to the other MLR models and is even worse than the reference model at $H = 0.5$, which indicates that a memory depth shorter than 0.5 s is undesirable. For the models $\Gamma_2 - \Gamma_5$ there is no significant difference in performance. Interestingly, the logarithmic model Γ_5 is performing as good as the remaining others, which makes it a preferable choice since it has the lowest number of coefficients among the models with higher performance. Therefore, the logarithmic model Γ_5 will be used for the rest of the discussion.

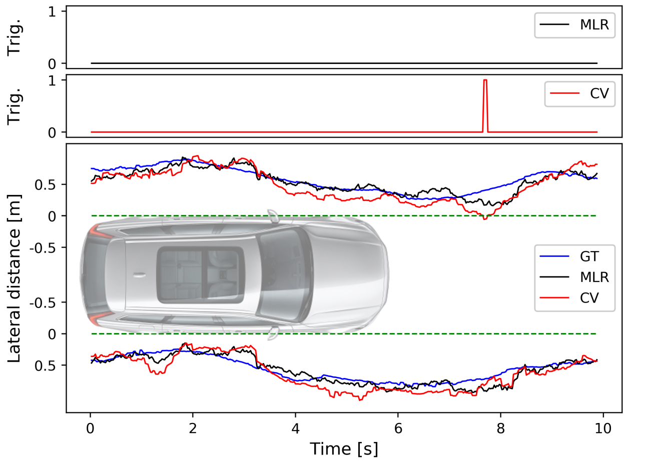

Fig. 4. An example of regular driving with a prediction horizon $H = 1.5$ s. A lane departure is determined when the perimeter of the vehicle, denoted by the green dashed line, exceeds the predicted lateral distance, illustrated by the red and black lines. The blue line indicates the ground-truth (GT). The CV based approach is falsely triggering interventions on the left hand side at $t = 7.8s$.

In order to highlight the strengths and weaknesses of the models, two representative cases are presented in Fig. 4 and Fig. 5 and will be further discussed in the sequel. In both figures: i) the lower panel presents the predicted distances (solid lines) to the lane markers relative the sides of the vehicle (represented by the horizontal dashed-lines in green); ii) the upper two panels show the triggering behavior of a decision-making algorithm (as detailed in Section IV-C), and based on the different models described before.

A snippet from the non-event set $\mathcal{B}$ is depicted in Fig. 4, exhibiting an example of a challenging driver behavior, that does not however lead to a lane departure. One can see that the driver has a wiggling driving style and is moving from side to side within the lane. In this case, it is evident that the kinematic model fails to capture the real risk of lane departure, triggering a false intervention at $t = 7.8$ s, see the upper panel of Fig. 4. On the other hand, the MLR model succeeds to identify such a situation as non-threatening, therefore not triggering any intervention as one would have expected. These results support the previous argumentation stating that the kinematic prediction models are prone to false triggerings, which can be avoided in many situations by using the MLR model.

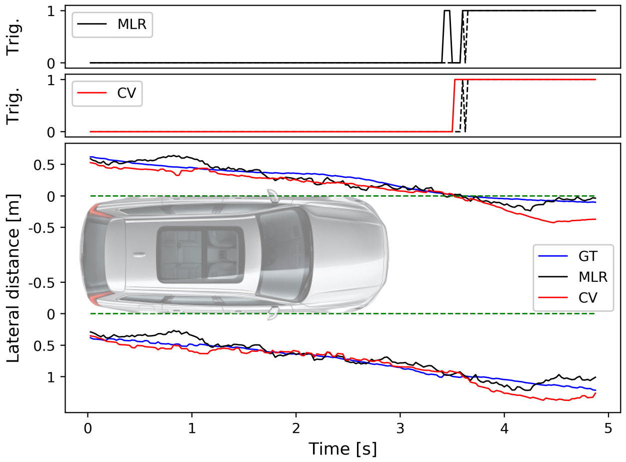

Fig. 5. An example of a vehicle departing from the lane on the left side with prediction horizon set to $H = 1.5$ s. The dashed black line in the two upper panels represents the triggering target (based on ground-truth data), while the solid line is the triggering based on the MLR prediction model. When compared with the target triggering signal, both the kinematic CV model and the MLR model, based on the offset pattern Γ_5, trigger earlier interventions. The oscillations in the predictions are due to that the prediction models are uncertain for the relatively long prediction horizon.

