2021 IEEE Intelligent Transportation Systems Conference (ITSC)
Indianapolis, USA. September 19-21, 2021

V2X Database Driven Traffic Speed Prediction

Daniel Adelberger[1], Junpeng Deng[1] and Luigi del Re[1]

Abstract— **Knowledge of the upcoming traffic velocity along a route can help in many respects, among them optimizing energy management for hybrid vehicles, which, for instance, could reduce instantaneous battery usage if a traffic jam is upcoming in the next future. While such kind of knowledge can hardly be precise on a single-vehicle level, we show in this paper that a prediction method which combines present and past Vehicle-to-Everything (V2X) information can strongly improve the energy efficiency. Our approach is first compared with other prevailing prediction methods and its advantages in terms of stability and accuracy are shown. Then the prediction results are applied in a hybrid powertrain control example, in which its potential in fuel savings are illustrated.**

I. Introduction

High-quality forecasts of vehicle velocity profiles can contribute to energy saving significantly, both at vehicle control level, e.g. eco-driving [1] or powertrain control level [2]. There have been lot of studies over the past decades, still velocity prediction is a challenging task due to the many environmental influential factors such as traffic, weather, road condition, driver's preference, etc.

Traffic prediction is usually accomplished by using a model, with either parametric methods or non-parametric methods [3]. In the former group, *a priori* assumption about the structure of the model is made, and the analytical form of the prediction function is determined. Model parameters can be then learned from the collected data. Parametric approaches are widely used to model the velocity of a vehicle or traffic on highways [4], [5]. Since these approaches are based on assumptions for specific scenarios, they may only function well in those scenarios. The second kind of approaches do not fix the analytical function beforehand, but learn from data instead, for example using Gaussian Mixture Regression [3] or Neural Networks [6], [7]. These are more useful in modeling complicated systems of which underlying physics are not well understood.

Although the model-based methods prevail, they have some limitations. First, a large number of data is the prerequisite for building a good model, while data is not widely available everywhere. Second, models always need some assumptions or simplifications. Combined with a limited amount of data this can lead to unrealistic results. Also, the traffic behaviors are largely dependent on local regulations/habits which are hardly covered by generic rules.

In terms of horizon, traffic prediction can be conducted on long-distance and short-distance scales. For long-term applications (next several minutes or next several kilometers), traffic states (traffic flow and traffic density) are predicted, and typically used in traffic operation and planning. Relevant researches include ramp metering at highway or advanced traffic signal control on urban roads. Short-term prediction is more for estimating the trajectory of single vehicles. For example, [8] employed Markov chains and neural networks for a short horizon prediction. Since an individual vehicle's action is strongly constrained by the nearby traffic dynamics, the prediction of driver's behaviors may be sensible for the next few seconds, but it is challenging to predict the next 10-15 seconds [3].

While a majority of the research focuses on short-term prediction, long-term prediction can also be essential in overall energy savings, especially for hybrid electric vehicles (HEVs) to plan their state of charge (SOC) profile [9]. Yet this has attracted much less attention. This is partly due to the limited range of vehicle's sensors – the traffic situation far away cannot be observed directly. Fortunately, the development of V2X communication [10] enables the timely acquisition of distant traffic information. Still, most studies utilized only short-term V2X information to achieve automotive energy savings (see e.g. [11], [12], [13]). Only few works used V2X information in long-term velocity prediction, for instance [14] developed a vehicle routing strategy based on the information of neighboring vehicles to avoid congestion. Besides, most of the works assumed ideal V2X information, which may not be representative for all real-world situations. Also, in case of a long drive, the traffic condition at far away locations may have already changed when the vehicle really arrives there. Using "static" V2X information obtained in the beginning of a trip can reduce the possible quality of results, which is tackled by our approach.

To fill the aforementioned gaps, this paper proposes a long-term velocity prediction method utilizing real-world V2X data. Instead of building a model, we use a different approach – database search. We develop a method that utilizes similar situation in the past to receive more realistic velocity profiles over a given route, both in the spatial as well as in the temporal future. Besides, kernel-based models, Bayesian networks and Auto-regressive models are also implemented to give a comparison for a deeper analysis. Moreover, we have applied our prediction results in hybrid powertrain energy management, and the fuel saving potential is illustrated. Here "long-term" refers to the length of a whole trip, in our case $14\,\mathrm{km}$ in the spatial dimension and up to $20\,\mathrm{min}$ into the temporal future. In principle the approach can be also applied to any longer trips.

The remainder of the paper is structured as follows.

