Interaction-Aware Motion Prediction for Autonomous Driving: A Multiple Model Kalman Filtering Scheme

Originally published in:
IEEE Robotics and Automation Letters 6(1), https://doi.org/10.1109/lra.2020.3032079

Funding acknowledgement:
157601 - Safety and Performance for Human in the Loop Control (SNF)

Interaction-Aware Motion Prediction For Autonomous Driving: A Multiple Model Kalman Filtering Scheme

Vasileios Lefkopoulos, Marcel Menner, Alexander Domahidi, and Melanie N. Zeilinger

This work has been accepted for publication in the IEEE Robotics and Automation Letters.
Digital Object Identifier (DOI): 10.1109/LRA.2020.3032079

Please cite this work as

V. Lefkopoulos, M. Menner, A. Domahidi and M. N. Zeilinger, "Interaction-Aware Motion Prediction for Autonomous Driving: A Multiple Model Kalman Filtering Scheme," in *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 80–87, Jan. 2021.

Interaction-Aware Motion Prediction For Autonomous Driving: A Multiple Model Kalman Filtering Scheme

Vasileios Lefkopoulos*, Marcel Menner*, Alexander Domahidi, and Melanie N. Zeilinger

***Abstract*—We consider the problem of predicting the motion of vehicles in the surrounding of an autonomous car, for improved motion planning in lane-based driving scenarios without inter-vehicle communication. First, we address the problem of single-vehicle estimation by designing a filtering scheme based on an Interacting Multiple Model Kalman Filter equipped with novel intention-based models. Second, we augment the proposed scheme with an optimization-based projection that enables the generation of non-colliding predictions. We then extend the approach to the problem of simultaneously estimating multiple vehicles by using a hierarchical approach based on a priority list. The priority list is dynamically adapted in real-time according to a proposed sorting algorithm. Finally, we evaluate the proposed scheme in simulations using real-life vehicle data from the Next Generation Simulation (NGSIM) dataset.**

***Index Terms*—Motion and Path Planning, Intelligent Transportation Systems, Probability and Statistical Methods.**

I. Introduction

MOTION prediction is a necessary part of any autonomous driving application that employs predictive planning techniques [1]. It can be classified into three categories with an increasing level of abstraction [2]: physics-based, maneuver-based, and interaction-aware motion models. *Physics-based motion models* assume that the vehicle's motion depends only on physical equations of motion. They are the simplest models (e.g., constant velocity, constant acceleration) with low computational complexity and, as a result, their predictions are typically only reliable for a short horizon. *Maneuver-based motion models* assume that the vehicle's motion can be represented by a series of maneuvers executed independently of other vehicles. Thus, in contrast to the physics-based models that work with basic motion primitives, the predictions of each vehicle are more reliable. *Interaction-aware motion models* consider the reactive part of multiple vehicles, thus leading to more accurate and realistic predictions, which come at the cost of increased computational complexity.

In this work, we propose a motion prediction scheme based on an Interacting Multiple Model Kalman Filter (IMM-KF) for multiple vehicles that combines ideas from physics-based, maneuver-based, and interaction-aware approaches. The proposed scheme is capable of predicting collision-free, interaction-aware trajectories of multiple traffic participants for multiple seconds. Our predictor consists of three components:

(i) intention-based motion models that intuitively capture the behavior most drivers exhibit in typical driving scenarios, i.e., velocity tracking and distance keeping from a leading vehicle,

(ii) a projection algorithm in order to generate interaction-aware and non-colliding motion predictions, and

(iii) a time-varying priority list based on the rules of the road that allows for the decomposition of the multiple-vehicle motion prediction problem into smaller tractable single-vehicle problems.

The scheme is tuned with a small amount of data and is computationally real-time capable on embedded platforms that are typical for autonomous driving applications.

Related work using an IMM filter for motion prediction is presented, e.g., in [3]–[10]. In [3], a prediction scheme for a vehicle's position is proposed, where the filter models are constant velocity, constant acceleration, and constant jerk. In [4], a position tracking scheme using an IMM filter with a variable structure is proposed, which is equipped with a constant velocity and a constant turn model. In [5], an estimation scheme that combines an IMM filter with a Bayesian network is proposed. In [6], a simplified kinematic bicycle model is used, whose linearization is estimated to incorporate the road curvature into the filter. In [7], an object tracking scheme for urban intersections is proposed with three filter models: going forward, turning left, and turning right. Further, [8] uses linearized bicycle dynamics and decouples longitudinal and lateral estimation. To predict longitudinal motion, constant velocity and constant acceleration models are used, whereas to predict laterally, lane-converging models based on a Linear Quadratic Regulator (LQR) are utilized. In [9], linearized versions of the constant velocity rectilinear, constant acceleration rectilinear, constant angular velocity curvilinear, and constant angular acceleration curvilinear motion are used as filter models.

Compared to [3]–[10], our contributions are the introduction of (i) intention-based longitudinal models, where the intention is

Manuscript received: May, 27, 2020; Revised August, 12, 2020; Accepted September, 20, 2020. This paper was recommended for publication by Editor N. Amato upon evaluation of the Reviewers' comments. The research of M. Menner was supported by the Swiss National Science Foundation under grant no. PP00P2_157601 / 1. *(Corresponding author: Marcel Menner.)*

* Co-first authorship.

V. Lefkopoulos, M. Menner, and M. N. Zeilinger are with the Institute for Dynamic Systems and Control, ETH Zurich, Zurich 8092, Switzerland (e-mail: vlefkopo@gmail.com; mmenner@ethz.ch; mzeilinger@ethz.ch).

A. Domahidi is with embotech AG, Zurich 8005, Switzerland (e-mail: domahidi@embotech.com).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org

Digital Object Identifier 10.1109/LRA.2020.3032079

inferred online, (ii) an optimization-based projection to generate interaction-aware and non-colliding motion predictions, and (iii) a priority list, which is adapted online, to make the problem real-time capable. These three components allow us to make collision-free and interaction-aware predictions of multiple traffic participants, with improved error statistics over IMM-KF schemes that use simpler physics-based motion models. Compared to deep learning-based techniques, e.g., [11]–[17], our method is implemented recursively, with a feedback mechanism to correct for errors, provides an interpretation of its outputs including a measure of certainty of its predictions, and does not require large amounts of training data. We quantitatively compare the proposed scheme with deep learning-based methods, using the Next Generation Simulation (NGSIM) dataset. The proposed scheme exhibits beneficial computation times on hardware that is easily embeddable in typical autonomous driving hardware, while achieving comparable prediction accuracy compared to deep learning-based techniques.

Notation & Preliminaries

We denote a conjunction (logical AND) by $\bigwedge$, a union of sets by $\bigcup$, and the set of binary numbers $\{0,1\}$ by $\mathbb{B}$. $\mathcal{N}(\mu,\Sigma)$ is the Gaussian distribution with mean $\mu\in\mathbb{R}^n$ and covariance $\Sigma\in\mathbb{R}^{n\times n}$, and $\mathbb{E}[x]$ is the expected value of x. I denotes the identity matrix, $\mathbb{0}$ denotes a zero vector or matrix, and $A\succ 0$ denotes a symmetric positive definite matrix.

