MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

A Framework for Joint Vehicle Localization and Road Mapping Using Onboard Sensors

Berntorp, Karl; Greiff, Marcus

TR2024-163 December 03, 2024

Abstract

This paper presents a modeling framework for joint estimation of a host vehicle state and a map of the road based on global navigation satellite system (GNSS) and camera measurements. We model the road using a spline representation based on lower-dimensional Bézier curves parametrized in generalized endpoints (GEPs) with implicit guarantees of continuous lane boundaries. We model the GEPs by a parameter vector having a Gaussian prior representing the uncertainty of the prior map, and provide a systematic way of defining this prior from generic map representations. Both GNSS and camera measurements, such as lane-mark measurements, have noise characteristics that vary in time. To adapt to the changing noise levels and hence improve positioning performance, we formulate the problem as a joint vehicle state, map parameter, and noise covariance estimation problem and present two noise-adaptive linear-regression Kalman filters (LRKFs); (i) an interacting multiple-model (IMM) LRKF and (ii) a variational-Bayes (VB) LRKF. We conduct a Monte-Carlo study and compare the two approaches in terms of estimation precision and computation times. Embedded implementations in an automotive-grade dSpace Micro Autobox-II indicate the real-time validity of both approaches, with turn-around times of between 2–80ms, depending on the problem size and if the map is updated. The results indicate that while the IMM-LRKF shows marginally better estimation accuracy, the VB-LRKF is at least a factor of 2 faster.

Control Engineering Practice 2024

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

A Framework for Joint Vehicle Localization and Road Mapping Using Onboard Sensors

Karl Berntorp[a,1], Marcus Greiff[a]

[a] *Mitsubishi Electric Research Laboratories, Cambridge, 02139 MA, USA*

Abstract

This paper presents a modeling framework for joint estimation of a host vehicle state and a map of the road based on global navigation satellite system (GNSS) and camera measurements. We model the road using a spline representation based on lower-dimensional Bézier curves parametrized in *generalized endpoints* (GEPs) with implicit guarantees of continuous lane boundaries. We model the GEPs by a parameter vector having a Gaussian prior representing the uncertainty of the prior map, and provide a systematic way of defining this prior from generic map representations. Both GNSS and camera measurements, such as lane-mark measurements, have noise characteristics that vary in time. To adapt to the changing noise levels and hence improve positioning performance, we formulate the problem as a joint vehicle state, map parameter, and noise covariance estimation problem and present two noise-adaptive linear-regression Kalman filters (LRKFs); (i) an interacting multiple-model (IMM) LRKF and (ii) a variational-Bayes (VB) LRKF. We conduct a Monte-Carlo study and compare the two approaches in terms of estimation precision and computation times. Embedded implementations in an automotive-grade dSpace Micro Autobox-II indicate the real-time validity of both approaches, with turn-around times of between 2–80ms, depending on the problem size and if the map is updated. The results indicate that while the IMM-LRKF shows marginally better estimation accuracy, the VB-LRKF is at least a factor of 2 faster.

Keywords: automotive, statistical estimation, adaptive Kalman filtering, SLAM

1. Introduction

High-precision vehicle positioning is becoming increasingly important as vehicles equipped with sophisticated advanced driver assistance systems (ADASs) and even autonomous driving (AD) features are becoming widespread. For such applications, high positioning accuracy is needed for safety-critical obstacle and lane-change maneuvering, and to provide comfortable autonomous vehicle control. While several production ADASs either focus on the longitudinal motion, such as in automated emergency braking (AEB) and adaptive cruise control (ACC), or the ego vehicle differential states, such as in electronic stability control (ESC), knowing the vehicle position with centimeter accuracy on a road with known geometry becomes important, particularly for AD [28].

Because of robustness and redundancy considerations, road-vehicle positioning is usually approached by fusion of multiple sensor modalities [26]. Several works perform vehicle localization using maps either determined a priori (e.g., using a mobile mapping system, MMS) or online jointly with the vehicle estimation. Using a predetermined map reduces the complexity of the positioning problem, but comes with drawbacks because maps from MMSs are updated infrequently, while higher-frequency changes to the map (e.g., road construction, lane repainting, temporary road rerouting) are not captured by an MMS-generated prior map. Hence, even a predetermined map needs to be updated regularly. For road estimation, many previous works model the road by clothoidal segments, sometimes in a spline setting (e.g., [22, 21, 31]). While this often works well, it has some limitations, as all road types are not designed according to the clothoidal assumption.

In this paper we present a framework for joint global vehicle positioning and map estimation where we represent the map using a spline representation, with implicit continuity by leveraging Bézier curves [18] and introducing generalized endpoints (GEPs). This results in a more expressive representation in terms of the types of roads the modeling framework can handle, as we are not restricted to the clothoidal assumption. As prior map information is typically given in the form of Cartesian or geodetic coordinates, we propose a systematic method to fit the prior map information to a collection of Bézier curves, where we assign Gaussian priors to the GEPs based on the uncertainty of the original map information. In addition, we fuse position measurements from a global navigation satellite system (GNSS) with a forward-looking camera, steering-wheel sensing, wheel-speed sensing, (optionally) an inertial measurement unit (IMU), and prior map information to update the map along with the vehicle state. In combination with a computer-vision (CV) algorithm,

Email addresses: karl.o.berntorp@ieee.org (Karl Berntorp), greiff@merl.com (Marcus Greiff)

[1] Corresponding author. *Phone*: +1 (617) 621-7583.

the camera provides measurements of the distance between the lane markings and the vehicle, in addition to measurements of the road geometry [20]. However, the quality of these measurements is time varying, for example, because of erroneous detection in the CV algorithm or because of other environmental effects, such as rain or lighting conditions that degrade the camera reliability. GNSS measurements provide global position information by estimating a receiver's (e.g., located in the vehicle) states from a set of code and carrier-phase measurements, acquired from one or several constellations of satellites and transmitted over one or more frequency bands [42, 35]. While being reliable most of the time, GNSS measurements are prone to occasional errors, which means that both camera and GNSS generate measurements with time-varying reliability.

The time-varying reliability of the measurements is hard to model and subsequently predict, because it depends on numerous external, partially unknown, factors. We account for the time-varying measurement reliability by letting the measurement-noise statistics be time varying. We pose the resulting nonlinear estimation problem in an adaptive Kalman-filter (KF) framework using linear-regression KFs (LRKFs), as the measurement model is not differentiable and therefore makes an extended KF impractical. We present two different approaches to handle time-varying sensor characteristics within the modeling framework. The first approach is based on the interactive multiple-model (IMM) methodology [14], where the noise variability is modeled by having a set of interacting LRKFs, each with a different measurement-noise covariance in a Gaussian-assumed noise distribution. The second approach is based on the variational-Bayes (VB) methodology [3], where the measurement covariance is explicitly estimated within a single LRKF. The methods come with their own set of benefits and drawbacks, and in a simulation study as well as in a hardware-in-the-loop (HIL) validation in an embedded implementation on a dSpace MicroAutobox-II we compare the two approaches and verify the real-time feasibility.

1.1. Relation to Previous Work

Joint vehicle state and road-map estimation has been researched using various sensor constellations and different estimation techniques. For instance, [21] fuses information from several (local) sensors to perform joint road geometry estimation and vehicle tracking. This work was extended in [31], where a forward looking camera and radar, together with an IMU, a steering wheel sensor, wheel speed sensors, and a new road-geometry model are leveraged in an extended KF. A similar work is [22], which in addition to the sensors in [31] develop a novel road model with claimed higher prediction accuracy compared to other established road models. However, none of these methods employ GNSS in the estimation formulation, nor do they perform global road-geometry estimation. Hence, they do not provide global positioning, which is important for several AD features, such as route planning and sophisticated motion-planning methods. Also, the cited prior work uses curvature-based road-map representations, whereas we employ a spline-based approach with implicit continuity enforcement. A curvature-based road representation is convenient because it needs few parameters, several textbook vehicle-control algorithms employ such representation [34], and nonurban roads can often be approximated well by a clothoidal representation [1]. However, a spline representation based on Bézier curves is more general, as we can also model roads in urban scenarios and quickly changing curvatures. The clothoidal approximation is violated in some situations, for example, when a part of the road is a straight line followed by a clothoidal stretch. In many situations (e.g., for highway or suburban driving) it is a good local approximation but it is unclear how to apply to global map estimation. Also, with our approach we can easily extract the estimated curvature from the estimated map should such quantity be sought, for example, for control purposes.

There are other vehicle state-estimation methods that rely on GNSS information. Three examples are: [6], which uses inertial sensors, wheel-speed sensors, and the steering-wheel angle sensor in combination with GNSS position measurements to perform vehicle-state estimation; [30], which performs tire radii estimation for improving vehicle odometry using GNSS measurements; and [43], which uses GNSS measurements in combination with camera, IMU, and range measurements. Both [6, 30] use GNSS to make related vehicle estimation problems observable and their focus is not on high-precision positioning, while [43] concerns multi-agent estimation.

Our prior work in this area includes [9], where we developed an IMM method for vehicle localization using GNSS measurements, wheel-speed sensors, a steering-wheel encoder, a prior map of the road, a camera that measures the distance to the left and right lanes, and optionally an IMU. That work did not include map estimation, as the map was assumed known. In [10] we relaxed this assumption, and introduced road-map estimation and a forward-looking camera into the problem formulation. In this article, compared to [9, 10], we generalize the framework and show that it is not limited to application of IMM-LRKFs. Specifically, compared to [10] we

- present an additional approach based on the VB methodology, which shows that our framework generalizes to different nonlinear filters, and we compare the two approaches in terms of estimation accuracy and computational demands;
- formalize the road-map modeling and its properties: we sketch the proof of the implicit continuity enforced by the introduction of GEPs, and we present a method that converts point-wise maps to our proposed map representation with continuous lane boundaries;

- provide a more extensive simulation study that includes comparisons between the two methods on a road segment corresponding to a real road; and
- we implement the proposed methods on an automotive-grade dSPACE MicroAutobox-II and throughly assess the computational demands.