[1]Daniel Adelberger, Junpeng Deng and Luigi del Re are with the Institute for Design and Control of Mechatronical Systems, Johannes Kepler University, 4040 Linz, Austria {daniel.adelberger, junpeng.deng, luigi.delre}@jku.at

2021 IEEE International Intelligent Transportation Systems Conference (ITSC) | 978-1-7281-9142-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ITSC48978.2021.9564696

Section II introduces the route we examined and gives an overview of typical velocities along the track during a typical day to visualize patterns. In Section III multiple methods for the prediction of velocity profiles are developed, based on different principles, utilizing a prerecorded database. The respective performance is validated in Section IV, followed by a practical example in Section V where the prediction methods are applied to optimize the SOC-profile of a HEV. Section VI concludes all outcomes and discusses the developed prediction approach.

II. Analysis of Traffic Behavior

For the underlying work, we focus on a city-highway route in Linz, Austria which has a total length of 14 km, displayed in Fig. 1. We use HERE Developer [15] as our V2X provider. The HERE Developer Application Programming Interface (API) returns an estimated velocity profile along the whole route on request. This was done from 28.08.2020 – 12.11.2020 with a request frequency of 2 min resulting in a spatial and temporal database of saved velocities, as visualized for two selected days in Fig. 2.

Fig. 1. JKU Linz – Franzosenhausweg Linz [16] where the data was recorded (◆ = start, ■ = end)

The data was verified by doing test drives and requesting V2X data in parallel. One of those drives with respective V2X data (matched afterwards) is exemplarily shown in Fig. 3. The real-world measurements were taken using a BMW F31 equipped with a D-GPS sensor. The profile has of course no perfect match as the V2X information only reflects the mean velocity along the track, where for example also trucks are included which usually drive with lower velocities than other traffic participants. Still, the dynamics are quite similar from a macroscopic point of view. Therefore, all of our predictions will focus not on the exact prediction of the velocity the ego vehicle might have, but on the prediction of the V2X-request velocities in the

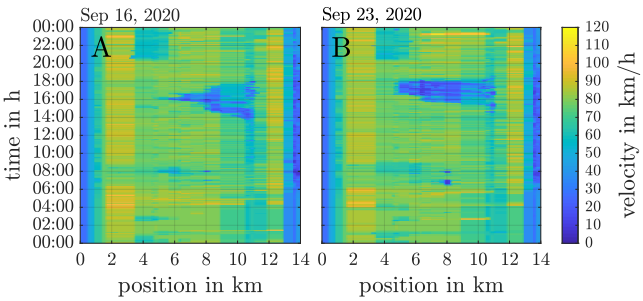

Fig. 2. Mean velocity along the route during two days

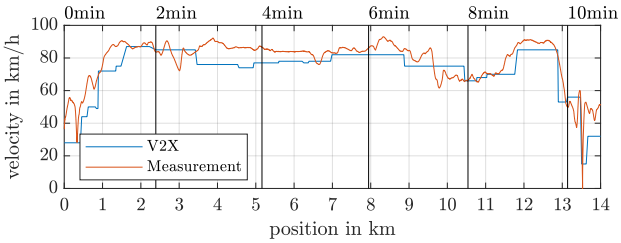

Fig. 3. Comparison of measured (driven) velocity and requested V2X information. The request time is denoted on top of the plot.

spatial and temporal future. Fig. 2 displays two days in the recorded database with a temporal difference of one week. The segments in the beginning and end of the route on both days show lower velocities as they reflect driving in city traffic. For each of the days, there is a small decrease in velocity (a jam) at around 07:00 as well as around 08:00 at a position of 8 km. This most probably indicates a time where people tend to drive to work. The same happens in a larger scale from 15:00 - 18:00 where people usually drive home from work. The affected region ranges from 8 km - 11 km in this case. Clearly, some similar patterns can be observed in general traffic situations which are utilized in the following.

III. Velocity Prediction

To formalize our concept, a velocity at a certain step in time k and a certain position in space n is denoted as $v\,(k \mid n)$ while a whole segment ranging from k_{start} to k_{end} and n_{start} to n_{end} is displayed as $v\left(\begin{smallmatrix} k_{\text{end}} \\ k_{\text{start}} \end{smallmatrix} \middle| \begin{smallmatrix} n_{\text{end}} \\ n_{\text{start}} \end{smallmatrix}\right)$.
$\hat{v}$ indicates a prediction and the subscript k in $v_{\backslash k}$ denotes a velocity relative to the route at the temporal step k ($v_{\backslash 0}\,(k \mid n) = v\,(k \mid n) - v\,(0 \mid n)$). The spatial dimension is discretized with a resolution of $\Delta d = 100$ m and the temporal with a resolution of $\Delta t = 2$ min which also reflects our request frequency from the HERE-Server. The time denotes to $t(k) = \Delta t \cdot k$ with $k \in \mathbb{N}$ (where $k = 0$ represents the present time) and the position along the route is defined as $d(n) = \Delta d \cdot n$ with $n \in \mathbb{N}_0$, where the number of samples is restricted by the length of the route to $N = 140$.