Kalman Filter: The KF is a state estimator for linear systems:

$$\begin{aligned} x_{k+1} &= F_k x_k + E_k + w_k\,,\\ y_k &= H_k x_k + v_k\,, \end{aligned}$$

where $k\in\mathbb{N}$ is the current time, $x_k\in\mathbb{R}^{n_x}$ is the state, E_k is the input, $y_k\in\mathbb{R}^{n_y}$ is the measurement, $w_k\sim\mathcal{N}(0,Q_k)$ is the process noise, and $v_k\sim\mathcal{N}(0,R_k)$ is the measurement noise. The initial state is $x_0\sim\mathcal{N}(\mu_{x_0},\Sigma_{x_0})$. The KF involves a prediction and an update step. The prediction step yields

$$\hat{x}_{k|k-1} = F_{k-1}\hat{x}_{k-1|k-1} + E_{k-1}\,, \tag{1a}$$
$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^\top + Q_{k-1}\,, \tag{1b}$$

where $\hat{x}_{k|k-1}$ is the predicted (a priori) state estimate and $P_{k|k-1}$ is the predicted error covariance. The update step is

$$\tilde{y}_k = y_k - H_k\hat{x}_{k|k-1}\,, \tag{2a}$$
$$S_k = H_k P_{k|k-1}H_k^\top + R_k\,, \tag{2b}$$
$$K_k = P_{k|k-1}H_k^\top {S_k}^{-1}\,, \tag{2c}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\tilde{y}_k\,, \tag{2d}$$
$$P_{k|k} = (I - K_k H_k)P_{k|k-1}\,, \tag{2e}$$

with the innovation residual $\tilde{y}_k$ and its covariance S_k, the Kalman gain K_k, the (a posteriori) state estimate $\hat{x}_{k|k}$, and the estimate covariance $P_{k|k}$, where $\hat{x}_{0|0}=\mu_{x_0}$, $P_{0|0}=\Sigma_{x_0}$ [18].

Interacting Multiple Model Kalman Filter: The IMM-KF is an estimation scheme for hybrid systems [19]. Consider the discrete Markov jump linear system:

$$\begin{aligned} x_{k+1} &= F_k^{(i)}x_k + E_k^{(i)} + w_k^{(i)}\,,\\ y_k &= H_k^{(i)}x_k + v_k^{(i)}\,, \end{aligned}$$

where (i) denotes the currently active model m_k from the model set: $\mathcal{M}=\{m^{(1)},\dots,m^{(M)}\}$. The transition from model $m^{(i)}$ to $m^{(j)}$ is described through a Markov chain, i.e., the probability of transitioning from $m^{(i)}$ to $m^{(j)}$ is

$$\Pr\big(m_{k+1}=m^{(j)}|m_k=m^{(i)}\big)=\pi_{ij}\,, \tag{3}$$

where $m_k\in\mathcal{M}$ is the model active at time k and $\pi_{ij}\in[0,1]$ is the transition probability. The basic principle of the IMM-KF is that separate filters are used (in parallel) for each $m^{(i)}$ and their estimates are then utilized to estimate the probability of each $m^{(i)}$ being active. The IMM-KF involves an interaction, a filtering, a probability update, and a combination step. In the interaction step, the individual filter estimates are mixed and used to initialize each filter:

$$c^{(i)} = \sum_{j=1}^{M}\pi_{ji}\mu_{k-1}^{(j)}, \tag{4a}$$
$$\mu_{k-1|k-1}^{(j|i)} = \frac{\pi_{ji}\mu_{k-1}^{(j)}}{c^{(i)}}, \tag{4b}$$
$$\bar{x}_{k-1|k-1}^{(i)} = \sum_{j=1}^{M}\mu_{k-1|k-1}^{(j|i)}\hat{x}_{k-1|k-1}^{(j)}\,, \tag{4c}$$
$$\bar{P}_{k-1|k-1}^{(i)} = \sum_{j=1}^{M}\mu_{k-1|k-1}^{(j|i)}\big(P_{k-1|k-1}^{(j)} + X_{k-1|k-1}^{(i,j)}\big) \tag{4d}$$

with $X_{k|k}^{(i,j)} = (\bar{x}_{k|k}^{(i)} - \hat{x}_{k|k}^{(j)})(\bar{x}_{k|k}^{(i)} - \hat{x}_{k|k}^{(j)})^\top$, the conditional model probability $\mu_{k-1|k-1}^{(j|i)}$ for transitioning from $m^{(j)}$ to $m^{(i)}$, the state estimate of each filter $\hat{x}_{k-1|k-1}^{(i)}$, its covariance $P_{k-1|k-1}^{(i)}$, the mixing of the state estimates $\bar{x}_{k-1|k-1}^{(i)}$, and its covariance $\bar{P}_{k-1|k-1}^{(i)}$. In the filtering step, each of the M filters is executed separately as in (1) and (2), to obtain the innovation residuals $\tilde{y}_k^{(i)}$ and their covariances $S_k^{(i)}$, as well as $\hat{x}_{k|k}^{(i)}$ with covariances $P_{k|k}^{(i)}$. In the probability update step, the filters' innovation residuals are used to update the model probabilities

$$L_k^{(i)} = \frac{\exp(-\frac{1}{2}\tilde{y}_k^{(i)\top}{S_k^{(i)}}^{-1}\tilde{y}_k^{(i)})}{|2\pi S_k^{(i)}|^{1/2}}, \tag{5a}$$
$$\mu_k^{(i)} = \frac{c^{(i)}L_k^{(i)}}{\sum_{j=1}^{M}c^{(j)}L_k^{(j)}}\,, \tag{5b}$$

where $L_k^{(i)}$ is the likelihood of the observation using $\tilde{y}_k^{(i)}$ and the updated model probability $\mu_k^{(i)}$. In the combination step, the filters' state estimates and their covariances are mixed, weighted by the updated model probabilities:

$$\hat{x}_{k|k} = \sum_{i=1}^{M}\mu_k^{(i)}\hat{x}_{k|k}^{(i)}\,, \tag{6a}$$
$$P_{k|k} = \sum_{i=1}^{M}\mu_k^{(i)}\big(P_{k|k}^{(i)} + (\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})(\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})^\top\big). \tag{6b}$$

II. Problem Statement

We consider the problem of estimating and predicting the state of N vehicles. In Bayesian estimation terms, we want to calculate the Probability Density Function (PDF)

$$p(x_{k:k+T}^{1:N}|y_{1:k}^{1:N})\,, \tag{7}$$

where $k \in \mathbb{N}$ is the current time, $x_k^\eta \in \mathbb{R}^{n_x}$ is the state of the η-th vehicle to be estimated and $x_{k+1:k+T}^\eta$ are its future states to be predicted, $y_{1:k}^\eta$ are measurements, and $T \in \mathbb{N}_{>0}$ is the prediction horizon. We work under the assumption of real traffic being collision-free and we want the predicted trajectories to exhibit the same behavior. Let the area occupied by the η-th vehicle be $\mathcal{P}^\eta(x_k^\eta) \in \mathbb{R}^2$. The condition of vehicle η not colliding with vehicle ν is formulated as $\mathcal{P}^\eta(x_k^\eta) \cap \mathcal{P}^\nu(x_k^\nu) = \emptyset$. Hence, the requirement of generating non-colliding predictions is formally stated as

$$\bigcup_{t=k+1}^{k+T} \mathcal{P}^\eta(\mathbb{E}[x_t^\eta|y_{1:k}^{1:N}]) \cap \mathcal{P}^\nu(\mathbb{E}[x_t^\nu|y_{1:k}^{1:N}]) = \emptyset \;\forall \eta \neq \nu, \tag{8}$$

i.e., we enforce the non-collision requirement in expectation.