1.2. Notation:

Throughout, $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicates that the vector $\boldsymbol{x} \in \mathbb{R}^{n_{\boldsymbol{x}}}$ is Gaussian distributed with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. We write

$$\boldsymbol{\Sigma} \sim \mathcal{IW}(\nu, \boldsymbol{V}) \propto |\boldsymbol{\Sigma}|^{-\frac{1}{2}(\nu+n_{\boldsymbol{y}}+1)} e^{\left(-\frac{1}{2}\mathrm{tr}(\boldsymbol{V}\boldsymbol{\Sigma}^{-1})\right)}$$

to imply that $\boldsymbol{\Sigma}$ is inverse-Wishart (IW) distributed with degree of freedom ν and scale matrix $\boldsymbol{V}$, where $\mathrm{tr}(\cdot)$ is the trace operator. The ith element of $\boldsymbol{x}$ is denoted with $[\boldsymbol{x}]_i$, matrices are written in capital bold font as $\boldsymbol{X}$, and the element on row i and column j of $\boldsymbol{X}$ is denoted with $[\boldsymbol{X}]_{ij}$. Similarly, the mth row of a matrix $\boldsymbol{X}$ is denoted with $[\boldsymbol{X}]_{m,:}$ We let $\hat{\boldsymbol{x}}_m$ denote the mean estimate of $\boldsymbol{x}$ at time step m given the measurement sequence $\boldsymbol{y}_{0:m} = \{\boldsymbol{y}_0, \dots \boldsymbol{y}_m\}$. With $p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k})$, we mean the posterior density function of the state $\boldsymbol{x}_k$ from time step 0 to time step k given $\boldsymbol{y}_{0:k}$ and $\hat{\boldsymbol{x}}_k^-$ is the one-step prediction of $\hat{\boldsymbol{x}}_{k-1}$. The concatenation of two vectors $\boldsymbol{x} \in \mathbb{R}^{n_{\boldsymbol{x}}}$ and $\boldsymbol{y} \in \mathbb{R}^{n_{\boldsymbol{y}}}$ is $[\boldsymbol{x}; \boldsymbol{y}] = [\boldsymbol{x}^\top, \boldsymbol{y}^\top]^\top \in \mathbb{R}^{n+m}$. Furthermore, $\boldsymbol{I}_n$ denotes the $n \times n$ identity matrix, $\boldsymbol{1}_n$ is a column vector of n elements equal to one, $(\boldsymbol{a})(\star)^\top = (\boldsymbol{a})(\boldsymbol{a})^\top$ for an expression $\boldsymbol{a}$, $\mathrm{vec}(\cdot)$ is the vectorization operator, and $\mathrm{blkdiag}(\boldsymbol{A}, \boldsymbol{B})$ denotes a block-diagonal matrix composed of $\boldsymbol{A}$ and $\boldsymbol{B}$. We let $f \in \mathcal{C}^n(A, B)$ denote functions $f : A \mapsto B$ whose first n derivatives are continuous. The notation $\boldsymbol{R}(\phi)$ means the 2D rotation matrix of angle ϕ. Finally, S^n denotes the n-sphere.

1.3. Outline:

Sec. 2 outlines the assumptions, the sensors, the dynamic models, the associated measurement models, and the spline-map representation. Sec. 3 presents the two proposed noise-adaptive LRKFs, with Sec. 3.2 detailing the IMM method and Sec. 3.3 outlining the VB method. In Sec. 4 we evaluate and compare these methods in terms of estimation accuracy, as well as assessing the real-time feasibility of the approaches using an automotive-grade dSpace MicroAutoBox-II rapid prototyping unit. Finally, Sec. 5 closes the paper.

2. Modeling

Fig. 1 shows the different coordinate frames used in this paper. The vehicle's coordinate frame O_E is located at the vehicle center of gravity. The vehicle yaw angle ψ describes the rotation of the vehicle frame O_E relative to the world frame O_W by the standard planar rotation

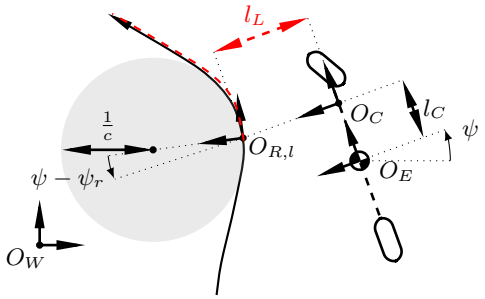

Figure 1: The relation between the vehicle frame O_E, the camera frame O_C, the road frame $O_{R,l}$, and the world frame O_W. The distance between the vehicle's longitudinal x-axis and the left lane boundary is $l_{R,l}$, and the shaded circle depicts the road curvature c (here exaggerated) at the origin of $O_{R,l}$. The lines in red dashed indicate measurements that can be obtained by the camera, which is located in O_C, for a given lookahead. The definition of $O_{R,r}$ is analogous to that of $O_{R,l}$.

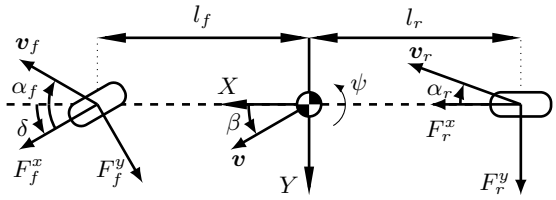

Figure 2: A schematic of the single-track model and related notation.

matrix $\boldsymbol{R}(\psi)$. The road-aligned frame $O_{R,l}$ is located on the left lane boundary, separated with a distance $l_{R,l}$ from the camera frame O_C, which is rigidly connected to O_E with distance l_C. The road-aligned frame $O_{R,r}$ is located on the right lane boundary, separated with a distance $l_{R,r}$ from the camera frame O_C.

2.1. Vehicle Models

We focus on normal driving maneuvers where it is appropriate to approximate the vehicle dynamics by a single-track (bicycle) model [16, 19, 34, 23]). For instance, as a rule of thumb, on dry asphalt the linear dynamic single-track model is valid for lateral accelerations up to $0.4g$, where g is the gravitational acceleration [4, 29, 39]. While a dynamic model based on force balances is generally more accurate than a kinematic model, the differences are small for regular driving [17]. Furthermore, a dynamic model depends on more parameters, such as the wheel radii, tire stiffness, and vehicle mass and inertia, which typically are unknown/uncertain and can be difficult to estimate in real time [7]. We introduce two assumptions that allows to model the vehicle dynamics by a single-track (i.e., bicycle) model.

Assumption 1 *The steering angles of the front left and right wheels are the same, here denoted by δ (see Fig. 2).*

Assumption 2 *The steering and acceleration commands are small, such that the vehicle operates in the linear region of the tire-force curve, with negligible inclination, roll, and road-bank angles.*

Remark 1 *Both of these assumptions can be relaxed but will cause higher complexity in the estimators to a limited benefit. While it is possible to incorporate more complex vehicle models than what we consider here, numerous papers have studied the validity of the approach we take in this paper, see [11, 7, 40, 17, 8, 12, 4] for a subset of such papers. For instance, [4] uses a dynamic linear single-track model applied to electronic stability control of an articulated bus and shows in a high-fidelity simulation that such simplified model is sufficient to represent the behavior of the substantially more complex dynamics of an articulated vehicle. The work [17] compares the kinematic and dynamic single-track model and concludes that a kinematic model is sufficient for normal driving (as considered in this paper), and [7, 12] show that a linear estimation model is sufficient for both estimation and control purposes in multiple real vehicle experiments, with scenarios resembling the ones considered in this paper.*

2.1.1. Dynamic Single-Track Model

In the following, F^x, F^y are the longitudinal and lateral tire forces, respectively, α is the wheel-slip angle, $\boldsymbol{v} = [v^X, v^Y]^\top$ is the velocity vector, and subscripts f, r denote front and rear, respectively. The state $\boldsymbol{x} \in^5$ consists of the Cartesian global vehicle position p^X, p^Y, the longitudinal and lateral velocity of the vehicle in O_E v^X, v^Ym , and the yaw rate $\dot{\psi}$. The equations of motion are

$$m(\dot{v}^X - v^Y\dot{\psi}) = F_f^x\cos(\delta) + F_r^x - F_f^y\sin(\delta), \tag{1a}$$
$$m(\dot{v}^Y + v^X\dot{\psi}) = F_f^y\cos(\delta) + F_r^y + F_f^x\sin(\delta), \tag{1b}$$
$$I\ddot{\psi} = l_f(F_f^y\cos(\delta) + F_f^x\sin(\delta)) - l_rF_r^y, \tag{1c}$$

where m is the vehicle mass and I is the inertia. From Assumption 2, the longitudinal and lateral tire forces can be approximated as linear functions of the wheel slip ratio κ and the slip angle α,

$$F_i^x \approx C_i^x\kappa_i, \quad F_i^y \approx C_i^y\alpha_i, \quad i = f, r, \tag{2}$$

where C_i^x, C_i^y are the longitudinal and lateral stiffness, respectively. The wheel slip is defined as in [39],

$$\kappa_i = \frac{v_i^x - R_w\omega_i}{\max(v_i^x, R_w\omega_i)}, \tag{3}$$

where ω_i is the wheel rotation rate, R_w is the effective wheel radius, and v_i^x is the wheel forward velocity in the wheel coordinate system. The slip angles are approximated as

$$\alpha_f \approx \delta - \frac{v_f^y + l_f\dot{\psi}}{v_f^x}, \quad \alpha_r \approx \frac{l_r\dot{\psi} - v_r^y}{v_r^x}. \tag{4}$$

To connect the global position with the velocity in O_E, let

$$\begin{bmatrix}\dot{p}^X & \dot{p}^Y\end{bmatrix}^\top = \boldsymbol{R}(\psi)\begin{bmatrix}v^X & v^Y\end{bmatrix}^\top, \tag{5}$$

The resulting model consisting of (1)–(5) is nonlinear, and after a discretization with a sampling period T_s, the dynamic single-track model can be written in shorthand as

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k^{\boldsymbol{x}}, \tag{6}$$

with input $\boldsymbol{u} = [\delta; \omega_f; \omega_r]$ and additive Gaussian zero-mean noise $\boldsymbol{w}_k^{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}^{\boldsymbol{x}})$, accounting for general model mismatches.

2.1.2. Kinematic Single-Track Model

While the kinematic single-track model is also based on the geometry in Fig. 2 but it needs fewer parameters, as it does not incorporate the mass, inertia, and friction parameters. The kinematic single-track model only uses the position andorientation as states, $\boldsymbol{z} = [p^X; p^Y; \psi] \in \mathbb{R}^3$, and therefore has lower complexity than the five-state dynamic single-track model (6). In this model, the wheel-speed measurements directly provide the vehicle velocity by $v^X = \frac{R_w}{2}(\omega_f + \omega_r)$ instead of affecting the motion through the slip κ_i and α_i. In continuous time, the model is

$$\dot{\boldsymbol{z}} = \begin{bmatrix} v^X\cos(\psi+\beta)/\cos(\beta) \\ v^X\sin(\psi+\beta)/\cos(\beta) \\ v^X\tan(\delta_f)/L \end{bmatrix}, \tag{7}$$

where $L = l_f + l_r$ and $\beta = \arctan(l_r\tan(\delta)/L)$ is the kinematic body-slip angle, which leads to

$$\boldsymbol{z}_{k+1} = \boldsymbol{g}(\boldsymbol{z}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k^{\boldsymbol{z}}, \tag{8}$$

after time discretization of (7), with Gaussian zero-mean process noise, $\boldsymbol{w}_k^{\boldsymbol{z}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}^{\boldsymbol{z}})$.