An example of a lane departure event is shown in Fig. 5. Both the MLR and the CV models trigger close to the designed prediction horizon, but the MLR performs better in terms of smaller deviations with respect to the ground-truth path. It can be noticed that the triggering is oscillating after the first triggering. This is not an issue in practise since an automated steering maneuver is activated on the first triggering instance, which lasts for several seconds and affects the vehicle's future behavior.

VI. Conclusion and Future Work

A *direct* h-step multiple linear regression prediction model, able to predict the lateral distances to the lane markers, is proposed and validated using real world data. The prediction model is used to detect unintended lane-departures and is shown to outperform a traditionally used kinematic model.

Future work could consider a thorough analysis of the effects of sparse sampling patterns such as linear and logarithmic sampling. It is also of interest to compare the performance of the linear prediction model presented in this work with a non-linear model, e.g., a neural network, in order to identify potential gains in terms of performance.

References

[1] European Commission, *Traffic Safety Basic Facts on Single Vehicle Accidents, European Commission, Directorate General for Transport*, 2018.

[2] J. Dahl, G. Rodrigues de Campos, C. Olsson, and J. Fredriksson, "Collision avoidance: A literature review on threat-assessment techniques," *IEEE Trans. on Intelligent Vehicles*, vol. 4, no. 1, pp. 101–113, 2019.

[3] D. Yi, J. Su, L. Hu, C. Liu, M. A. Quddus, M. Dianati, and W.-H. Chen, "Implicit personalization in driving assistance: State-of-the-art and open issues," *IEEE Trans. on Intelligent Vehicles*, 2019.

[4] A. A. Albousefi, H. Ying, D. Filev, F. Syed, K. O. Prakah-Asante, F. Tseng, and H.-H. Yang, "A support vector machine approach to unintentional vehicle lane departure prediction," *IEEE Intelligent Vehicles Symposium*, 2014.

[5] J. M. Ambarak, H. Ying, F. Syed, and D. Filev, "A neural network for predicting unintentional lane departures," *IEEE International Conference on Industrial Technology*, 2017.

[6] J. Dahl, R. Jonsson, A. Kollmats, G. R. de Campos, and J. Fredriksson, "Automotive safety: a neural network approach for lane departure detection using real world driving data," *IEEE Intelligent Transportation Systems Conference*, 2019.

[7] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, no. 1, p. 1, 2014.

[8] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," *11th International Conference on Information Fusion*, pp. 1–6, 2008.

[9] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," *2013 IEEE/RSJ international conference on intelligent robots and systems*, pp. 4363–4369, 2013.

[10] W. Wang, D. Zhao, W. Han, and J. Xi, "A learning-based approach for lane departure warning systems with a personalized driver model," *IEEE Trans. on Vehicular Technology*, vol. 67, no. 10, pp. 9145–9157, 2018.

[11] K. Min, D. Kim, J. Park, and K. Huh, "RNN-based path prediction of obstacle vehicles with deep ensemble," *IEEE Trans. on Vehicular Technology*, vol. 68, no. 10, pp. 10 252–10 256, 2019.

[12] V. Ilić, D. Kukolj, M. Marijan, and N. Teslić, "Predicting positions and velocities of surrounding vehicles using deep neural networks," *IEEE Zooming Innovation in Consumer Technologies Conference*, pp. 126–129, 2019.

[13] H. Lütkepohl, *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

[14] L. Ljung, "System identification: theory for the user," *Prentice Hall PTR*, 1999.

[15] H. Cheng, P.-N. Tan, J. Gao, and J. Scripps, "Multistep-ahead time series prediction," *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 765–774, 2006.

[16] S. B. Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition," *Expert systems with applications*, vol. 39, no. 8, pp. 7067–7083, 2012.

[17] D. Pollock, "Stochastic processes of limited frequency and the effects of oversampling," *Econometrics and Statistics*, vol. 7, pp. 18 – 29, 2018.