The rest of this paper is organized as follows. First, we address the sub-problem of estimating and predicting a single vehicle, under the assumption that the trajectories of the other traffic participants are known (Section III). Then, we estimate and predict multiple vehicles using a hierarchical approach that uses priorities in order to formulate the multiple-vehicle problem as N single-vehicle estimation and prediction problems (Section IV).

III. Single-Vehicle Estimation

For a single vehicle with index N, (7) simplifies to

$$p(x_{k:k+T}^N|y_{1:k}^N, x_{1:k+T}^{1:N-1}).$$

In order to ease exposition, the index N is omitted from x_k^N, y_k^N and we denote the other vehicles' states $x_k^{1:N-1}$ by χ_k. We model the vehicle's behavior as linear, time-varying with

$$x_{k+1} = F_k(\chi_k, z_k)x_k + g_k(\chi_k, z_k)\,, \tag{9}$$

where z_k denotes all external variables that affect the driver's behavior, such as the intention of the driver (e.g., what speed to maintain, whether to change lanes, etc.). Modeling the exact dependence between the vehicle's state x_k and the environment z_k is in general intractable and context-dependent. In this work, we address lane-based driving scenarios, which are structured and allow us to approximate (9) with a finite set of maneuvers $\mathcal{M} = \{m^{(1)}, \dots, m^{(M)}\}$ modeled as linear and time-varying:

$$x_{k+1} = F_k^{(i)}(\chi_k)x_k + E_k^{(i)}(\chi_k) + w_k^{(i)}, \tag{10}$$

where $i = 1, \dots, M$ denotes which of the modes $m^{(1)}$, …, $m^{(M)}$ is active at time k. The process noise accounts for the modeling errors introduced by the approximation in (10). We model the transitions between the different modes of $\mathcal{M}$ as described by (3) allowing for estimating the active model m_k with an IMM-KF. For generating predictions, we use the most likely model m, i.e.,

$$m = \arg\max_i \mu_k^{(i)}. \tag{11}$$

A. Intention-based motion models

In the following, we propose intention-based motion models $m^{(i)}$ to be used in (10). Each motion model corresponds to a specific maneuver of a vehicle and the IMM-KF is used to identify which of the motion models is active.

1) Longitudinal motion: We propose two longitudinal motion models, Velocity Tracking (VT) and Distance Keeping (DK), with the longitudinal state

$$x_{\text{lon},k} \coloneqq \begin{bmatrix} p_{\text{lon},k} & v_{\text{lon},k} & a_{\text{lon},k} \end{bmatrix}^\top,$$

where $p_{\text{lon},k}, v_{\text{lon},k}, a_{\text{lon},k} \in \mathbb{R}$ are the longitudinal position, velocity, and acceleration, respectively.

Velocity tracking: The VT model captures driving maneuvers in which the traffic participant tracks a certain velocity reference. The VT vehicle dynamics is given by

$$x_{\text{lon},k+1} = \begin{bmatrix} 1 & T_s & T_s^2/2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} x_{\text{lon},k} + \begin{bmatrix} 0 \\ T_s^2/2 \\ T_s \end{bmatrix} u_{\text{lon},k}^{(\text{VT})}\,,$$

where $T_s \in \mathbb{R}_{>0}$ is the sampling time and $u_{\text{lon},k}^{(\text{VT})} \in \mathbb{R}$ is the control input that represents the intention of a driver to track a velocity. We model the VT driver input as an LQR-based feedback controller to track the target reference velocity:

$$u_{\text{lon},k}^{(\text{VT})} = -K_{\text{LQR}}^{(\text{VT})} \left(\begin{bmatrix} v_{\text{lon},k} & a_{\text{lon},k} \end{bmatrix}^\top - \begin{bmatrix} v_{\text{ref},k} & 0 \end{bmatrix}^\top \right)$$

with the velocity reference $v_{\text{ref},k}$. As the intended velocity reference $v_{\text{ref},k}$ of the driver is unknown, we augment the state $x_{\text{lon},k}$ such that both $x_{\text{lon},k}$ and $v_{\text{ref},k}$ are estimated simultaneously. Overall, the VT model for the filter design is given by

$$\begin{bmatrix} x_{\text{lon},k+1} \\ v_{\text{ref},k+1} \end{bmatrix} = F_{\text{lon}}^{(\text{VT})}(K_{\text{LQR}}^{(\text{VT})}) \begin{bmatrix} x_{\text{lon},k} \\ v_{\text{ref},k} \end{bmatrix} + w_k^{(\text{VT})}\,, \tag{12}$$

where $w_k^{(\text{VT})}$ allows for tracking variations in $v_{\text{ref},k}$.

Distance keeping: The DK model captures cruise-control maneuvers in which a traffic participant keeps a specific distance from a leading vehicle. The discrete-time dynamics of the DK model is

$$\tilde{x}_{\text{lon},k+1} = \begin{bmatrix} 1 & T_s & T_s^2/2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \tilde{x}_{\text{lon},k} + \begin{bmatrix} T_s^3/6 \\ T_s^2/2 \\ T_s \end{bmatrix} u_{\text{lon},k}^{(\text{DK})}\,,$$
$$\tilde{x}_{\text{lon},k} = x_{\text{lon},k} - x_{\text{lead},k},$$

where $u_{\text{lon},k}^{(\text{DK})} \in \mathbb{R}$ is the driver's control input for the DK model and $x_{\text{lead},k}$ is the longitudinal state of the leading vehicle. We model the DK input as an LQR controller to track a longitudinal time gap $t_{\text{gap},k}$ to a leading vehicle

$$u_{\text{lon},k}^{(\text{DK})} = -K_{\text{LQR}}^{(\text{DK})} \left(\tilde{x}_{\text{lon},k} - \begin{bmatrix} -v_{\text{lead},k} t_{\text{gap},k} & 0 & 0 \end{bmatrix}^\top \right).$$

Similarly to (12), we augment the state with the reference time gap $t_{\text{gap},k}$ to be estimated and thus, the DK model for the filter design is given by

$$\begin{aligned} \begin{bmatrix} x_{\text{lon},k+1} \\ t_{\text{gap},k+1} \end{bmatrix} &= F_{\text{lon},k}^{(\text{DK})}(K_{\text{LQR}}^{(\text{DK})}, \chi_k) \begin{bmatrix} x_{\text{lon},k} \\ t_{\text{gap},k} \end{bmatrix} \\ &\quad + E_{\text{lon},k}^{(\text{DK})}(K_{\text{LQR}}^{(\text{DK})}, \chi_{k:k+1}) + w_k^{(\text{DK})}. \end{aligned} \tag{13}$$

2) Lateral motion: We model lateral maneuvers with multiple Lane Changing (LC) motion models, which involve the lateral state

$$x_{\text{lat},k} \coloneqq \begin{bmatrix} p_{\text{lat},k} & v_{\text{lat},k} & a_{\text{lat},k} \end{bmatrix}^\top,$$

where $p_{\text{lat},k}, v_{\text{lat},k}, a_{\text{lat},k} \in \mathbb{R}$ are the lateral position, velocity, and acceleration, respectively. The total number of lateral models $m^{(\text{LC})}$ is in general equal to the number of lanes of the road. However, we focus only on lane change maneuvers of up to one lane at a time, effectively leading to three $m^{(\text{LC})}$ models: one for staying on the lane, one for a lane change maneuver to the left, and one for a lane change to the right, denoted $m^{(\text{LC1})}$, $m^{(\text{LC2})}$, and $m^{(\text{LC3})}$, respectively. This is a reasonable assumption as maneuvers with multiple, consecutive lane changes are captured by the algorithm due to its recursive implementation, i.e., as soon as the a lane change is detected, the algorithm can predict a subsequent lane change at the next sampling time.