2.2. Road Model

We consider $M-1$ Bézier curves resulting in a spline $\mathcal{S}$ to represent the map [18]. In this section, we start by defining a spline-map representation of the road in Sec. 2.2.1. We then discuss how such representations can be projected to a space of maps with continuous lane boundaries in Sec. 2.2.2 and define an implicitly continuous map representation. Finally, in Sec. 2.2.3, we show how these maps can be generated from point-wise measurements of the road.

2.2.1. Linear Discontinuous Representation

We consider Bézier curves to build a distribution of the road map.

Definition 1 (Bézier curve) *A Bézier curve of degree n denoted by $\boldsymbol{b} : [0,1] \mapsto \mathbb{R}^d$ is defined by $n+1$ control points $\mathcal{P}_m = \{\boldsymbol{c}_{m,i} \in \mathbb{R}^d : d > 1, i \in [0, ..., n]\}$ as an interpolation*

$$\boldsymbol{b}(\lambda, \mathcal{P}_m) = \sum_{i=0}^{n} \underbrace{\binom{n}{i}(1-\lambda)^{(n-i)}\lambda^i}_{\zeta_{i,n}(\lambda)}\boldsymbol{c}_{m,i}, \tag{9}$$

where $\lambda \in [0,1]$. We build the maps using two such curves:

- One of degree $n = 3$, dimension $d = 2$, with points $\mathcal{P}_m = \{\boldsymbol{c}_{m,0}, \boldsymbol{c}_{m,1}, \boldsymbol{c}_{m,2}, \boldsymbol{c}_{m,3}\}$. This curve is denoted by $\boldsymbol{c}_m(\lambda) = \boldsymbol{b}(\lambda, \mathcal{P}_m)$ and represents the center lane;
- The other curve represents the half-width of the lane. This curve is denoted by $w_m(\lambda) = \boldsymbol{b}(\lambda, \mathcal{W}_m)$, is of degree $n = 1$, $d = 1$, and $\mathcal{W}_m = \{w_{m,0}, w_{m,1}\}$. If there are multiple lanes, the dimension d can be increased.

In the following, $\boldsymbol{r}_m = [\boldsymbol{c}_m; w_m] : [0, 1] \mapsto \mathbb{R}^2 \times \mathbb{R}_{>0}$ is a three-dimensional curve, and $\boldsymbol{r} = [\boldsymbol{c}; w] : [0, M-1] \mapsto \mathbb{R}^2 \times \mathbb{R}_{>0}$ denotes $M-1$ consecutive such curves such that

$$\boldsymbol{r}(s) = \begin{cases} \boldsymbol{r}_m(s - m + 1) & \text{if } s \in (0, M-1] \\ \boldsymbol{r}_1(0) & \text{if } s = 0 \end{cases}, \tag{10}$$

where $m = \lceil s \rceil$. We can express a normal direction as $\boldsymbol{n}(s) = \boldsymbol{R}(\pi/2)\boldsymbol{c}'(s)\|\boldsymbol{c}'(s)\|_2^{-1}$. The left and right lane boundaries are defined as $\boldsymbol{c}(s) \pm \boldsymbol{n}(s)w(s)$. The map parameters are

$$\bar{\boldsymbol{\gamma}} = [\mathrm{vec}(\mathcal{P}_1); ...; \mathrm{vec}(\mathcal{P}_{M-1}); \mathrm{vec}(\mathcal{W}_1); ...; \mathrm{vec}(\mathcal{W}_{M-1})]. \tag{11}$$

The problem with (11) is that the curves (10) need not be continuous at an integer s unless we impose constraints on $\bar{\boldsymbol{\gamma}}$. Furthermore, to achieve continuity of the lane boundaries, which is necessary for our proposed algorithms, we require at least $\boldsymbol{c} \in \mathcal{C}^1([0, M-1], \mathbb{R}^2)$ and $w \in \mathcal{C}^0([0, M-1], \mathbb{R}_+)$.

2.2.2. Nonlinear Implicitly Continuous Representation

While explicitly enforcing constraints on $\bar{\boldsymbol{\gamma}}$ can be done by numerical methods, we instead enforce continuity implicitly by defining a set of GEPs that by construction yields continuity. Specifically, consider a representation with M GEPs, $\{\boldsymbol{\gamma}_m\}_{m=1}^M$, which relate to the set of control points $\{(\mathcal{P}_m, \mathcal{W}_m)\}_{m=1}^{M-1}$ as

$$[\boldsymbol{\gamma}_m]_1 = x_m = [\boldsymbol{c}_{m,0}]_1 = [\boldsymbol{c}_{m-1,n}]_1, \tag{12a}$$
$$[\boldsymbol{\gamma}_m]_2 = y_m = [\boldsymbol{c}_{m,0}]_2 = [\boldsymbol{c}_{m-1,n}]_2, \tag{12b}$$
$$[\boldsymbol{\gamma}_m]_3 = \phi_m = \arctan(\tfrac{[\boldsymbol{c}_{m,0} - \boldsymbol{c}_{m-1,2}]_2}{[\boldsymbol{c}_{m,0} - \boldsymbol{c}_{m-1,2}]_1}), \tag{12c}$$
$$[\boldsymbol{\gamma}_m]_4 = r_m = \|\boldsymbol{c}_{m,1} - \boldsymbol{c}_{m-1,n-1}\|_2/2, \tag{12d}$$
$$[\boldsymbol{\gamma}_m]_5 = w_m = w_{m,0} = w_{m-1,1}, \tag{12e}$$

for all $m = 2, .., M-1$, with $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_M$ defined analogously.

Proposition 1 *Writing the segment $\boldsymbol{r}_m$ in $\boldsymbol{\gamma}_m$ and $\boldsymbol{\gamma}_{m+1}$ ensures that $\boldsymbol{c} \in \mathcal{C}^1([0, M-1], \mathbb{R}^2)$ and $w \in \mathcal{C}^0([0, M-1], \mathbb{R}_+)$.*

Proof 1 (Sketch) *Continuity of w is immediate due to* (12e)*. Continuity of the curve $\boldsymbol{c}$ similarly follows from* (12a) *and* (12b)*, where $\lim_{\lambda\to 1} \boldsymbol{c}_{m-1}(\lambda) = \boldsymbol{c}_{m-1,n} = \boldsymbol{c}_{m,0} = \lim_{\lambda\to 0} \boldsymbol{c}_m(\lambda)$. To show continuity of $\mathrm{d}\boldsymbol{c}/\mathrm{d}s$, we use that the derivative of* (9) *is a Bézier curve of degree $n-1$ where the first and last control points are given by $n^{-1}(\boldsymbol{c}_{m,1} - \boldsymbol{c}_{m,0})$ and $n^{-1}(\boldsymbol{c}_{m,n-1} - \boldsymbol{c}_{m,n})$, respectively. Hence, from* (12c) *and* (12d)*,*

$$\lim_{\lambda\to 0} \mathrm{d}\boldsymbol{c}_m(\lambda)/\mathrm{d}\lambda - \lim_{\lambda\to 1} \mathrm{d}\boldsymbol{c}_{m-1}(\lambda)/\mathrm{d}\lambda \tag{13a}$$
$$= n^{-1} \cdot (\boldsymbol{c}_{m,1} - \boldsymbol{c}_{m,0}) - n^{-1} \cdot (\boldsymbol{c}_{m-1,n-1} - \boldsymbol{c}_{m-1,n}) \tag{13b}$$
$$= n^{-1} \cdot (\boldsymbol{R}(\phi_m)[r_m; 0]) - n^{-1} \cdot (\boldsymbol{R}(\phi_m)[r_m; 0]) \tag{13c}$$
$$= \boldsymbol{0}. \tag{13d}$$

Hence, $\lim_{s\nearrow m} \mathrm{d}\boldsymbol{c}(s)/\mathrm{d}s = \lim_{s\searrow m} \mathrm{d}\boldsymbol{c}(s)/\mathrm{d}s$ for all $m = 2, ..., M-1$. □

We concisely write the GEPs defining the map, hereinafter referred to as the *map parameters*, as

$$\boldsymbol{\gamma} = [\boldsymbol{\gamma}_1; \boldsymbol{\gamma}_2; \cdots; \boldsymbol{\gamma}_M] \in \mathbb{M}^M = (\mathbb{R}^2 \times \mathsf{S}^1 \times \mathbb{R}_+^2)^M. \tag{14}$$

Next, we introduce uncertainty in the map by assigning a Gaussian prior on each $\boldsymbol{\gamma}_m$, $\boldsymbol{\gamma}_m \sim \mathcal{N}(\boldsymbol{\mu}_m^\gamma, \boldsymbol{\Sigma}_m^\gamma)$, that will be updated recursively as measurements are gathered. The time evolution of these parameters is hard to model from physical reasoning and an accurate prediction model is therefore difficult to obtain, as map segments are likely to stay constant for an extended period with occasional discrete changes (e.g., due to temporary road work). Therefore,

$$\boldsymbol{\gamma}_{k+1} = \boldsymbol{\gamma}_k + \boldsymbol{w}_k^\gamma, \quad \boldsymbol{w}_k^\gamma \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}^\gamma). \tag{15}$$

With the model (15), the process noise $\boldsymbol{Q}^\gamma$ determines how much uncertainty accumulates in the map as time increases. In summary, for a more compact notation, in the following we use $\bar{\boldsymbol{x}}$ to mean either $\bar{\boldsymbol{x}} = [\boldsymbol{x}; \boldsymbol{\gamma}]$ or $\bar{\boldsymbol{x}} = [\boldsymbol{z}; \boldsymbol{\gamma}]$. Similarly $\bar{\boldsymbol{f}}(\bar{\boldsymbol{x}}; \boldsymbol{u})$ denotes the functional relationships in (6) or (8) combined with (15), often omitting $\boldsymbol{u}$ for brevity. Also, we let the full process noise be denoted by $\bar{\boldsymbol{w}} \sim \mathcal{N}(\boldsymbol{0}, \bar{\boldsymbol{Q}})$.

Remark 2 *The function $F : \mathbb{M}^M \mapsto \bar{\mathbb{M}}^M$, that is, from $\boldsymbol{\gamma}$ to $\bar{\boldsymbol{\gamma}}$, is nonlinear but unique. The function $G : \bar{\mathbb{M}}^M \mapsto \mathbb{M}^M$, that is, from $\bar{\boldsymbol{\gamma}}$ to $\boldsymbol{\gamma}$, defined through* (12)*, is not generally invertible, but is a projection to the space of maps with continuous lane boundaries. If indeed a map in $\bar{\mathbb{M}}^M$ has continuous lane boundaries, $G = F^{-1}$.*

2.2.3. Regression with Continuity Constraints

Prior maps are usually not represented in a form that allows a direct conversion to the proposed spline representation with GEPs (12), and we therefore need a way to transform the prior maps. Here, we present a systematic way to go from point-wise maps (e.g., obtained by sampling points from a given prior map) to the proposed map representation that has continuous lane boundaries (c.f. Proposition 1).