In the following, multiple prediction methods are described, where SSP_W (referring to a Situation Similarity based Prediction with Window based selection – the respective components will be discussed later) represents a novel

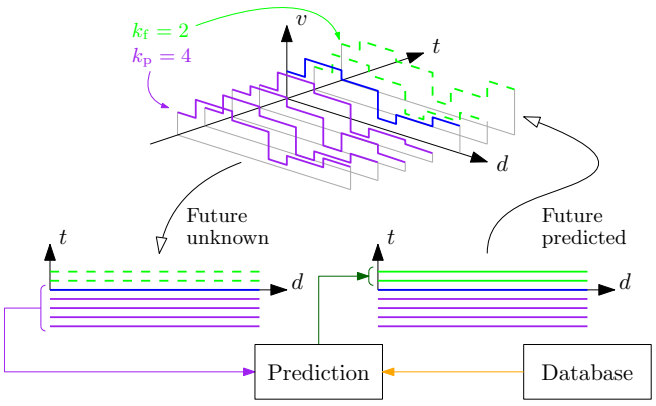

Fig. 4. Schematic concept of the prediction with $k_\mathrm{f} = 2$ prediction steps and $k_\mathrm{p} = 4$ past spatial velocity profiles utilized for the database comparison

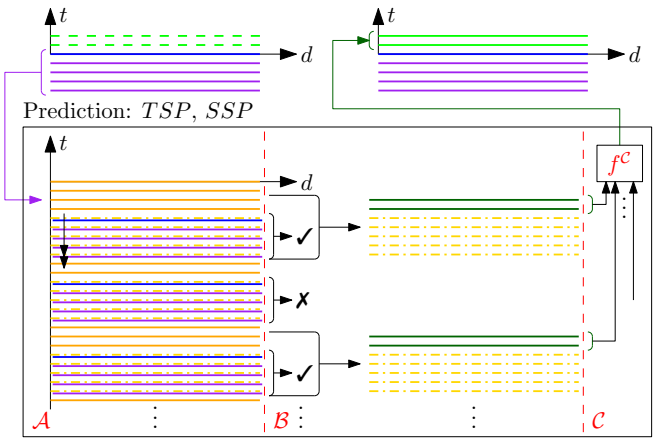

Fig. 5. Prediction concept for the TSP and SSP method, which differ in terms of the ranking process that happens in segment $\mathcal{B}$

approach that directly predicts the future based on similar past situations. For performance comparison, an additional set of typical methods is tested, including Auto-regressive (AR), Kernel-based (KM) as well as Bayesian Network (BN) based prediction models which are (for the validation) also combined with the selection method used for SSP_W.

A. Prediction methods

Based on the visible patterns in Fig. 2, various variants for velocity prediction can be developed. The framework for prediction is displayed in Fig. 4, the respective contents of the prediction block within the figure is described in the following.

ZOH: This method serves as a benchmark, the initial HERE-request is kept over time (Zero Order Hold).Therefore it represents "static" V2X information.

$$\hat{v}_{\backslash 0}\left(k \,|\, n\right) = v_{\backslash 0}\left(0 \,|\, n\right) \tag{1}$$

TSP: Makes use of velocities at similar days and times (Temporal Similarity based Prediction). The database (segment $\mathcal{A}$) is searched for entries with the same weekday and time of the day (segment $\mathcal{B}$, comparison with $\mathrm{equaltime}()$) and the mean of the respective future of all selected entries is calculated (function $f^{\mathcal{C}}$ that is implemented in Eq. (2) in segment $\mathcal{C}$) which serves then as the predicted value. The whole process is sketched in Fig. 5.

$$\hat{v}_{\backslash 0}\left(k \,|\, n\right) = \frac{1}{\#K^*} \sum_{k' \in K^*} v_{\backslash k'}\left(k' + k \,|\, n\right) \tag{2}$$

where the selected set in segment $\mathcal{B}$ is obtained by

$$K^* = \{k' \,|\, \mathrm{equaltime}(t(k'), t(0)) = \mathrm{true} \wedge k' \in K\} \tag{3}$$

with a cardinality $\#K^*$ directly depending on the size of the available training dataset K (database).

SSP: The database is searched for situations similar (similarity is determined by calculating the inverse FIT-value) to the current one and the mean of their respective future is utilized to predict the future in the present situation (for the complete route). The whole process is sketched in Fig. 5. The difference to the TSP method is the selection of past elements, for the SSP method K^* is calculated according to Eq. (4) and not Eq. (3)

$$K^* = \cup_{i=1}^{i_\mathrm{t}} k'_i \tag{4}$$

$$k'_1 = \arg\min_{k'} \mathrm{iFIT}(k'),\ k' \in K \tag{5}$$

$$k'_i = \arg\min_{k'} \mathrm{iFIT}(k'),\ k' \in K \setminus \{k'_1, \ldots, k'_{i-1}\} \tag{6}$$

with $\mathrm{iFIT}(k')$ as defined in Eq. (7) where $n_L = 0$ and $n_R = N$ while i_t represents the number of training datasets. $\overline{v}$ denotes the mean velocity of a segment (which is a scalar value).