The LC model captures lateral driving maneuvers in which the vehicle tracks the lateral position of a lane. We model the discrete-time dynamics of the LC model as

$$x_{\text{lat},k+1} = \begin{bmatrix} 1 & T_s & T_s^2/2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} x_{\text{lat},k} + \begin{bmatrix} T_s^3/6 \\ T_s^2/2 \\ T_s \end{bmatrix} u_{\text{lat},k}^{(\text{LC})},$$

where $u_{\text{lat},k}^{(\text{LC})} \in \mathbb{R}$ is the driver's control input used to track a certain lane. Again, we assume that the driver employs a linear feedback controller to converge to a reference lane

$$u_{\text{lat},k}^{(\text{LC})} = -K_{\text{LQR}}^{(\text{LC})} \left(x_{\text{lat},k} - \begin{bmatrix} p_{\text{ref}} & 0 & 0 \end{bmatrix}^\top \right),$$

where p_{ref} is the lateral position of the lane being tracked. This LQR-based tracking model is reasonable because of the jerk-minimizing trajectories drivers usually employ to increase comfort [20]. In order to make the lane-change maneuver as realistic as possible, we use an inverse optimal control scheme [21] to learn $K_{\text{LQR}}^{(\text{LC})}$ from vehicle data from the NGSIM dataset. Finally, the VT model for the filter design is

$$x_{\text{lat},k+1} = F_{\text{lat}}^{(\text{LC})}(K_{\text{LQR}}^{(\text{LC})}) x_{\text{lat},k} + E_{\text{lat}}^{(\text{LC})}(K_{\text{LQR}}^{(\text{LC})}, p_{\text{ref}}) + w_k^{(\text{LC})},$$

where $w_k^{(\text{LC})}$ allows for tracking variations in the reference lane's lateral position.

Remark 1: For simplicity, we learn one set of parameters, $K_{\text{LQR}}^{(\text{LC})}$, for all drivers. It is similarly possible to consider multiple models of differing driving styles, e.g., aggressive or timid, or to learn one $K_{\text{LQR}}^{(\text{LC})}$ for each driver. The individuality of different drivers is highlighted, e.g., in [8], [22].

3) Joint motion: Finally, the three lateral motion models are combined with the two proposed longitudinal motion models, i.e., in total there are six maneuvers in our filter design: $\mathcal{M} = \{\text{VT-LC1, VT-LC2, VT-LC3, DK-LC1, DK-LC1, DK-LC1}\}$.

The joint motion models involve the full state of the vehicle $x_k \coloneqq [x_{\text{lon},k}^\top \; x_{\text{lat},k}^\top]^\top$ as a *common* state shared among models. Each model however has its own separate full state $x_k^{(i)}$ related to the common state through $x_k = V x_k^{(i)}$, where $x_k^{(i)} \in \mathbb{R}^{n_x^{(i)}}$ is the full state of model $m^{(i)}$ and V is a selection matrix. In our case, $n_x^{(i)} = 7$ and $V \in \mathbb{R}^{6\times 7}$ is the matrix that selects all states but the reference states $v_{\text{ref},k}$ and $t_{\text{gap},k}$. The joint motion dynamics of each model is given by

$$x_{k+1}^{(i)} = F_k^{(i)} x_k^{(i)} + E_k^{(i)} + w_k^{(i)},$$

where each $F_k^{(i)}, E_k^{(i)}$ corresponds to one of the six maneuvers in the filter design and consists of a longitudinal and a lateral motion model:

$$F_k = \begin{bmatrix} F_{\text{lon},k} & \mathbb{0} \\ \mathbb{0} & F_{\text{lat},k} \end{bmatrix}, \; E_k = \begin{bmatrix} E_{\text{lon},k} \\ E_{\text{lat},k} \end{bmatrix}.$$

B. Non-colliding predictions for single vehicle

In the following, we show how to use the six proposed intention-based motion models to generate non-colliding predictions. As we only use the most probable model m in the prediction step, we omit the index m in the exposition.

1) Distribution of predictions: First, we derive the distribution of the motion predictions. Given that the state estimate is Gaussian distributed with $p(x_k|y_{1:k}, x_{1:k+T}^{1:N-1}) = \mathcal{N}(\hat{x}_{k|k}, P_{k|k})$ and the proposed filter models are linear, the propagated state predictions are also Gaussian:

$$p\left(x_t|y_{1:k}, x_{1:k+T}^{1:N-1}\right) = \mathcal{N}\left(\hat{x}_{t|k}, P_{t|k}\right)$$

for $t = k+1, \ldots, k+T$, where the state predictions $\hat{x}_{t|k}$ and the covariance matrices $P_{t|k}$ are given by

$$\hat{x}_{t|k} = \Phi(t,k)\hat{x}_{k|k} + \sum_{i=k+1}^{t} \Phi(t,i) E_i \,, \tag{14a}$$

$$P_{t|k} = \Phi(t,k) P_{k|k} \Phi(t,k)^\top + \sum_{i=k+1}^{t} \Phi(t,i) Q_t \Phi(t,i)^\top, \tag{14b}$$

$$\Phi(t,i) = \begin{cases} (\prod_{j=i}^{t-1} F_j^\top)^\top & \text{if } t > i \\ I & \text{if } t = i. \end{cases} \tag{14c}$$

2) Collision constraints: Next, we formulate a non-collision condition that is subsequently used as constraint in an optimization framework. We model the geometrical shape of vehicles as rectangles with constant and known width W^i and length L^i of the i-th vehicle. For the single vehicle N, the condition in (8) simplifies to

$$\bigcup_{t=k+1}^{k+T} \mathcal{P}^N(\hat{x}_{t|k}) \cap \mathcal{P}^\eta(x_k^\eta) = \emptyset \quad \forall \eta = 1, ..., N-1 \tag{15}$$

with the motion prediction $\hat{x}_{t|k} = \mathbb{E}[x_t|y_{1:k}, x_{1:k+T}^{1:N-1}]$ of vehicle N. Due to the structure of the lateral motion models in Section III-A2, each model corresponds to a specific lane. Hence, the model m that is used for generating the predictions fully defines the predicted lateral position $\hat{p}_{\text{lat},t}$ of the vehicle. Thus, (15) is equivalent to satisfying

$$\bigwedge_{t\in\mathcal{I}^\eta} \left(|\hat{p}_{\text{lon},t|k} - p_{\text{lon},t}^\eta| > \frac{L^N + L^\eta}{2} \right) \; \forall \eta = 1, ..., N-1 \tag{16}$$

with the index set

$$\mathcal{I}^\eta = \left\{ t = k+1, \ldots, k+T : |\hat{p}_{\text{lat},t|k} - p_{\text{lat},t|k}^\eta| \leq \tfrac{W^N + W^\eta}{2} \right\}$$

defining the time indices in which two vehicles are laterally close to each other. Intuitively, (16) states that for all time-steps in which two vehicles are laterally close to each other, the vehicles must not be close longitudinally. Next, we use the condition in (16) as constraint in our optimization-based projection to generate non-colliding motion predictions.