We assume that the map is represented by J data points

$$\mathcal{D} = \{(\tilde{\boldsymbol{c}}_j, s_j) \in \mathbb{R}^2 \times [0, M-1], s_j - s_{j-1} = h\}_{j=1}^J \tag{16}$$

for some $h > 0$ and express the cost as the total variation (TV),

$$\begin{aligned}\mathrm{TV}(\bar{\boldsymbol{\gamma}}) &= \int \|\boldsymbol{c}(s) - \tilde{\boldsymbol{c}}(s)\|_2^2 \mathrm{d}s \propto h \sum_{m=1}^{M} \|[\boldsymbol{A}]_{m,:}\bar{\boldsymbol{\gamma}} - [\boldsymbol{a}]_m\|_2^2 \\ &= \|\boldsymbol{A}\bar{\boldsymbol{\gamma}} - \boldsymbol{a}\|_2^2. \end{aligned} \tag{17}$$

where $\boldsymbol{A}$ and $\boldsymbol{a}$ are formed in $\zeta_{i,n}(\lambda)$ defined in (9) and the data $\mathcal{D}$. Enforcing first-order continuity as in (13a) is a linear equality constraint in $\bar{\boldsymbol{\gamma}}$, $\boldsymbol{C}\bar{\boldsymbol{\gamma}} = \boldsymbol{0}$. Minimizing (17) is then a constrained least-squares problem, whose solution is given by the KKT system [15, Chap. 10.1.1]

$$\boldsymbol{M}_{\mathrm{KKT}} \begin{bmatrix} \bar{\boldsymbol{\gamma}} \\ \boldsymbol{\lambda} \end{bmatrix} \triangleq \begin{bmatrix} \boldsymbol{A}^\top \boldsymbol{A} & \boldsymbol{C}^\top \\ \boldsymbol{C} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{\gamma}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} 2\boldsymbol{a}^\top \boldsymbol{A} \\ \boldsymbol{0} \end{bmatrix}, \tag{18}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers.

Furthermore, if the sampled points $\tilde{\boldsymbol{c}}_k$ originate from a prior map that is unbiased with a Gaussian-distributed error, the errors of $\tilde{\boldsymbol{c}}_k$ are zero-mean Gaussian distributed, $\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$, and enter additively in $\boldsymbol{a}$. Hence, this error propagates to the optimal solution of (18) according to

$$\mathrm{Cov}\left(\begin{bmatrix} \bar{\boldsymbol{\gamma}} \\ \boldsymbol{\lambda} \end{bmatrix} \right) = \boldsymbol{M}_{\mathrm{KKT}}^{-1} \begin{bmatrix} 2\boldsymbol{A} \\ \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \sigma^2 \boldsymbol{I}_K & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \left(\boldsymbol{M}_{\mathrm{KKT}}^{-1} \begin{bmatrix} 2\boldsymbol{A} \\ \boldsymbol{0} \end{bmatrix} \right)^\top. \tag{19}$$

We can therefore represent the map parameters (11) with a Gaussian distribution, $\bar{\boldsymbol{\gamma}} \sim \mathcal{N}(\boldsymbol{\mu}^{\bar{\gamma}}, \boldsymbol{\Sigma}^{\bar{\gamma}})$, where $\boldsymbol{\mu}^{\bar{\gamma}}$ is given by the solution to (18) and $\boldsymbol{\Sigma}^{\bar{\gamma}}$ is the upper left block of the covariance matrix in (19). Finally, we approximate the distribution of the map parameters $\boldsymbol{\gamma}$ through G in Remark 2—that is, $\boldsymbol{\gamma} \sim \mathcal{N}(\boldsymbol{\mu}^{\gamma}, \boldsymbol{\Sigma}^{\gamma})$, from a Taylor expansion of G,

$$\boldsymbol{\mu}^{\gamma} = G(\boldsymbol{\mu}^{\bar{\gamma}}), \tag{20a}$$

$$\boldsymbol{\Sigma}^{\gamma} = \left(\frac{\partial G}{\partial \bar{\boldsymbol{\gamma}}}(\bar{\boldsymbol{\gamma}}) \right) \boldsymbol{\Sigma}^{\bar{\gamma}} \left(\frac{\partial G}{\partial \bar{\boldsymbol{\gamma}}}(\bar{\boldsymbol{\gamma}}) \right)^\top. \tag{20b}$$

Hence, we have a systematic way to go from the uncertainty of a prior map of any form using a set of points $\mathcal{D}$ to an approximate Gaussian distribution in the proposed nonlinear but implicitly continuous representation using GEPs (12). Algorithm 1 summarizes the steps to create the proposed spline-based map representation from a prior map.

Algorithm 1 Pseudo-code of the prior-map regression

Input: Prior map $\mathcal{M}$.

1: Sample $\mathcal{M}$ in a set of J points $\tilde{\boldsymbol{c}}_j$ to obtain (16).
2: Solve (18) and (19) to form $\bar{\boldsymbol{\gamma}} \sim \mathcal{N}(\boldsymbol{\mu}^{\bar{\gamma}}, \boldsymbol{\Sigma}^{\bar{\gamma}})$.
3: Evaluate (20) using (12) to obtain $\boldsymbol{\gamma} \sim \mathcal{N}(\boldsymbol{\mu}^{\gamma}, \boldsymbol{\Sigma}^{\gamma})$.

Output: Mean $\boldsymbol{\mu}^{\bar{\gamma}}$, covariance $\boldsymbol{\Sigma}^{\bar{\gamma}}$

Remark 3 *The ratio J/M of the number of map data points and number of segments depends on the resolution of the map in terms of data points J and the length of the segments $\boldsymbol{c}_m$ required to have sufficient accuracy in the spline-map representation. Irrespective of these factors, storing the map in the spline representation is usually efficient compared to storing the map data points. For example, as a qualitative argument, to generate the map used in the numerical validation, the center lane of the original map is stored in $J = 2000$ points (256kB of memory), whereas storage using $M - 1 = 40$ Bézier curves requires a magnitude less of memory (25.6kB).*

Remark 4 *If the map is prohibitively large, it is possible to leverage Remark 2 and solve several smaller optimization problems, before doing a projection $\bar{\boldsymbol{\gamma}} \leftarrow F(G(\bar{\boldsymbol{\gamma}}))$. This idea can also be leveraged to generate maps recursively, for example, when map points are provided recursively ahead of the vehicle, which necessitates map stitching.*

2.3. Measurement Model

We rely on GNSS position measurements $\boldsymbol{y}_k^p$ as the source of global information. These measurements are generated by an estimator using code and carrier-phase measurements, for example, using any of the methods in [13, 24, 25]. We assume the position measurements to be unbiased and Gaussian distributed. Because the estimation quality will continuously change with environmental conditions and receiver movements, both the mean $\boldsymbol{\mu}_k^p$ and covariance $\boldsymbol{\Sigma}_k^p$ are time varying, resulting in $\boldsymbol{y}_k^p \sim \mathcal{N}(\boldsymbol{\mu}_k^p, \boldsymbol{\Sigma}_k^p)$. For simplicity but without loss of generality, we let $\boldsymbol{y}^p \in \mathbb{R}^2$.

The camera in combination with a CV algorithm provides measurements of the road geometry and the vehicle position relative to the road. We assume intermediary processing such that the distance from O_C and the left/right lane boundaries, l_L, l_R, and a polynomial approximation of the lane markings, f_L, f_R, in front of the vehicle for a look-ahead defined by the CV algorithm, see Fig. 1. To use the polynomial approximation for inference, the measurement equation needs particular values at each time step. Hence, we sample the polynomials from the CV algorithm uniformly at n_s points over their domain defined in s in the camera frame O_C, $\{s_L^i, s_R^i\}_{i=1}^{n_s}$. This gives

$$\boldsymbol{h}^c = [l_L; l_R; f_L(s_L^1); \cdots; f_L(s_L^{n_s}); f_R(s_R^1); \cdots; f_R(s_R^{n_s})]. \tag{21}$$

The camera measurements $\boldsymbol{y}_k^c$ are assumed Gaussian distributed according to $\boldsymbol{y}_k^c \sim \mathcal{N}(\boldsymbol{h}^c(\bar{\boldsymbol{x}}_k), \boldsymbol{\Sigma}_k^c$.

When we use the dynamic single-track model in the estimation, the velocity and heading rate are also included in the state. Hence, we can utilize an IMU measuring the vehicle body frame longitudinal, $a_k^X = \dot{v}_k^X - v_k^Y \dot{\psi}_k$, and lateral, $a_k^Y = \dot{v}_k^Y + v_k^X \dot{\psi}$, acceleration and the yaw rate $\dot{\psi}_k$. The estimator uses the acceleration, a_k^X, a_k^Y, and yaw-rate $\dot{\psi}_k$ as measurements, forming the measurement vector $\boldsymbol{y}_k^a = [a_k^X, a_k^Y, \dot{\psi}_k]^\top$. Automotive-grade inertial sensors usually have a slowly time-varying bias, which should be modeled for any implementation supposed to run longer

than a few minutes [7, 8, 27]. For now, we assume that the bias has been predetermined offline but refer to some of our previous work [7, 8] for IMU bias estimation in automotive applications.

The complete measurement model is

$$\boldsymbol{y}_k = \boldsymbol{h}(\bar{\boldsymbol{x}}_k, \boldsymbol{u}_k) + \boldsymbol{e}_k \in \mathbb{R}^{n_{\boldsymbol{y}}}, \tag{22}$$

where $\boldsymbol{y}_k = [\boldsymbol{y}_k^p; \boldsymbol{y}_k^a; \boldsymbol{y}_k^c] \in \mathbb{R}^{7+2n_s}$ for the dynamic single-track model (6) and $\boldsymbol{y}_k = [\boldsymbol{y}_k^p; \boldsymbol{y}_k^c] \in \mathbb{R}^{2+2n_s}$ for the kinematic single-track model (8), and $\boldsymbol{e}_k$ is zero-mean Gaussian distributed with a block-diagonal covariance matrix $\boldsymbol{\Sigma}_k$.