$$\mathrm{iFIT}(k') = \frac{\sqrt{\sum_{k=-k_\mathrm{p}}^{0} \sum_{n=n_L}^{n_R} \left|v\left(k \,|\, n\right) - v\left(k' + k \,|\, n\right)\right|^2}}{\sqrt{\sum_{k=-k_\mathrm{p}}^{0} \sum_{n=n_L}^{n_R} \left|v\left(k \,|\, n\right) - \overline{v}\left(\begin{smallmatrix}0\\-k_\mathrm{p}\end{smallmatrix} \middle| \begin{smallmatrix}n_R\\n_L\end{smallmatrix}\right)\right|^2}} \tag{7}$$

SSP_W: Relies on the same principles as SSP, but in this case only a small spatial window is compared that yields the future for the spatial position in the center of the window with set width w (discrete steps). The same equations as for the standard SSP method hold, with the difference that n_L and n_R now represent the left and right margin of the window.

$$\begin{aligned} n_L &= \max\{0,\, n - w/2\} \\ n_R &= \min\{N,\, n + w/2\} \end{aligned} \tag{8}$$

This window is then shifted along the whole route ($n = 0, \ldots, N$). The method is sketched for one step n in Fig. 6 where in the latter part the segment $\mathcal{C}_W$ is the relevant one for the method. If not further specified, $w = 10$ is selected for all following studies.

B. Comparative methods for prediction

For comparison, three other models are presented too, which are either trained using random samples from the whole dataset (displayed without subscript) or with training

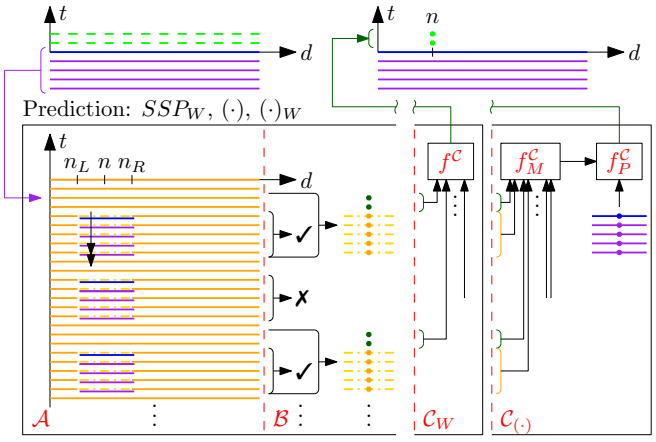

Fig. 6. Prediction concept for the SSP_W method (where the last segment is defined by $\mathcal{C}_W$) and all comparative methods (AR, AR_W, KM, KM_W, BN, BN_W) where the last segment is defined by $\mathcal{C}_{(\cdot)}$

samples selected similarly to SSP_W (which are also marked with a subscript W):

AR/AR_W: For the Auto-regressive prediction of the velocity, for each prediction step k_i a separate $\hat{\theta}_i$ is trained using the relation

$$\hat{\theta}_k = \arg\min_{\theta_k} \sum_{k' \in K^*} \left(v_{\backslash k'} \left(k' + k \,|\, n \right) - \varphi_{k'}^{\top} \cdot \theta_k \right)^2 \tag{9}$$

$$\varphi_{k'}^{\top} = \left[v_{\backslash k'} \left(k' \,|\, n \right) \; \ldots \; v_{\backslash k'} \left(k' - k_p \,|\, n \right) \right] \tag{10}$$

which is represented by the block $f_M^{\mathcal{C}}$ in Fig. 6. K^* is determined according to Eq. (4) and Eq. (8) with $i_\mathrm{t} = 100$ for the AR_W method and randomly selected from the set K with a cardinality of $i_\mathrm{t} = 1000$ for the AR method (both represented in segment $\mathcal{B}$ in Fig. 6). The prediction is then obtained using the current past (block $f_P^{\mathcal{C}}$ in Fig. 6)

$$\hat{v}_{\backslash 0} \left(k \,|\, n \right) = \varphi_0^{\top} \cdot \hat{\theta}_k \tag{11}$$

KM/KM_W: The Kernel method obtains the prediction of the velocity based on a Gaussian Kernel with optimized kernel parameter σ and regularization parameter α, chosen from a pre-defined set

$$\sigma \in \{0.5, 1, \ldots, 5\}, \; \alpha \in \{10^{-6}, 10^{-5}, \ldots, 1\}. \tag{12}$$

The best combination (trained on the dataset specified by K^*) is determined as followed. K^* is selected as for the AR/AR_W method, with the same differentiation for KM and KM_W.