3) Estimate projection: In order to obtain non-colliding predictions satisfying (16), we project the state estimate in (14a) as follows:

$$\min_{x\in\mathbb{R}^{n_x}} \ (x-\hat{x}_{k|k})^\top W(x-\hat{x}_{k|k}) \tag{17a}$$

$$\text{s.t. } \hat{x}_{t|k} = \Phi(t,k)x + \sum_{i=k+1}^{t} \Phi(t,i)E_i \tag{17b}$$

$$\hat{x}_{t|k} \text{ satisfies (16) } \forall t = k+1,\dots,k+T, \tag{17c}$$

where we choose $W \succ 0$ such that the projection of the reference estimates v_{ref} and t_{gap} is encouraged rather than x_{lon} and x_{lat}, i.e., we incorporate a non-collision intention into v_{ref} and t_{gap}. The non-convex optimization problem (17) is not in a form that can be directly handled by off-the-shelf optimization solvers because of the disjunctions in (17c). However, (17) can be reformulated into a tractable, equivalent Mixed-Integer Quadratic Program (MIQP) using the Big-M method [23]:

$$\min_{x\in\mathbb{R}^{n_x},z_{\eta,t}\in\mathbb{B}} \ (x-\hat{x}_{k|k})^\top W(x-\hat{x}_{k|k}) \tag{18a}$$

$$\text{s.t. } \hat{x}_{t|k} = \Phi(t,k)x + \sum_{i=k+1}^{t} \Phi(t,i)E_i \tag{18b}$$

$$\hat{p}_{\text{lon},t|k} \geq (p^{\eta}_{\text{lon},t} + \bar{L}^{\eta})z_{\eta,t} - M(1-z_{\eta,t}) \tag{18c}$$

$$\hat{p}_{\text{lon},t|k} \leq (p^{\eta}_{\text{lon},t} - \bar{L}^{\eta})(1-z_{\eta,t}) + Mz_{\eta,t} \tag{18d}$$

for all $\eta = 1,\dots,N-1$ and $t \in \mathcal{I}^{\eta}$, where $\bar{L}^{\eta} = \frac{L^N+L^{\eta}}{2}$. As a result, the motion prediction

$$\hat{x}_{t|k} = \Phi(t,k)\hat{x}^{\text{proj}}_{k|k} + \sum_{i=k+1}^{t} \Phi(t,i)E_i \tag{19}$$

satisfies (15), where $\hat{x}^{\text{proj}}_{k|k}$ is the minimizer of (18).

4) Projection pseudo-measurement: The minimizer $\hat{x}^{\text{proj},(i)}_{k|k}$ of (18) realizes non-colliding predictions for each motion model $m^{(i)}$. Additionally, we want to explicitly consider the non-colliding motion predictions in the IMM-KF for choosing the prediction model $m^{(i)}$ as in (11), i.e., if a maneuver exhibits high projection cost in (18a), it should be considered less likely. In order to make the models whose unprojected maneuvers exhibit collisions less probable, we augment the physical measurements' residuals $\tilde{y}^{(i)}_k$ of each model $m^{(i)}$ with the pseudo-measurement

$$\tilde{y}^{(i)}_{\text{pseudo},k} = (\hat{x}^{\text{proj},(i)}_{k|k} - \hat{x}^{(i)}_{k|k})^\top W(\hat{x}^{\text{proj},(i)}_{k|k} - \hat{x}^{(i)}_{k|k}) \tag{20}$$

and use the augmented residual and augmented covariance to compute the likelihood $L^{(i)}_k$ in (5a):

$$\tilde{y}^{(i)}_{\text{aug},k} = \begin{bmatrix} \tilde{y}^{(i)}_k \\ \tilde{y}^{(i)}_{\text{pseudo},k} \end{bmatrix}, \ S^{(i)}_{\text{aug},k} = \begin{bmatrix} S^{(i)}_k & \mathbb{0} \\ \mathbb{0} & S^{(i)}_{\text{pseudo},k} \end{bmatrix}.$$

Remark 2: The estimates $\hat{x}^{(i)}_{k|k}$ of the filters are not altered by the pseudo-measurement but only the motion predictions and the IMM-KF's probability estimates $\mu^{(i)}_k$.

Remark 3: The minimizers of (17) and (18) are equivalent if M is larger than any possible constraint value (Big-M), i.e., $M \geq \hat{p}_{\text{lon},t|k} \ \forall t = k+1,\dots,k+T$.

Remark 4: In this work, we consider motion predictions with up to one lane change over the prediction horizon. Hence, the predicted vehicle cannot be both in front and in the back of vehicle η over the prediction horizon as such a maneuver would require two lane changes. As a consequence, we can fix $z_{\eta,t} = z_{\eta} \ \forall t = k+1,\dots,k+T$ in (18), which reduces the number of binary variables to one per vehicle η and renders (18) computationally more efficient.

Remark 5: Problem (18) can be solved efficiently as n_x is small and there are algorithms that solve combinatorial problems efficiently, e.g., branch and bound techniques [24]. We highlight the algorithm's real-time capabilities and provide computation times in Section V.

C. Overall algorithm for single vehicle

Algorithm 1 summarizes the single-vehicle procedure given the obstacle vehicles' trajectories $x^{1:N-1}_{k:k+T}$, where Step 2–6 is the state estimation and Step 7–11 is the motion prediction.

Algorithm 1 Single-vehicle estimation: recursion, time k

1: **input:** y_k, $\hat{x}^{(i)}_{k-1|k-1}$, $P^{(i)}_{k-1|k-1}$, $\mu^{(i)}_{k-1}$, $x^{1:N-1}_{k:k+T}$
2: Run interaction step of IMM-KF (4)
3: Re-init. KFs with $\bar{x}^{(i)}_{k-1|k-1}, \bar{P}^{(i)}_{k-1|k-1} \ \forall m^{(i)}$
4: Run KF (1), (2) $\forall m^{(i)}$
5: Project (18) and get pseudo measurement (20) $\forall m^{(i)}$
6: Run IMM-KF (5), (6) $\forall m^{(i)}$
7: Find most probable model $m \leftarrow \arg\max_i \mu^{(i)}_k$
8: Project state estimate $\hat{x}^{(m)}_{k|k}$ for model m, $\hat{x}^{\text{proj}}_{k|k} \leftarrow$ (18)
9: **for** $t \leftarrow k+1,\dots,k+T$
10: Predict $\hat{x}^{(m)}_{t|k}, P^{(m)}_{t|k}$ with (14), (19) for model m
11: Compute $\hat{x}_{t|k}, P_{t|k} \leftarrow V\hat{x}^{(m)}_{t|k}, VP^{(m)}_{t|k}V^\top$
12: **return:** $\hat{x}_{k:k+T}, P_{k:k+T}, \hat{x}^{(i)}_{k|k}, P^{(i)}_{k|k}, \mu^{(i)}_k$

IV. Priority-based Multiple-Vehicle Estimation

We address multiple-vehicle estimation in a hierarchical approach, where the hierarchy is given by a priority list and dynamically adapted using ideas of the bubble sort algorithm [25]. The proposed sorting procedure exploits the rules of the road in which trailing vehicles must avoid collisions with a leading vehicle.