Remark 5 *As the GNSS provides global position measurements of the vehicle and the camera provides map measurements relative to the vehicle, the models* (6) *or* (8) *in combination with* (22) *renders* $\bar{\boldsymbol{x}}_k$ *locally observable.*

3. Bayesian Sensor Fusion of GNSS and Camera

The resulting estimation problem involving estimating the vehicle state, the map, and the noise covariance is nonlinear and the Jacobian of the measurement equation is not known in closed form. For instance, given p^X, p^Y, ψ and $\bar{\boldsymbol{\gamma}}$, the distance l_L in (21) is found by applying a univariate Newton method to compute a path length $s_L^\star$ corresponding to the origin of $O_{R,l}$ in the global frame, before evaluating l_L. Thus, l_L is a function of $\bar{\boldsymbol{x}}$, but this function is not differentiable. Hence, we consider derivative-free nonlinear KFs. While an EKF may work by using numerical differentiation, it may be inaccurate. Also, due to the rich set of measurements, the estimation problem is likely unimodal and a particle filter (PF) is therefore not needed. A convenient middle-ground between complexity and accuracy is to use LRKFs, which we employ in this work embedded in two noise-adaptive formulations.

3.1. Linear-Regression Kalman Filter

LRKFs approximate the posterior density by its first two moments in a Gaussian approximation,

$$p(\bar{\boldsymbol{x}}_k|\boldsymbol{y}_{0:k}) \approx \mathcal{N}\big(\bar{\boldsymbol{x}}_k|\hat{\bar{\boldsymbol{x}}}_{k|k}, \boldsymbol{P}_{k|k}\big). \tag{23}$$

Given (23) at time step k, the distribution of the state prediction at time step $k+1$ is approximated by a Gaussian,

$$p(\bar{\boldsymbol{x}}_{k+1}|\bar{\boldsymbol{x}}_k, \boldsymbol{y}_{0:k}) \approx \mathcal{N}\left(\bar{\boldsymbol{x}}_{k+1}|\hat{\bar{\boldsymbol{x}}}_{k+1|k}, \boldsymbol{P}_{k+1|k}\right), \tag{24}$$

by direct evaluation of the associated moment integrals

$$\hat{\bar{\boldsymbol{x}}}_{k+1|k} = \int \bar{\boldsymbol{f}}(\bar{\boldsymbol{x}}_k) p(\bar{\boldsymbol{x}}_k|\boldsymbol{y}_{0:k})\, \mathrm{d}\bar{\boldsymbol{x}}_k, \tag{25a}$$

$$\boldsymbol{P}_{k+1|k} = \int \left(\bar{\boldsymbol{f}}(\bar{\boldsymbol{x}}_k) - \hat{\bar{\boldsymbol{x}}}_{k+1|k}\right)(\star)^\top p(\bar{\boldsymbol{x}}_k|\boldsymbol{y}_{0:k})\, \mathrm{d}\bar{\boldsymbol{x}}_k + \bar{\boldsymbol{Q}}_k, \tag{25b}$$

simplified by the assumption of additive noise $\bar{\boldsymbol{w}}_k$. Using the approximation (23), (25) reduces to two Gaussian integrals. For a general state-transition function $\bar{\boldsymbol{f}}$ no closed-form solutions exist, so we leverage numerical integration methods also known as *cubature rules* [41]. To this end, we transform the coordinates using the Cholesky factors of the covariance matrix $\boldsymbol{P}_{k|k} = \boldsymbol{L}_{k|k}\boldsymbol{L}_{k|k}^\top$, such that in the transformed coordinates, the integration is over a unit Gaussian. The LRKFs approximate the transformed integrals by evaluating the nonlinearity $\bar{\boldsymbol{f}}$ in a set of integration points defined by $\mathcal{I} = \{\varpi^i, \boldsymbol{\eta}^i\}_{i=1}^{|\mathcal{I}|}$, where $|\mathcal{I}|$ is the total number of points. Hence, for each such point $\boldsymbol{\eta}^i$,

$$\hat{\bar{\boldsymbol{x}}}_{k+1|k}^i = \bar{\boldsymbol{f}}\left(\hat{\bar{\boldsymbol{x}}}_{k|k} + \boldsymbol{L}_{k|k}\,\boldsymbol{\eta}^i\right), \tag{26}$$

and we approximate the moment integrals in (25) as

$$\hat{\bar{\boldsymbol{x}}}_{k+1|k} \approx \sum_{i=1}^{|\mathcal{I}|} \varpi^i \hat{\bar{\boldsymbol{x}}}_{k+1|k}^i, \tag{27a}$$

$$\boldsymbol{P}_{k+1|k} \approx \sum_{i=1}^{|\mathcal{I}|} \varpi^i (\hat{\bar{\boldsymbol{x}}}_{k+1|k}^i - \hat{\bar{\boldsymbol{x}}}_{k+1|k})(\star)^\top. \tag{27b}$$

For the measurement update, the joint density is approximated using the same integration techniques,

$$p\big([\bar{\boldsymbol{x}}_{k+1}; \boldsymbol{y}_{k+1}]|\bar{\boldsymbol{x}}_k, \boldsymbol{y}_{0:k}\big) \approx \mathcal{N}\left(\begin{bmatrix}\hat{\bar{\boldsymbol{x}}}_{k+1|k}\\ \hat{\boldsymbol{y}}_{k+1|k}\end{bmatrix}, \begin{bmatrix}\boldsymbol{P}_{k+1|k} & \boldsymbol{P}_{k+1|k}^{\bar{\boldsymbol{x}}\boldsymbol{y}}\\ \boldsymbol{P}_{k+1|k}^{\boldsymbol{y}\bar{\boldsymbol{x}}} & \boldsymbol{P}_{k+1|k}^{\boldsymbol{y}\boldsymbol{y}}\end{bmatrix}\right) \tag{28}$$

with moment integrals (dropping time indexing),

$$\hat{\boldsymbol{y}} = \int \boldsymbol{h}(\bar{\boldsymbol{x}})\mathcal{N}(\bar{\boldsymbol{x}}|\hat{\bar{\boldsymbol{x}}}, \boldsymbol{P})\mathrm{d}\bar{\boldsymbol{x}}, \tag{29a}$$

$$\boldsymbol{P}^{\bar{\boldsymbol{x}}\boldsymbol{y}} = \int (\hat{\bar{\boldsymbol{x}}} - \bar{\boldsymbol{x}})(\hat{\boldsymbol{y}} - \boldsymbol{h}(\bar{\boldsymbol{x}}))^\top \mathcal{N}(\bar{\boldsymbol{x}}|\hat{\bar{\boldsymbol{x}}}, \boldsymbol{P})\mathrm{d}\bar{\boldsymbol{x}}, \tag{29b}$$

$$\boldsymbol{P}^{\boldsymbol{y}\boldsymbol{y}} = \int (\hat{\boldsymbol{y}} - \boldsymbol{h}(\bar{\boldsymbol{x}}))(\star)^\top \mathcal{N}(\bar{\boldsymbol{x}}|\hat{\bar{\boldsymbol{x}}}, \boldsymbol{P})\mathrm{d}\bar{\boldsymbol{x}} + \boldsymbol{\Sigma}. \tag{29c}$$

Eq. (29) implies integrating over all of the GEPs $\boldsymbol{\gamma}$, which is computationally prohibitive when using automotive-grade ECUs with limited computation and memory bandwidth. To make the problem real-time feasible, we introduce Assumption 3.

Assumption 3 $\boldsymbol{\gamma}_i$ *and* $\boldsymbol{\gamma}_j$ *are independent for* $i \neq j$.

Using Assumption 3, we perform the integration with respect to each individual curve in the spline. When performing the moment evaluations, it is possible that the parameters of two adjacent curves are needed. At any rate, this leads to a significantly smaller estimation problem than considering the whole spline simultaneously. Conditioning of the joint density in (28) on the new measurement $\boldsymbol{y}_{k+1}$ amounts to the update

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1|k}^{\bar{\boldsymbol{x}}\boldsymbol{y}}(\boldsymbol{P}_{k+1|k}^{\boldsymbol{y}\boldsymbol{y}})^{-1}, \tag{30a}$$

$$\hat{\bar{\boldsymbol{x}}}_{k+1|k+1} = \hat{\bar{\boldsymbol{x}}}_{k+1|k} + \boldsymbol{K}_{k+1|k}(\boldsymbol{y}_{k+1} - \hat{\boldsymbol{y}}_{k+1|k}), \tag{30b}$$

$$\boldsymbol{P}_{k+1|k+1} = \boldsymbol{P}_{k+1|k} - \boldsymbol{K}_{k+1|k}\boldsymbol{P}_{k+1|k}^{\boldsymbol{y}\bar{\boldsymbol{x}}}, \tag{30c}$$

which is done with respect to the vehicle state and the currently relevant map parameters. Specifically, any GEP required in the process of evaluating (22) is included explicitly in the domain over which the integral is computed.

Remark 6 *Assumption 3 is introduced to make the estimation problem computationally tractable, as updating a full map representation of a potentially vast area at each time step is infeasible both from a memory and computation standpoint. However, the approximation errors introduced by Assumption 3 can be controlled by choosing the length of each curve.*

To account for the time-varying measurement reliability, we now present two different noise-adaptive methods. In Sec. 3.2 we detail the IMM-LRKF (see our prior work [10] for additional details) and this is followed by the VB-LRKF (see [37, 36]) in Sec. 3.3.

Remark 7 *While it is possible to use any cubature rule $\mathcal{I}$, we use the spherical Cubature rule [2] as it is simple and scales favorably when compared to Gauss-Hermite methods. For the current application, any performance difference between the different schemes is negligible.*

3.2. Interacting Multiple-Model LRKF

In the IMM-LRKF we have a set of B models that differ only in their measurement-noise characteristics. At each time step k, the IMM assigns a weight q_k to each model reflecting its probability of explaining the measurements. In this framework,

$$\begin{aligned}\bar{\boldsymbol{x}}_{k+1} &= \bar{\boldsymbol{f}}(\bar{\boldsymbol{x}}_k;\boldsymbol{u}_k)+\bar{\boldsymbol{w}}_k, \qquad \bar{\boldsymbol{w}}_k \sim \mathcal{N}(\boldsymbol{0},\bar{\boldsymbol{Q}}_k),\\ \boldsymbol{y}_k &= \boldsymbol{h}(\bar{\boldsymbol{x}}_k,\boldsymbol{u}_k)+\boldsymbol{e}_k(\theta_k),\ \ \boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{\Sigma}_k^{\theta_k}),\end{aligned}$$

where the mode parameter $\theta_k \in [1, B]$ evolves according to a finite-state Markov chain with transition probability matrix $\boldsymbol{\Pi} \in [0,1]^{B\times B}$, where all rows and columns sum to 1. For every possible θ_k, we assign a unique measurement-noise covariance matrix from

$$\{\boldsymbol{\Sigma}^{\theta_k} \in \mathbb{R}^{n_{\boldsymbol{e}}\times n_{\boldsymbol{e}}} | \boldsymbol{\Sigma}^{\theta_k} = (\boldsymbol{\Sigma}^{\theta_k})^\top, \boldsymbol{\Sigma}^{\theta_k} \succ \boldsymbol{0}\}_{\theta_k=1}^{B}.$$