$$(\alpha_n^*, \sigma_n^*) = \arg\min_{\alpha,\sigma} \sum_{k' \in K^*} \left(v_{\backslash k'} \left(\begin{matrix} k' + k_f \\ k' \end{matrix} \middle| \begin{matrix} n \\ n \end{matrix} \right) - \mathcal{K}_{K^*}^{\alpha,\sigma}(\Phi_{k'}^{\top}) \right)^2 \tag{13}$$

with

$$\Phi_{k'}^{\top} = v_{\backslash k'} \left(\begin{matrix} k' \\ k' - k_p \end{matrix} \middle| \begin{matrix} n_R \\ n_L \end{matrix} \right) \tag{14}$$

reflecting the modeling part (represented by the block $f_M^{\mathcal{C}}$ in Fig. 6) and the best combination is then used for the prediction utilizing the current past (block $f_P^{\mathcal{C}}$ in Fig. 6)

$$\hat{v}_{\backslash 0} \left(\begin{matrix} k_f \\ 0 \end{matrix} \middle| \begin{matrix} n \\ n \end{matrix} \right) = \mathcal{K}_{K^*}^{\alpha_n^*, \sigma_n^*}(\Phi_0^{\top}) \tag{15}$$

BN/BN_W: Here, the prediction of the velocity is based on a (selectively) trained Bayesian Network which is again done in two ways (different determination of K^*) resulting in BN and BN_W. The network $\mathcal{N}_{K^*}^n$ is trained to predict the respective $v_{\backslash k'} \left(\begin{matrix} k' + k_f \\ k' \end{matrix} \middle| \begin{matrix} n \\ n \end{matrix} \right)$ with $\varphi_{k'}^{\top}$ for all $k' \in K^*$ (represented by the block $f_M^{\mathcal{C}}$ in Fig. 6). Then, the trained network is directly used for prediction (block $f_P^{\mathcal{C}}$ in Fig. 6)

$$\hat{v}_{\backslash 0} \left(\begin{matrix} k_f \\ 0 \end{matrix} \middle| \begin{matrix} n \\ n \end{matrix} \right) = \mathcal{N}_{K^*}^n(\varphi_0^{\top}) \tag{16}$$

IV. Validation

In order to evaluate the performance of the introduced prediction methods, the available database is split into a training dataset K and a validation dataset K^v with a cardinality of $\#K^v = 100$ traffic situations. The training dataset represents the database that the prediction methods can access. To compare the performance of the various methods, reasonable measures are required. We decided to evaluate the Root Mean Squared Error (RMSE) as well as the Median Maximum Error (MME), which represents the median of the maximum absolute prediction errors for all validations.

$$\mathrm{RMSE}(k) = \sqrt{\frac{1}{\#K^v \cdot N} \sum_{k^v \in K^v} \sum_{n=0}^{N} (\Delta v)^2} \tag{17}$$

$$\mathrm{MME}(k) = \underset{k^v \in K^v}{\mathrm{median}} \left(\max_n (\Delta v) \right) \tag{18}$$

$$\Delta v = \left| v_{\backslash k^v} \left(k^v + k \,|\, n \right) - \hat{v}_{\backslash k^v} \left(k^v + k \,|\, n \right) \right| \tag{19}$$

For all investigations, the variable determining how many past requests are taken into account is set to $k_\mathrm{p} = 4$ and the number of prediction samples is set to $k_\mathrm{f} = 10$. The meaning of both variables is displayed in Fig. 4. The following figures show the results of the validation in terms of RMSE and MME, both over the whole prediction horizons. In general it is to say that the MME is more robust to outliers due to the median that is calculated (compared to the mean in case of the RMSE).

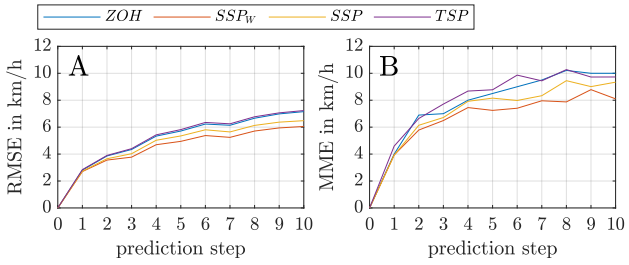

Fig. 7. Comparison of prediction methods, including a reduction of the training dataset

Fig. 7 shows the course of the prediction performance for ZOH, TSP, SSP as well as SSP_W. The ZOH method already performs very well, mostly due to the generally

low variation of the velocity over time, as observable in Fig. 2. The TSP method performs worse, mostly due to the reason that macroscopic patterns (see Fig. 2) do not tend to appear at the exact same microscopic time. The SSP method copes with this problem by predicting the velocity based on similar situations, leading to a better performance than ZOH. Still, a situation might be similar at one part of the route but completely different at a latter part which would lead to a bias in this case.This problem motivates the SSP_W method that only searches for similar situation in the same spatial region. Apparently this assumption holds, as the SSP_W method performs superior compared to the other methods.