We propose the following sorting strategy:

- If two vehicles are on the same lane, the vehicle in front has a higher priority.
- If two vehicles are on different lanes, the one with larger longitudinal progress over the prediction horizon T has higher priority.

It is easy to see that it is always possible to derive a sorted list l^{sort} using these two criteria.

Formally, the priority list is stated as

$$l^{\text{sort}} = [l^1, l^2, \dots, l^{N-1}, l^N], \tag{21}$$

where $l^j = \eta$ indicates that vehicle η has priority j. Note that priority 1 is the highest and priority N is the lowest. Consequently, the obstacle vehicles that vehicle η encounters are given by the subset of vehicles

$$\mathcal{L}^\eta := \{l^1, \dots, l^{\eta-1}\}. \tag{22}$$

This hierarchical approach allows us to decouple the multiple-vehicle estimation problem into multiple single-vehicle estimation problems that are tractably evaluated as in Section III, which is shown in Lemma 1 with proof in the appendix.

Lemma 1: The Bayesian tracking problem of the PDF in (7) can be decomposed into N separate Bayesian tracking problems using the priority list in (21).

Algorithm 2 summarizes the overall recursive procedure for the estimation and motion prediction of multiple vehicles.

Algorithm 2 Multiple-vehicle estimation: recursion, time k

input: l^{sort}, $y_k^{1:N}$, $\hat{x}_{k-1|k-1}^{1:N}$, $P_{k-1|k-1}^{1:N}$, $\mu_{k-1}^{1:N}$
for $\eta \leftarrow l^1, \dots, l^N$ ▷ descending priority order
  obstacles $\leftarrow \hat{x}_{k:k+T|k}^{l^{1:\eta-1}}$
  $\hat{x}_{k:k+T|k}^{\eta}, P_{k:k+T|k}^{\eta} \leftarrow$ Single-Veh. Est. ▷ Alg. 1
$l^{\text{sort}} \leftarrow$ Priority List Sort
return $l^{\text{sort}}, \hat{x}_{k:k+T}^{1:N}, P_{k:k+T}^{1:N}$

V. Simulation Results

A. Illustrative example

We first examine a synthetic scenario to illustrate our method. Fig. 1a shows the scenario in which a dangerous lane change is performed by vehicle 3 going from the slow right lane to the fast left lane with a constant longitudinal velocity of $19.4\,\mathrm{m/s}$. In front of vehicle 3 on the right lane, vehicle 2 travels with a constant velocity of $19.4\,\mathrm{m/s}$. On the left lane, vehicle 1 traveling with a higher velocity of $25\,\mathrm{m/s}$ has to react to this sudden lane change by braking and keeping a distance from vehicle 3.

We predict the motion plans for the three vehicles with (a) the well-established Constant Velocity (CV) and Constant Acceleration (CA) models, (b) the VT-DK models with a static priority list (from lowest to highest: vehicle 3, 1, 2), and (c) the VT-DK models with the dynamically adapted priority list. Fig. 2 shows a snapshot of the scenario, the moment when vehicle 3 merges into the left lane in front of vehicle 1. The CV-CA models shown in (a) predict a collision where vehicle 1 ends up in front of vehicle 3; the VT-DK models with the static priority list shown in (b) produce a non-colliding prediction but incorrectly predict that vehicle 3 aborts its lane merging maneuver; whereas when the priority list is sorted online (c), we correctly predict that vehicle 1 will brake to keep a safe distance from vehicle 3. We have included a supplementary video to illustrate this scenario and highlight the advantages of the intention-based models in conjunction with the sorting algorithm of the priority list.

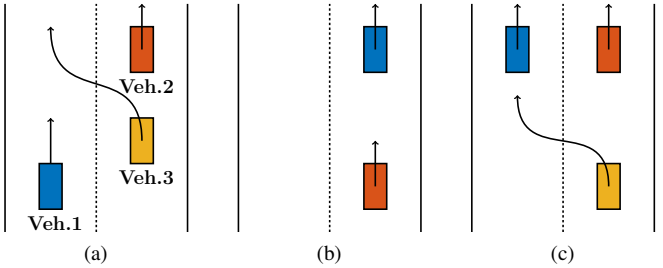

Figure 1. Test cases for evaluation of the proposed scheme. (a) Synthetic dangerous lane change scenario, (b) NGSIM cruise control scenarios involving keeping distance from a leading vehicle without changing lanes, and (c) NGSIM lane change scenarios involving changing between two lanes with both being blocked by leading vehicles.

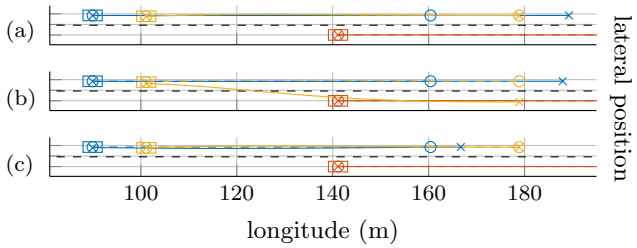

Figure 2. Snapshot of dangerous lane change scenario using (a) the CV-CA models, (b) the VT-DK models with a constant priority list, and (c) the VT-DK models with the priority list sorted online. The predictions for vehicle (solid lines) and their start/end point (crosses) are displayed, along with the actual trajectories (dashed lines) and their start/end points (circles).

B. Real-life vehicle data

In order to systematically evaluate the overall performance of our method, we use the NGSIM I-80 dataset [26], which consists of 45 minutes of vehicle trajectory data with $0.1\,\mathrm{s}$ sampling time. We extract specific test cases from the dataset that correspond to typical highway driving scenarios: cruise control (Fig. 1b) and lane change (Fig. 1c). In total, we predict 244 cruise control and 115 lane change scenarios with prediction horizons of $1\,\mathrm{s}$, $2\,\mathrm{s}$, $3\,\mathrm{s}$, and $4\,\mathrm{s}$.

Fig. 3 compares the predicted motion plans resulting from Algorithm 2 with the ones resulting from the CV-CA models. It presents the means as well as the standard deviations of the position prediction errors at the end of the prediction horizon. It can be seen that for short prediction horizons, the two model sets show comparable performance. For longer predictions, however, the VT-DK model set produces smaller prediction errors (between $30.5\,\%$ and $53.3\,\%$ for $4\,\mathrm{s}$) and reduced uncertainty about its predictions (between $59.7\,\%$ and $68.6\,\%$ for $4\,\mathrm{s}$). This can be attributed to the intention-based nature of non-colliding predictions, along with the stable VT and DK models that limit the growth of the uncertainty.