At each time step, the IMM uses the transition matrix $\boldsymbol{\Pi}$ to perform a mixing of the B model estimates and weights,

$$\bar{q}_k^i = \sum_{j=1}^{B} \Pi_{ij} q_{k-1}^j, \tag{31a}$$

$$\hat{\bar{\boldsymbol{x}}}_{k-1}^i = \sum_{j=1}^{B} \Pi_{ij} \frac{q_{k-1}^j}{\bar{q}_k^i} \hat{\bar{\boldsymbol{x}}}_{k-1}^j, \tag{31b}$$

$$\boldsymbol{P}_{k-1}^i = \sum_{j=1}^{B} \Pi_{ij} \frac{q_{k-1}^j}{\bar{q}_k^i} \Big(\boldsymbol{P}_{k-1}^j + (\hat{\bar{\boldsymbol{x}}}_{k-1}^j - \hat{\bar{\boldsymbol{x}}}_{k-1}^i)(\star)^\top \Big). \tag{31c}$$

The IMM-LRKF executes B LRKFs to find the estimate of $\bar{\boldsymbol{x}}_k$, where the jth LRKF executes using the noise-covariance hypothesis $\boldsymbol{\Sigma}^j$. The state posterior is approximated as a Gaussian mixture of B components,

$$\begin{aligned} p(\bar{\boldsymbol{x}}_k|\boldsymbol{y}_{0:k}) &= \sum_{j=1}^{B} p(\bar{\boldsymbol{x}}_k^j|\boldsymbol{y}_{0:k}) = \sum_{j=1}^{B} \frac{p(\boldsymbol{y}_k|\bar{\boldsymbol{x}}_k^j)p(\bar{\boldsymbol{x}}_k|\boldsymbol{y}_{0:k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{0:k-1})} \\ &\approx \sum_{j=1}^{B} q_k^j \mathcal{N}(\bar{\boldsymbol{x}}_k^j|(\hat{\bar{\boldsymbol{x}}}_k^j)^-, (\boldsymbol{P}_k^j)^-), \end{aligned}$$

where

$$q_k^j \propto p(\boldsymbol{y}_k|\bar{\boldsymbol{x}}_k^j) = \mathcal{N}\Big(\boldsymbol{y}_k|((\hat{\boldsymbol{y}}_k^j)^-, (\boldsymbol{P}_k^{\boldsymbol{yy},j})^-\Big)\bar{q}_k^j, \ \ \forall j \in [1,B], \tag{32}$$

and $p(\boldsymbol{y}_k|\boldsymbol{y}_{0:k-1})$ is a normalization constant. The mean $(\hat{\boldsymbol{y}}_k^j)^-$ and covariance prediction $(\boldsymbol{P}_k^{\boldsymbol{yy},j})^-$ are determined by the corresponding LRKF. In this paper we determine the state estimate as the weighted average

$$\hat{\bar{\boldsymbol{x}}}_k = \sum_{j=1}^{B} q_k^j \hat{\bar{\boldsymbol{x}}}_k^j, \tag{33a}$$

$$\boldsymbol{P}_k = \sum_{j=1}^{B} q_k^j \Big(\boldsymbol{P}_k^j + (\hat{\bar{\boldsymbol{x}}}_k^j - \hat{\bar{\boldsymbol{x}}}_k)(\star)^\top \Big), \tag{33b}$$

which is the minimum mean-square estimate, but other estimates, such as the maximum aposteriori estimate, can be used. Algorithm 2 summarizes the proposed IMM-LRKF for adaptive sensor fusion of GNSS and camera measurements.

Algorithm 2 Pseudo-code of the proposed IMM-LRKF

Initialize: $\{\boldsymbol{\eta}^i, \varpi^i\}_{i=1}^{|\mathcal{I}|}, \{\hat{\bar{\boldsymbol{x}}}_{-1}^j, \boldsymbol{P}_{-1}^j, \boldsymbol{R}^j, q_{-1}^j\}_{j=1}^{B}, \boldsymbol{\Pi}$
for $k = 0, 1, \ldots$ **do**
  for $i \in \{1, \ldots, B\}$ **do**
    Mix estimates according to (31).
  end for
  for $j \in \{1, \ldots, B\}$ **do**
    for $i \in \{1, \ldots, |\mathcal{I}|\}$ **do**
      Determine $(\hat{\bar{\boldsymbol{x}}}_{k|}^i)^-$ according to (26).
    end for
    Determine $(\hat{\bar{\boldsymbol{x}}}_k^j)^-$, $(\boldsymbol{P}_k^j)^-$ according to (27).
    for $i \in \{1, \ldots, |\mathcal{I}|\}$ **do**
      Determine $(\hat{\boldsymbol{y}}_k^i)^-$ akin to (26)
    end for
    Determine $(\hat{\boldsymbol{y}}_k^j)^-$, $(\boldsymbol{P}_k^{\boldsymbol{yy},j})^-$, $(\boldsymbol{P}_k^{\bar{\boldsymbol{x}}\boldsymbol{y},j})^-$ akin to (27).
    Determine $\hat{\bar{\boldsymbol{x}}}_k^j$, $\boldsymbol{P}_k^j$ using (30).
    Determine q_k^j according to (32).
  end for
  Determine $\hat{\bar{\boldsymbol{x}}}_k$, $\boldsymbol{P}_k$ according to (33).
end for

3.3. Variational-Bayes LRKF

VB methods in Kalman filtering rely on a free-form approximation of the joint posterior distribution of $\bar{\boldsymbol{x}}_k$ and

$\boldsymbol{\Sigma}_k$,

$$p(\bar{\boldsymbol{x}}_k, \boldsymbol{\Sigma}_k | \boldsymbol{y}_{0:k}) \approx \underbrace{p(\bar{\boldsymbol{x}}_k | \boldsymbol{y}_{0:k})}_{Q_{\bar{\boldsymbol{x}}}(\bar{\boldsymbol{x}}_k)} \underbrace{p(\boldsymbol{\Sigma}_k | \boldsymbol{y}_{0:k})}_{Q_{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_k)}$$

where the objective is to determine $Q_{\bar{\boldsymbol{x}}}(\bar{\boldsymbol{x}}_k)$ and $Q_{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_k)$. The VB approximation minimizes the Kullback-Leibler (KL) divergence between an approximate distribution

$$Q_{\bar{\boldsymbol{x}}}(\bar{\boldsymbol{x}}_k) Q_{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_k)$$

and the true distribution $p(\bar{\boldsymbol{x}}_k, \boldsymbol{\Sigma}_k | \boldsymbol{y}_{0:k})$. This amounts to an optimization problem

$$\min_{Q_{\bar{\boldsymbol{x}}}, Q_{\boldsymbol{\Sigma}}} \mathrm{KL}(Q_{\bar{\boldsymbol{x}}}(\bar{\boldsymbol{x}}_k) Q_{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_k) || p(\bar{\boldsymbol{x}}_k, \boldsymbol{\Sigma}_k | \boldsymbol{y}_{0:k})) = \int Q_{\bar{\boldsymbol{x}}}(\bar{\boldsymbol{x}}_k) Q_{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_k) \log \left(\frac{Q_{\bar{\boldsymbol{x}}}(\bar{\boldsymbol{x}}_k) Q_{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_k)}{p(\bar{\boldsymbol{x}}_k, \boldsymbol{\Sigma}_k | \boldsymbol{y}_{0:k})} \right) \mathrm{d}\bar{\boldsymbol{x}}_k \mathrm{d}\boldsymbol{\Sigma}_k,$$

where the minimizers are [37, 36]

$$Q_{\bar{\boldsymbol{x}}}(\bar{\boldsymbol{x}}_k) \propto \exp\left(\int \log p(\boldsymbol{y}_k, \bar{\boldsymbol{x}}_k, \boldsymbol{\Sigma}_k | \boldsymbol{y}_{0:k-1}) Q_{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_k) \mathrm{d}\boldsymbol{\Sigma}_k \right), \tag{34a}$$

$$Q_{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_k) \propto \exp\left(\int \log p(\boldsymbol{y}_k, \bar{\boldsymbol{x}}_k, \boldsymbol{\Sigma}_k | \boldsymbol{y}_{0:k-1}) Q_{\bar{\boldsymbol{x}}}(\bar{\boldsymbol{x}}_k) \mathrm{d}\bar{\boldsymbol{x}}_k \right). \tag{34b}$$

Eq. (34) cannot be solved directly as there is coupling between $Q_{\bar{\boldsymbol{x}}}$ and $Q_{\boldsymbol{\Sigma}}$. However, when the state posterior is Gaussian assumed, $Q_{\bar{\boldsymbol{x}}}(\bar{\boldsymbol{x}}_k) = \mathcal{N}(\bar{\boldsymbol{x}}_k | \hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{P}_k)$ and the noise is IW distributed, $Q_{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_k) = \mathcal{IW}(\boldsymbol{\Sigma}_k | \nu_k, \boldsymbol{V}_k)$, the integrals in the exponentials of (34) can be written out explicitly. Subsequently, this leads to the coupled set of equations [36]

$$\begin{aligned}
\hat{\boldsymbol{y}}_k &= \int \boldsymbol{h}(\bar{\boldsymbol{x}}_k) \mathcal{N}(\bar{\boldsymbol{x}}_k | \hat{\bar{\boldsymbol{x}}}_k^-, \boldsymbol{P}_k^-) \mathrm{d}\bar{\boldsymbol{x}}_k, \\
\boldsymbol{P}_k^{\bar{\boldsymbol{x}}\boldsymbol{y}} &= \int (\hat{\bar{\boldsymbol{x}}}_k^- - \bar{\boldsymbol{x}}_k)(\hat{\boldsymbol{y}}_k - \boldsymbol{h}(\bar{\boldsymbol{x}}_k))^\top \mathcal{N}(\bar{\boldsymbol{x}}_k | \hat{\bar{\boldsymbol{x}}}_k^-, \boldsymbol{P}_k^-) \mathrm{d}\bar{\boldsymbol{x}}_k, \\
\boldsymbol{T}_k &= \int (\hat{\boldsymbol{y}}_k - \boldsymbol{h}(\bar{\boldsymbol{x}}_k))(\star)^\top \mathcal{N}(\bar{\boldsymbol{x}}_k | \hat{\bar{\boldsymbol{x}}}_k^-, \boldsymbol{P}_k^-) \mathrm{d}\bar{\boldsymbol{x}}_k, \\
\boldsymbol{S}_k &= \boldsymbol{T}_k + (\nu_k - n_{\boldsymbol{y}} - 1)^{-1} \boldsymbol{V}_k, \\
\boldsymbol{K}_k &= \boldsymbol{P}_k^{\bar{\boldsymbol{x}}\boldsymbol{y}} \boldsymbol{S}_k^{-1}, \\
\hat{\bar{\boldsymbol{x}}}_{k|k} &= \hat{\bar{\boldsymbol{x}}}_k^- + \boldsymbol{K}_k (\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k), \\
\boldsymbol{P}_{k|k} &= \boldsymbol{P}_k^- - \boldsymbol{K}_k \boldsymbol{S}_k \boldsymbol{K}_k^\top, \\
\nu_k &= \nu_k^- + 1, \\
\boldsymbol{V}_k &= \boldsymbol{V}_k^- + \int (\boldsymbol{y}_k - \boldsymbol{h}(\bar{\boldsymbol{x}}_k))(\star)^\top \mathcal{N}(\bar{\boldsymbol{x}}_k | \hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{P}_k) \mathrm{d}\bar{\boldsymbol{x}}_k,
\end{aligned} \tag{35}$$

where $\boldsymbol{\Sigma}_k = (\nu_k - n_{\boldsymbol{y}} - 1)^{-1} \boldsymbol{V}_k$. The first six equations in (35) are the usual KF equations and the integrals involved can be approximated as in Sec. 3.1. The solution to (35) is found by fixed-point iterations reminiscent to expectation-maximization (EM) methods and (locally) converge asymptotically under reasonable assumptions [32, 38].