A. Comparison with alternative prediction methods

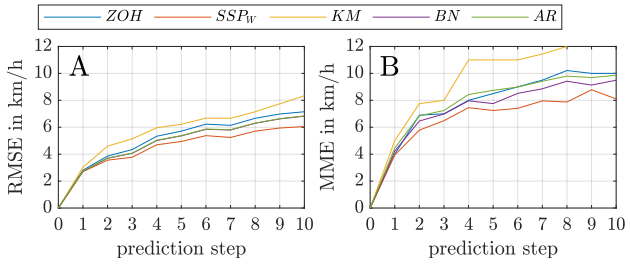

Fig. 8. Comparative prediction methods in relation to the SSP_W method as well as the ZOH approach

Now, the developed SSP_W method is compared to typical prediction methods, displayed in Fig. 8. While all methods perform similar or better compared to ZOH (except KM which seems to be not suitable for this specific type of prediction), SSP_W still shows a superior performance. This is most likely due to the fact that it is based specifically on situations that show similar dynamics as the one that needs to be predicted, leading to the next question: how would the methods perform if they are trained with the same data that is utilized by the SSP_W method? We analyze this in the following.

B. Selective training of alternative prediction methods

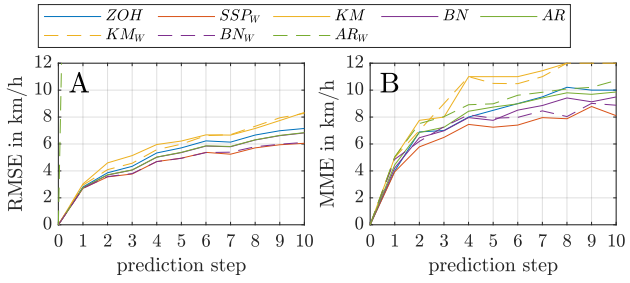

Fig. 9. Influence of selective training on comparative prediction methods

As shown in Fig. 9, a selection of training data leads to a significant performance increase for BN_W compared to BN which appears intuitive and a loss in performance for AR_W. Here, apparently the training data was not rich enough such that at least some predictions got unstable, this is clearly visible when looking at the RMSE. Still, the MME indicates that this was not too often the case as it only got a little worse. KM_W shows a slightly better performance for longer prediction horizons, but performs a little worse for shorter prediction horizons. Again, the performance is worse than that of the simple ZOH method. A comparison of all methods in relation to ZOH is shown in Table I, performances superior to ZOH are displayed in bold and the best respective performance for each criterion is underlined. The proposed SSP_W method turns out to be the best in three of four cases.

TABLE I
COMPARISON OF ALL PREDICTION METHODS

	Error difference compared to ZOH in %			
Method	RMSE(5)	RMSE(10)	MME(5)	MME(10)
TSP	1.89	1.11	3.25	**-2.77**
SSP	**-6.36**	**-8.77**	**-4.06**	**-9.90**
SSP_W	**-13.40**	**-14.91**	**-14.76**	**-12.15**
KM	8.92	10.84	29.41	29.30
KM_W	4.85	13.75	23.53	20.00
BN	**-6.13**	**-5.22**	**-8.74**	**-8.67**
BN_W	**-13.51**	**-14.35**	**-6.90**	**-9.82**
AR	**-5.78**	**-4.98**	2.83	**-3.12**
ARP_W	5705.45	8318.51	5.70	2.09

The next section shows very briefly how our developed methods can be applied to a practical problem in the field of HEV to increase fuel saving by supporting the energy management strategy.

V. APPLICATION IN ENERGY MANAGEMENT FOR HYBRID VEHICLES

We applied the proposed prediction method in a hybrid powertrain energy management example to show its potential in terms of fuel saving. The detailed vehicle model and optimization problem can be seen in [17]. Since the predicted profiles are piece-wise constant, which is unrealistic in powertrain control, they are smoothened using Fourier transformation to eliminate the high frequency parts before being sent to the cloud, as shown in [17]. The control scheme is displayed in Fig. 10, consisting of a remote and an on-board part.

The general workflow is:

1) Before departure, the on-board controller predicts a velocity profile $\hat{v}_{\mathrm{pre}}$ and sends it to the remote cloud.
2) An optimal long-term powertrain reference $x_{\mathrm{r}}^*(d)$ is calculated by the cloud and returned to the vehicle.
3) The vehicle departs. The on-board receding horizon controller tries to follow the reference $x_{\mathrm{r}}^*(d)$, calculating the optimal powertrain action from actual power demand P_{act} and gives actions u_k to the powertrain.