C. Real-time computation capabilities

Next, we evaluate the real-time capability of the proposed algorithm. The only computationally demanding step of our method is solving the projection optimization problems of (18). We evaluate the computational burden by solving 784 MIQPs from the NGSIM test cases using FORCES PRO [27], [28] for a

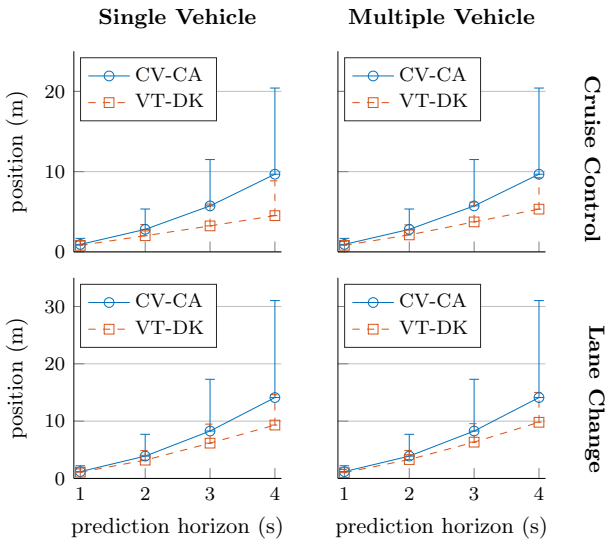

Figure 3. Mean longitudinal position prediction errors using models CV-CA (blue circles) and VT-DK (red squares) and their corresponding standard deviations, for prediction horizons of 1 s, 2 s, 3 s, and 4 s. The results are obtained from single/multiple-vehicle predictions of many cruise control/lane change scenarios from the NGSIM dataset.

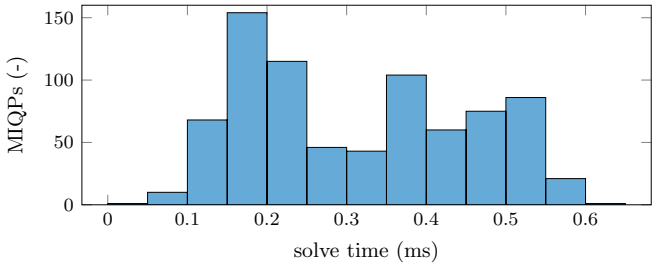

Figure 4. Histogram of exemplary projection MIQP solve times of (18) using FORCES PRO [27], [28] for a prediction horizon of 4 s ($T = 40$).

prediction horizon of 4 s ($T = 40$). The resulting solve times are displayed in Fig. 4. FORCES PRO needs approximately 0.2 ms to 0.5 ms for most of the solves. Assuming a conservative average solve time of 0.5 ms per MIQP, the projection of all $M = 6$ models of $\mathcal{M}$ takes 3 ms per IMM-KF recursion. For a local cluster of $N = 10$ vehicles, evaluating each of them sequentially with Algorithm 2 takes 30 ms. This is well within the available time margin (e.g., for a 0.1 s sampling time), thus making the method executable in real-time. This can be further improved by parallelizing the M MIQPs of the IMM-KF. The computations were carried out on an Intel i7 CPU at 3.60 GHz with 32 GB memory.

VI. Discussion

In this section, we compare the performance of the proposed prediction scheme with state-of-the-art deep learning-based approaches that similarly use the NGSIM dataset. Table I reports the Root Mean Squared Error (RMSE) of trajectory predictions for the proposed method and [11]–[13] for various prediction horizons. The RMSE is computed over all prediction time-steps and trajectories and is also used in the review [29] as a unified evaluation metric. The RMSE values for [11]–[13] are taken from [29] and are also stated in the respective references. The computation time for the proposed method results from the analysis in Section V-C with 0.03 s for a cluster of 10 vehicles. Like many papers on deep learning-based methods, [11]–[13] did not report their methods' corresponding computation times; a metric that is crucial in autonomous driving applications [29]. Conceptually similar approaches are typically computationally more expensive, e.g., with 0.35 s and 0.06 s for 10 vehicles, cf. [14], [15].

Table I
Comparison with Related Works on Trajectory Prediction

Work	RMSE (m)				
	1 s	2 s	3 s	4 s	5 s
[11]	0.72	2.00	3.76	5.97	9.01
[12]	0.49	1.41	2.60	4.06	5.79
[13]	0.56	1.19	1.93	2.78	3.76
ours	0.58	1.36	2.28	3.37	4.55

The proposed method exhibits one of the lowest RMSEs for smaller prediction horizons, while also being comparable to state-of-the-art results for prediction horizons up to 5 s. While exhibiting comparable RMSEs, our method has some favorable aspects compared to deep learning-based techniques. First, our method requires only a few lane-change maneuvers to calibrate $K_{\text{LQR}}^{(\text{LC})}$, whereas deep learning-based methods need a significant part of the NGSIM dataset to train their algorithms (i.e., thousands of trajectories). Second, our method has the smallest computation time out of the available results, supporting its real-time capabilities. In contrast to most of the related works, our method is easily embeddable in typical autonomous driving hardware configurations, without e.g., the need for powerful GPUs [11], [12], [15].

The presented results rely on the priority-based estimation technique, as well as the lane-based prediction models. For scenarios that significantly differ from regular (and legal) driving behavior, the prediction accuracy may deteriorate, e.g., if a motorcycle violates the rules of the road by traveling in-between lanes. The handling of such corner cases can be addressed via extensions of the proposed method.

VII. Conclusion

This paper proposed a framework for motion estimation and prediction of traffic participants in highway driving scenarios. First, we focused on estimation of a single vehicle and designed an IMM-KF-based algorithm using a novel, intention-based model set, consisting of typical driving maneuvers exhibited by traffic participants during highway driving. Further, we extended the algorithm to generate non-colliding predictions using projections onto a feasible collision-free space. Lastly, we addressed motion estimation and prediction for multiple vehicles by assuming a hierarchical structure encoded through a vehicle priority list. Simulation results with the NGSIM dataset showed (i) increased motion prediction accuracy compared to similar approaches in literature that use an IMM-KF and comparable results with state-of-the-art deep learning-based methods, and (ii) the proposed algorithm's real-time capabilities.

APPENDIX: PROOF OF LEMMA 1

Proof: Without loss of generality, let the priority list l in (21) be sorted in ascending order according to the vehicles' indices. Then, the index sets in (22) take the form

$$\mathcal{L}^{\eta} = \{1, \dots, \eta - 1\} \quad \forall \eta = 1, \dots, N. \tag{23}$$

The PDF in (7) is evaluated as

$$p(x_k^{1:N}|y_{1:k}^{1:N}) = \sum_{m_k^{1:N}} p(x_k^{1:N}, m_k^{1:N}|y_{1:k}^{1:N}) .$$

Using Bayes' theorem, we obtain

$$\begin{aligned} &p(x_k^{1:N}, m_k^{1:N}|y_{1:k}^{1:N}) \\ &= \frac{p(y_k^{1:N}|x_k^{1:N}, m_k^{1:N}) p(x_k^{1:N}, m_k^{1:N}|y_{1:k-1}^{1:N})}{p(y_k^{1:N}|y_{k-1}^{1:N})}, \end{aligned}$$

where $p(y_k^{1:N}|y_{k-1}^{1:N})$ is a normalization constant that can be readily calculated, and

$$p(y_k^{1:N}|x_k^{1:N}, m_k^{1:N}) = \prod_{\eta=1}^{N} p(y_k^{\eta}|x_k^{\eta}, m_k^{\eta}).$$