Algorithm 3 Pseudo-code of the proposed VB-LRKF

Initialize: $\{\boldsymbol{\eta}^i, \varpi^i\}_{i=1}^{|\mathcal{I}|}, \hat{\bar{\boldsymbol{x}}}_{-1}, \boldsymbol{P}_{-1}, \nu_{-1}, \boldsymbol{V}_{-1}$

1: **for** $k = 0, 1, \ldots$ **do**
2: **for** $i \in \{1, \ldots, |\mathcal{I}|\}$ **do**
3: Determine $\hat{\bar{\boldsymbol{x}}}_{k|k-1}^i$ according to (26).
4: **end for**
5: Determine $\hat{\bar{\boldsymbol{x}}}_{k|k-1}$, $\boldsymbol{P}_{k|k-1}$ according to (27).
6: Set $\nu_k^- = \rho(\nu_{k-1} - n_{\boldsymbol{y}} - 1) + n_{\boldsymbol{y}} + 1$, $\boldsymbol{V}_k^- = \rho^2 \boldsymbol{I}_{n_{\boldsymbol{y}}} \boldsymbol{V}_{k-1}$.
7: **for** $i \in \{1, \ldots, |\mathcal{I}|\}$ **do**
8: Determine $\hat{\boldsymbol{y}}_{k|k-1}^i$ akin to (26)
9: **end for**
10: Determine $\hat{\boldsymbol{y}}_{k|k-1}$, $\boldsymbol{T}_k$, $\boldsymbol{P}_{k|k-1}^{\bar{\boldsymbol{x}}\boldsymbol{y}}$ akin to (27).
11: Set $\hat{\bar{\boldsymbol{x}}}_k^{(0)} = \hat{\bar{\boldsymbol{x}}}_{k|k-1}$, $\boldsymbol{P}_k^{(0)} = \boldsymbol{P}_{k|k-1}$, $\nu_k = 1 + \nu_k^-$, $\boldsymbol{V}_k^{(0)} = \boldsymbol{V}_k^-$, $j = 0$.
12: **while** `not_converged` **do**

$$\begin{aligned}
\boldsymbol{S}_k^{(j+1)} &= \boldsymbol{T}_k + (\nu_k - n_{\boldsymbol{y}} - 1)^{-1} \boldsymbol{V}_k^j \\
\boldsymbol{K}_k^{(j+1)} &= \boldsymbol{P}_k^{\bar{\boldsymbol{x}}\boldsymbol{y}} (\boldsymbol{S}_k^{(j+1)})^{-1} \\
\hat{\bar{\boldsymbol{x}}}_k^{(j+1)} &= \hat{\bar{\boldsymbol{x}}}_{k|k-1} + \boldsymbol{K}_k^{(j+1)} (\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1}) \\
\boldsymbol{P}_k^{(j+1)} &= \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k^{(j+1)} \boldsymbol{S}_k^{(j+1)} (\boldsymbol{K}_k^{(j+1)})^\top \\
\boldsymbol{V}_k^{(j+1)} &= \boldsymbol{V}_k^- + \int (\boldsymbol{y}_k - \boldsymbol{h}(\bar{\boldsymbol{x}}_k))(\star)^\top \\
&\quad \cdot \mathcal{N}(\bar{\boldsymbol{x}}_k | \hat{\bar{\boldsymbol{x}}}_k^{(j+1)}, \boldsymbol{P}_k^{(j+1)}) \mathrm{d}\bar{\boldsymbol{x}}_k
\end{aligned}$$

13: $j = j + 1$
14: **end while**
15: Set $\hat{\bar{\boldsymbol{x}}}_{k|k} = \hat{\bar{\boldsymbol{x}}}_k^{(j)}$, $\boldsymbol{P}_{k|k} = \boldsymbol{P}_k^{(j)}$, $\boldsymbol{V}_k = \boldsymbol{V}_k^{(j)}$.
16: **end for**

The prediction step of the sufficient statistics is in this paper chosen consistent with [36]:

$$\nu_k^- = \rho(\nu_{k-1} - n_{\boldsymbol{y}} - 1) + n_{\boldsymbol{y}} + 1, \tag{36}$$

$$\boldsymbol{V}_k^- = \rho \boldsymbol{V}_{k-1}, \tag{37}$$

where $\rho \in (0, 1]$ provides exponential forgetting.

Algorithm 3 summarizes the proposed VB-LRKF. The stopping criteria on Line 12 in Algorithm 3 can be designed in several ways; for example, by setting a maximum number of iterations or monitoring the change of the estimates between iterations according to some metric. In this paper, we terminate when the measurement error $\boldsymbol{e}^j = \boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^{(j)}$ between two consecutive iterations j and $j+1$ is smaller than some ϵ, that is, $\|\boldsymbol{e}_k^{(j+1)} - \boldsymbol{e}_k^{(j)}\| < \epsilon$. For the conducted evaluations, the method converges within $2 - 10$ iterations using $\epsilon \approx 10^{-3}$, depending on the road segment and driving behavior.

4. Simulation and Hardware-in-the-Loop Results

In this section we validate the proposed framework in a Monte-Carlo simulation study, and compare Algorithms 2 and 3 in terms of estimation accuracy. Further-

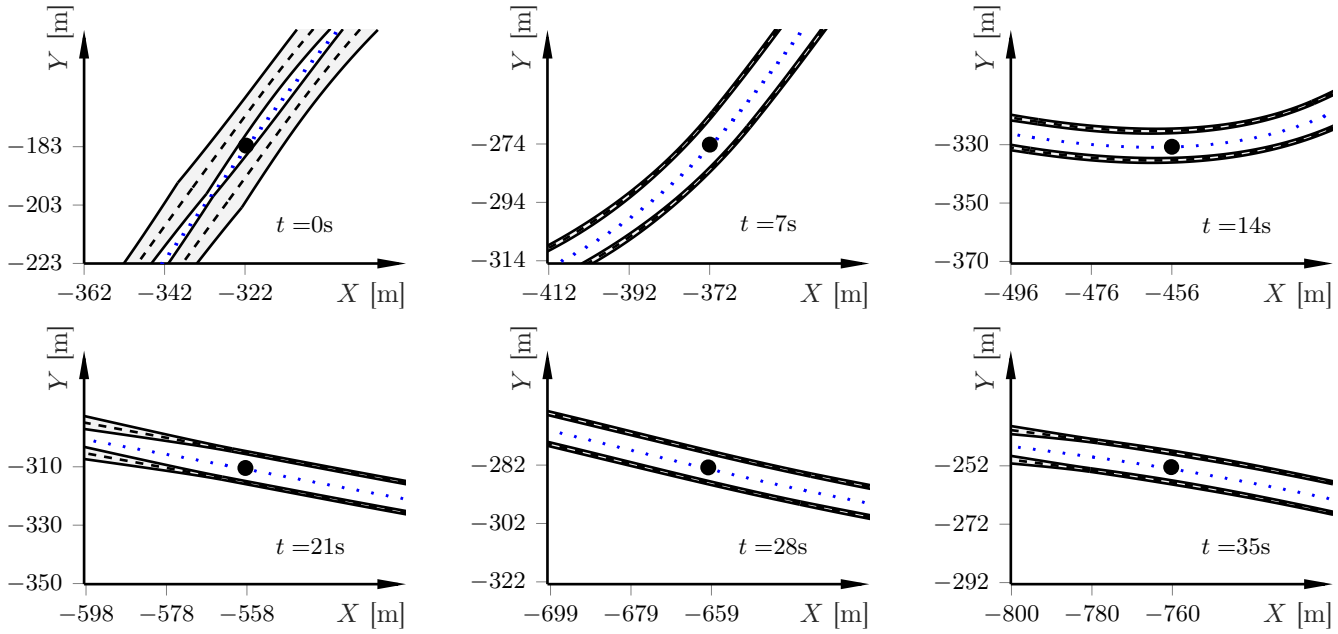

Figure 3: Snapshots of the estimated map for different time steps including the estimated vehicle position (black dot), estimated lane boundaries (black dashed), and the posterior with the map uncertainty illustrated by the gray areas. Note that the estimated uncertainty is magnified for visualization purposes. The blue dots indicate the middle of the road. The vehicle is initially moving in the south-west direction.

more, we validate the proposed methods for computational timing in a real-time computing environment. The computation times are obtained in dSPACE MicroAutoBox-II rapid prototyping unit equipped with a 900MHz PowerPC real-time processor (IBM PPC 750GL) and 16MB of RAM. The dSPACE MicroAutoBox-II rapid prototyping unit reflects the current and next-generation capabilities of embedded microcontrollers in automotive systems, and is a common platform for evaluating the real-time feasibility of automotive algorithms.

4.1. Preliminaries

We validate the proposed method in a Monte-Carlo simulation study using a set of 100 Monte-Carlo runs, each 40s long. For generating synthetic data, the vehicle is modeled by a dynamic single-track model in closed loop with a reference tracking controller driving on a one-lane road [10]. The route is extracted using the open-source routing machine (OSRM) tool [33] from a road segment in the vicinity of Boston, MA, and the map is represented by a sequence of points, corresponding to the output of an MMS. To generate our spline-based map, we run Algorithm 1 using 20 Bézier curves. We regress two maps: one with little noise ($\sigma = 1$cm), which is the ground-truth map, and one with greater noise ($\sigma = 10$cm), which is a prior for the road-map estimation.