Apparently, the more accurate $\hat{v}_{\mathrm{pre}}$, the better the resulting powertrain control (in this case fuel consumption). We used three different types of $\hat{v}_{\mathrm{pre}}$ for comparison:

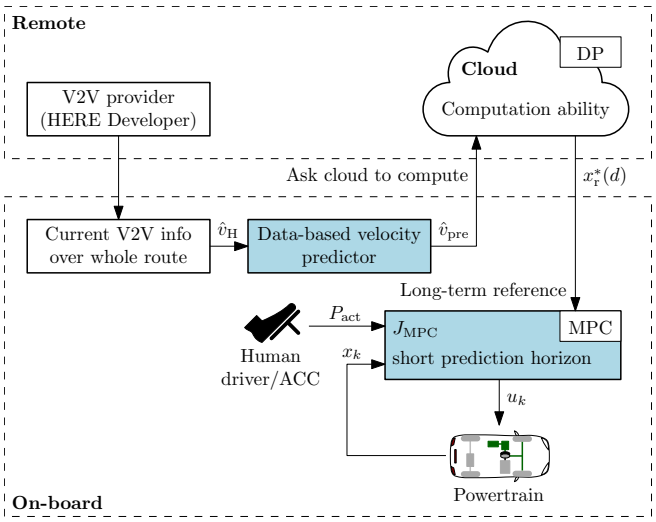

Fig. 10. Structure of the control scheme

1) Past real velocity profiles, which can to some extent reflect the traffic situation of the route. This is referred to as the group "without V2X information".
2) Our predictions. We request a live velocity profile $\hat{v}_{\mathrm{H}}$ from HERE, which goes through the predictor (i.e. with 4 prediction variants ZOH, SSP, SPP_W and TSP) and post-processing (smoothing) to become $\hat{v}_{\mathrm{pre}}$. This is referred to as the group "with V2X information".
3) Perfect prediction (the whole velocity profile is assumed to be known beforehand) as a baseline.

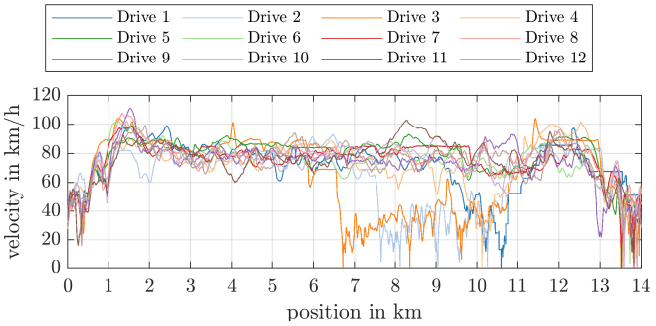

Fig. 11. Measurements on different days and times

Fig. 11 shows 12 drive cycles recorded at different times on the route from Fig. 1, representing various traffic situations. We assume drive $1-5$ are past drives, and $6-12$ are current drives that can be used for validation. Simulations were run with each method for drive $6-12$, and the additional fuel consumption, compared to the baseline solution, is plotted in Fig. 12.

The left plot of Fig. 12 shows the statistics of the additional fuel consumption for all tested cycles with and without V2X information. The median of additional fuel consumption is 28.9% for the non-V2X group and 22.7% for the V2X group. In the extreme case, there can be 45.1% additional fuel for the non-V2X group and 35.6% for the V2X group. The non-V2X group has a more scattered distribution (longer length of the box plot) while the V2X group has a more concentrated distribution. This illustrates that using V2X information can lead to a more stable performance. In other words, in general the V2X approach is much more likely to yield better performance, even in extreme situations. This illustrates that utilizing the proposed prediction methods with live information can in general bring a benefit in terms of fuel consumption.

The right plot of Fig. 12 compares the statistics of four V2X variants. Using static V2X data directly ($HERE$) gives the most "unstable" results. With prediction, the performances are relatively condensed. There is little difference in the medians. It is to notice that SSP_W does not always behave the "best" like it could be assumed based on Section IV. This is due to several reasons. First, the traffic information we used only reflects the general behavior of all the vehicles, not that of the single vehicle, especially not the specific microscopic dynamics (where acceleration and deceleration happen). Second, a drive of $14\,\mathrm{km}$ might be not long enough to show significant changes in traffic, and thus not able to show the full potential of the method. Note that the real-world V2X information is not perfect (see Fig. 3), but the quality is expected to get better in the future as the technology develops. In other words, the results shown are rather conservative estimates, which are however promising.