Further, marginalizing over $x_{k-1}^{1:N}$ and $m_{k-1}^{1:N}$ yields

$$\begin{aligned} &p(x_k^{1:N}, m_k^{1:N}|y_{1:k-1}^{1:N}) \\ &= \int \sum_{m_{k-1}^{1:N}} p(x_k^{1:N}, m_k^{1:N}, x_{k-1}^{1:N}, m_{k-1}^{1:N}|y_{1:k-1}^{1:N}) \, \mathrm{d}x_{k-1}^{1:N} \end{aligned}$$

where, using conditional probabilities and the Bayesian network's structure, yields

$$\begin{aligned} &p(x_k^{1:N}, m_k^{1:N}, x_{k-1}^{1:N}, m_{k-1}^{1:N}|y_{1:k-1}^{1:N}) \\ &= p(x_k^{1:N}, m_k^{1:N}|x_{k-1}^{1:N}, m_{k-1}^{1:N}, y_{1:k-1}^{1:N}) p(x_{k-1}^{1:N}, m_{k-1}^{1:N}|y_{1:k-1}^{1:N}) \\ &= p(x_k^{1:N}, m_k^{1:N}|x_{k-1}^{1:N}, m_{k-1}^{1:N}) p(x_{k-1}^{1:N}, m_{k-1}^{1:N}|y_{1:k-1}^{1:N}). \end{aligned}$$

The term $p(x_{k-1}^{1:N}, m_{k-1}^{1:N}|y_{1:k-1}^{1:N})$ is known due to the recursive nature of the employed Bayesian tracking scheme and

$$\begin{aligned} &p(x_k^{1:N}, m_k^{1:N}|x_{k-1}^{1:N}, m_{k-1}^{1:N}) \\ &= \prod_{\eta=1}^{N} p(x_k^{\eta}, m_k^{\eta}|x_{k-1}^{1:N}, m_{k-1}^{1:N}, x_k^{1:\eta-1}, m_k^{1:\eta-1}) , \end{aligned} \tag{24}$$

as we can evaluate the individual PDFs of (24) in hierarchical order according to (23). Finally, the predictions of the vehicles' states $x_{k+1:k+T}^{1:N}$ can be evaluated by repeating (in hierarchical order) the IMM-KF steps for each vehicle. ■

REFERENCES

[1] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, 2016.

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

[3] C. Barrios, H. Himberg, Y. Motai, and A. Sad, "Multiple model framework of adaptive extended kalman filtering for predicting vehicle location," in *IEEE Intell. Transp. Syst. Conf.*, 2006, pp. 1053–1059.

[4] M. Zhang, S. Knedlik, and O. Loffeld, "An adaptive road-constrained IMM estimator for ground target tracking in GSM networks," in *11th Int. Conf. Inf. Fusion*, 2008, pp. 1–8.

[5] R. Schubert and G. Wanielik, "Unifying bayesian networks and IMM filtering for improved multiple model estimation," in *12th Int. Conf. Inf. Fusion*, 2009, pp. 810–817.

[6] R. Toledo-Moreo and M. A. Zamora-Izquierdo, "IMM-based lane-change prediction in highways with low-cost GPS/INS," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 180–185, 2009.

[7] H. Dyckmanns, R. Matthaei, M. Maurer, B. Lichte, J. Effertz, and D. Stüker, "Object tracking in urban intersections based on active use of a priori knowledge: Active interacting multi model filter," in *IEEE Intell. Veh. Symp. (IV)*, 2011, pp. 625–630.

[8] A. Carvalho, Y. Gao, S. Lefevre, and F. Borrelli, "Stochastic predictive control of autonomous vehicles in uncertain environments," in *12th Int. Symp. Adv. Veh. Control*, 2014.

[9] B. Kim, K. Yi, H. Yoo, H. Chong, and B. Ko, "An IMM/EKF approach for enhanced multitarget state estimation for application to integrated risk management system," *IEEE Trans. Veh. Technol.*, vol. 64, no. 3, pp. 876–889, 2015.

[10] K. Jo, M. Lee, J. Kim, and M. Sunwoo, "Tracking and behavior reasoning of moving vehicles based on roadway geometry constraints," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 2, pp. 460–476, 2017.

[11] F. Altché and A. de La Fortelle, "An LSTM network for highway trajectory prediction," in *IEEE Intell. Trans. Syst. Conf. (ITSC)*, 2017, pp. 353–359.

[12] L. Xin, P. Wang, C. Chan, J. Chen, S. E. Li, and B. Cheng, "Intention-aware long horizon trajectory prediction of surrounding vehicles using dual LSTM networks," in *IEEE Intell. Trans. Syst. Conf. (ITSC)*, 2018, pp. 1441–1446.

[13] S. Dai, L. Li, and Z. Li, "Modeling vehicle interactions via modified LSTM models for trajectory prediction," *IEEE Access*, vol. 7, pp. 38 287–38 296, 2019.

[14] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 1549–15 498.

[15] X. Li, X. Ying, and M. C. Chuah, "GRIP: Graph-based interaction-aware trajectory prediction," in *IEEE Intell. Trans. Syst. Conf. (ITSC)*, 2019, pp. 3960–3966.

[16] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, "Multi-agent tensor fusion for contextual trajectory prediction," in *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 118–12 126.

[17] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs," in *IEEE Intell. Veh. Symp. (IV)*, 2018, pp. 1179–1184.

[18] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Fluids Eng.*, vol. 82, no. 1, pp. 35–45, 1960.

[19] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: a survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 103–123, 1998.

[20] P. Bosetti, M. D. Lio, and A. Saroldi, "On the human control of vehicles: an experimental study of acceleration," *Eur. Transp. Research Rev.*, vol. 6, no. 2, pp. 157–170, 2013.

[21] M. Menner and M. N. Zeilinger, "Convex formulations and algebraic solutions for linear quadratic inverse optimal control problems," in *Eur. Control Conf.*, 2018, pp. 2107–2112.

[22] M. Menner, K. Berntorp, M. N. Zeilinger, and S. Di Cairano, "Inverse learning for data-driven calibration of model-based statistical path planning," *IEEE Trans. Intell. Veh.*, 2020, doi: 10.1109/TIV.2020.3000323.

[23] T. Schouwenaars, B. D. Moor, E. Feron, and J. P. How, "Mixed integer programming for multi-vehicle path planning," in *Eur. Control Conf.*, 2001, pp. 2603–2608.

[24] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations Res.*, vol. 14, no. 4, pp. 699–719, 1966.

[25] D. Knuth, *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, 1998.

[26] U.S. Department of Transportation. (2006, Dec.) Next Generation Simulation (NGSIM). Accessed: 11/05/2019. [Online]. Available: https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

[27] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs," *Int. J. Control*, pp. 1–17, 2017.

[28] A. Domahidi and J. Jerez, "FORCES Professional," Embotech AG, https://embotech.com/FORCES-Pro, 2014–2020.

[29] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," *IEEE Trans. Intell. Trans. Syst.*, pp. 1–15, 2020.