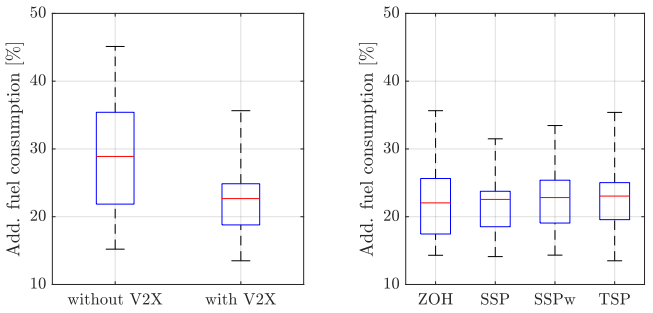

Fig. 12. Overall statistics of the additional fuel consumption of with/without V2X information (left) and comparison between prediction methods (right)

VI. Conclusion and Future Works

In the present work, a method for long-term prediction of macroscopic velocities along a given track is developed. Firstly, a route which is mainly based on city-highways is analyzed. The presence of visible velocity-patterns at specific times serves as a base for the then developed prediction methods. These are derived from scratch and the various steps during development are commented and reasoned. Additionally, three comparative methods are introduced and all approaches are validated in order to be able to discuss the respective performance. The developed prediction methods are applied on a practical example to show their potential in

terms of fuel saving in the field of hybrid powertrain energy management. As our approach was already successfully applied in the field of hybrid powertrain energy management, it is planned to expand the scope of usage to the field of autonomous safety systems. For instance, it might be possible to increase safety for Advanced Driver Assistance Systems (ADAS) by predicting and detecting road-segments with apparently untypical behavior in terms of traffic flow, enabling a planned reaction on the potential disruption.

VII. Acknowledgments

This work has been supported by the LCM – K2 Center within the framework of the Austrian COMET-K2 program.

References

[1] A. Sciarretta, G. De Nunzio, and L. L. Ojeda, "Optimal ecodriving control: Energy-efficient driving of road vehicles as an optimal control problem," *IEEE Control Systems Magazine*, vol. 35, no. 5, pp. 71–90, 2015.

[2] F. Mensing, "Optimal energy utilization in conventional, electric and hybrid vehicles and its application to eco-driving," Ph.D. dissertation, INSA de Lyon, 2013.

[3] S. Lefèvre, C. Sun, R. Bajcsy, and C. Laugier, "Comparison of parametric and non-parametric approaches for vehicle speed prediction," in *2014 American Control Conference*. IEEE, 2014, pp. 3494–3499.

[4] M. Papageorgiou, J.-M. Blosseville, and H. Hadj-Salem, "Macroscopic modelling of traffic flow on the boulevard périphérique in paris," *Transportation Research Part B: Methodological*, vol. 23, no. 1, pp. 29–47, 1989.

[5] R. Damrath and M. Rose, "Dynamische verkehrsprognosen auf der basis makroskopischer modellansätze [dynamic traffic forecasts on the basis of macroscopic model approaches]," *Forschung Straßenbau und Straßenverkehrstechnik*, no. 854, 2002.

[6] A. Fotouhi, M. Montazeri, and M. Jannatipour, "Vehicle's velocity time series prediction using neural network," *International Journal of Automotive Engineering*, vol. 1, no. 1, pp. 21–28, 2011.

[7] C. Xiang, F. Ding, W. Wang, and W. He, "Energy management of a dual-mode power-split hybrid electric vehicle based on velocity prediction and nonlinear model predictive control," *Applied energy*, vol. 189, pp. 640–653, 2017.

[8] C. Sun, X. Hu, S. J. Moura, and F. Sun, "Velocity predictors for predictive energy management in hybrid electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1197–1204, 2014.

[9] C. Sun, S. J. Moura, X. Hu, J. K. Hedrick, and F. Sun, "Dynamic traffic feedback data enabled energy management in plug-in hybrid electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1075–1086, 2014.

[10] 5G Americas, "5G Americas White Papers," https://www.5gamericas.org/en/resources/white-papers/, 2018, accessed: 2020-12-17.

[11] D. Moser, H. Waschl, R. Schmied, H. Efendic, and L. del Re, "Short term prediction of a vehicle's velocity trajectory using its," *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, vol. 8, no. 2015-01-0295, pp. 364–370, 2015.

[12] G. De Nunzio, C. C. De Wit, P. Moulin, and D. Di Domenico, "Eco-driving in urban traffic networks using traffic signals information," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 6, pp. 1307–1324, 2016.

[13] D. Baker, Z. D. Asher, and T. Bradley, "V2v communication based real-world velocity predictions for improved hev fuel economy," SAE Technical Paper, Tech. Rep., 2018.

[14] J. W. Wedel, B. Schünemann, and I. Radusch, "V2x-based traffic congestion recognition and avoidance," in *2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks*. IEEE, 2009, pp. 637–641.

[15] HERE Technologies, "HERE Developer," https://developer.here.com/, 2020, accessed: 2020-12-17.

[16] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org, 2017.

[17] J. Deng, D. Adelberger, and L. del Rel, "Hybrid powertrain control with dynamic velocity prediction based on real-world v2x information [accepted]," preprint available at https://drive.jku.at/filr/public-link/file-preview/0cce88f175e46cbd01762d43a1c7511d/23425/3338653387304714541, 2021.