The GNSS position measurements are nominally Gaussian zero-mean with covariance $\mathbf{\Sigma}^p_{\text{nom}} = \text{diag}(0.2^2 \boldsymbol{I}_2)$. Furthermore, the camera provides lane measurements that nominally are Gaussian distributed, $\boldsymbol{y}^c_k \sim \mathcal{N}(\boldsymbol{h}^c(\bar{\boldsymbol{x}}_k), \mathbf{\Sigma}^c)$, where $\mathbf{\Sigma}^c_{\text{nom}} = \text{diag}(0.14^2 \boldsymbol{I}_{10})$. The GNSS covariance has been chosen to resemble the reliability of high-performing GNSS estimation modules under nominal conditions, and the camera covariance has been determined by a qualitative analysis of typical camera-based lane-estimation modules. Note, however, that because the lane measurements are sampled from a polynomial that is fitted to the lane markings, in general the measurements, even without adding noise, will not fit perfectly to the road. Hence, we can expect the noise estimates to be slightly larger than the added random noise. The IMU measurement noise is comparable to the noise for a low-cost IMU. Note that only the filters using the dynamic single-track model leverage the IMU.

For each Monte-Carlo run, we generate the data using different noise realizations, we sample the initial state from a Gaussian distribution with 1m initial standard deviation on the position, and we sample new map realizations *used by the estimators* from the mean and covariance outputted by Algorithm 1 using $\sigma = 10$cm. All measurements arrive with sampling rate 10Hz but the prediction step is performed at 100Hz, that is, when executing Algorithms 2 and 3 at 100Hz, the measurement update step and weight update (IMM only) are executed every tenth time step. The reason for the mixed rates between the prediction and measurement step is that GNSS and camera measurements usually arrive at a rate that is lower than the internal vehicle data from the CAN bus. To generate the measurements, we consider three different models:

1. $\mathbf{\Sigma}^p = \mathbf{\Sigma}^p_{\text{nom}}, \mathbf{\Sigma}^c = \mathbf{\Sigma}^c_{\text{nom}}$;
2. $\mathbf{\Sigma}^p = 10^2\mathbf{\Sigma}^p_{\text{nom}}, \mathbf{\Sigma}^c = \mathbf{\Sigma}^c_{\text{nom}}$;
3. $\mathbf{\Sigma}^p = \mathbf{\Sigma}^p_{\text{nom}}, \mathbf{\Sigma}^c = 10^2\mathbf{\Sigma}^c_{\text{nom}}$.

The first model models the reliability of the measurements in nominal conditions, the second model mimics the case when GNSS measurements are unreliable, and the third model includes unreliable camera measurements. The in-

Table 1: Vehicle model parameters used in (1)–(7) for the results presented in this study.

Notation	Value	Unit
l_f	1.432	m
l_r	1.472	m
R_w	0.333	m
C_f^x	96000	N
C_r^x	120000	N
C_f^y	96000	N/rad
C_r^y	120000	N/rad
m	2200	kg
g	9.81	m/s^2

flation by a factor of 10 is chosen to give enough difference from the nominal measurement noise while not entirely discarding them in the estimation. GNSS outliers occur every tenth second starting at 5s that last three seconds. Similarly, we have camera outliers occurring every tenth second that last three seconds, starting at 10s.

In the IMM, the transition probability matrix $\mathbf{\Pi}$ is set to have diagonal elements around 0.9 with identical off-diagonal elements. From our experience, the design of the mixing matrix is not critical to performance, although we acknowledge that better performance can likely be obtained by more careful tuning. In IMM-LRKF, we use the six noise covariance matrices

1. $\mathbf{\Sigma}^p = \mathbf{\Sigma}^p_{\text{nom}}, \mathbf{\Sigma}^c = \mathbf{\Sigma}^c_{\text{nom}}$;
2. $\mathbf{\Sigma}^p = 10^2\mathbf{\Sigma}^p_{\text{nom}}, \mathbf{\Sigma}^c = 2^2\mathbf{\Sigma}^c_{\text{nom}}$;
3. $\mathbf{\Sigma}^p = \mathbf{\Sigma}^p_{\text{nom}}, \mathbf{\Sigma}^c = 10^2\mathbf{\Sigma}^c_{\text{nom}}$;
4. $\mathbf{\Sigma}^p = \mathbf{\Sigma}^p_{\text{nom}}, \mathbf{\Sigma}^c = 5^2\mathbf{\Sigma}^c_{\text{nom}}$;
5. $\mathbf{\Sigma}^p = 10^2\mathbf{\Sigma}^p_{\text{nom}}, \mathbf{\Sigma}^c = \mathbf{\Sigma}^c_{\text{nom}}$;
6. $\mathbf{\Sigma}^p = \mathbf{\Sigma}^p_{\text{nom}}, \mathbf{\Sigma}^c = 2^2\mathbf{\Sigma}^c_{\text{nom}}$.

That is, IMM-LRKF includes the noise models that generate the synthetic data in addition to variations of the camera covariance matrix to account for that the polynomial fitting to the road is imperfect.

The vehicle parameters are from a mid-sized SUV, and the tire parameters correspond to driving on a dry asphalt road [5]. Table 1 provides the vehicle and tire parameters.

4.1.1. Illustrative Simulation

For illustration purposes, we first present results from a single realization using IMM-LRKF (Algorithm 2).

Fig. 3 displays snapshots of excerpts of the estimated map using IMM-LRKF at different time instants, with the vehicle initially moving in the negative Y-direction. As measurements are gathered, the map estimate is improved, and the uncertainty in the map is decreased. Note the decrease of the uncertainty ahead of the vehicle due to the forward-looking camera. For instance, at $t = 0$s, the uncertainty is large due to the initialized prior from Algorithm 1,

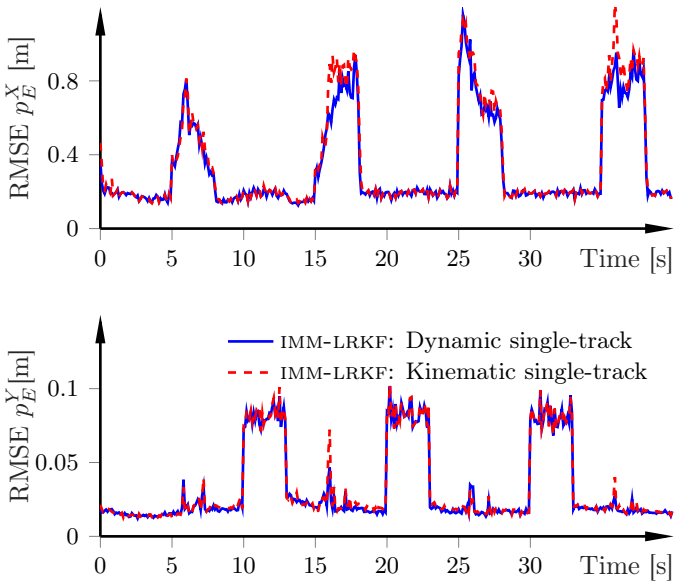

Figure 4: Position RMSE of IMM-LRKF (in the vehicle frame) over 100 Monte-Carlo runs with 0.1m initial map standard deviation (see Remark 2). The improvement of using the dynamic single-track model compared to the kinematic-single track model is minor.

but at later time steps the decrease in uncertainty is noticeable. While we only visualize the covariances using IMM-LRKF, VB-LRKF qualitatively yields the same results.

4.1.2. Results from a Monte-Carlo Study

For the Monte-Carlo study, we have performed 100 Monte-Carlo runs of the same scenario as in Sec. 4.1.1 for different forgetting factors in VB-LRKF, using both the dynamic and kinematic single-track model, and using Algorithms 2 and 3 both with and without map updates.

Fig. 4 displays the position RMSE in the vehicle frame for IMM-LRKF when using the dynamic and kinematic single-track model, respectively. Comparing the performance when using different vehicle models, there is a small improvement when using the dynamic single-track model, but it is minor, except for the longitudinal error when there are GNSS outliers (see, e.g., around $t = 17$s). This is expected as when there are GNSS outliers, the longitudinal positioning error is largely dependent on the prediction model. These conclusions also hold for VB-LRKF, and they indicate that for the driving behavior considered in this example, positioning accuracy is sufficient using the kinematic single-track model. Hence, in the remainder of this paper we only consider the kinematic single-track model in the estimator, with the dynamic single-track model only used to generate the synthetic data.

Fig. 5 shows the position RMSEs, and Fig. 6 shows the corresponding noise estimates averaged over the 100 Monte-Carlo runs using Algorithm 2 (IMM-LRKF) and Algorithm 3 (VB-LRKF) for different forgetting factors *with* map updates, using the kinematic single-track model. Overall, IMM-LRKF performs slightly better than VB-LRKF, irrespective of forgetting factor. This is expected, as IMM-LRKF includes the correct noise models for both inlier and outlier measurements. IMM-LRKF has near instantenous

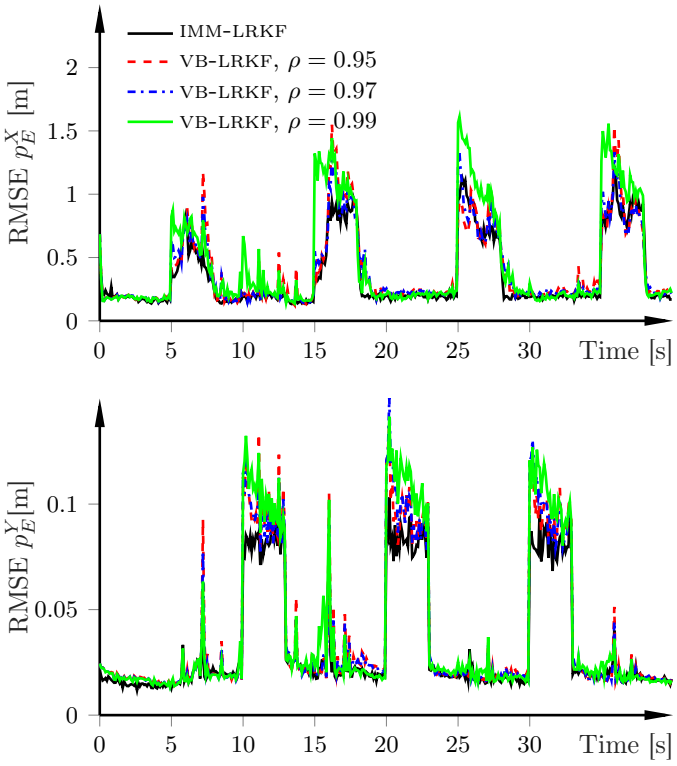

Figure 5: Position RMSE (in the vehicle frame) for IMM-LRKF and VB-LRKF with different forgetting factors over 100 Monte-Carlo runs with 0.1m initial map standard deviation (see Remark 2), using the kinematic single-track model in the estimators.

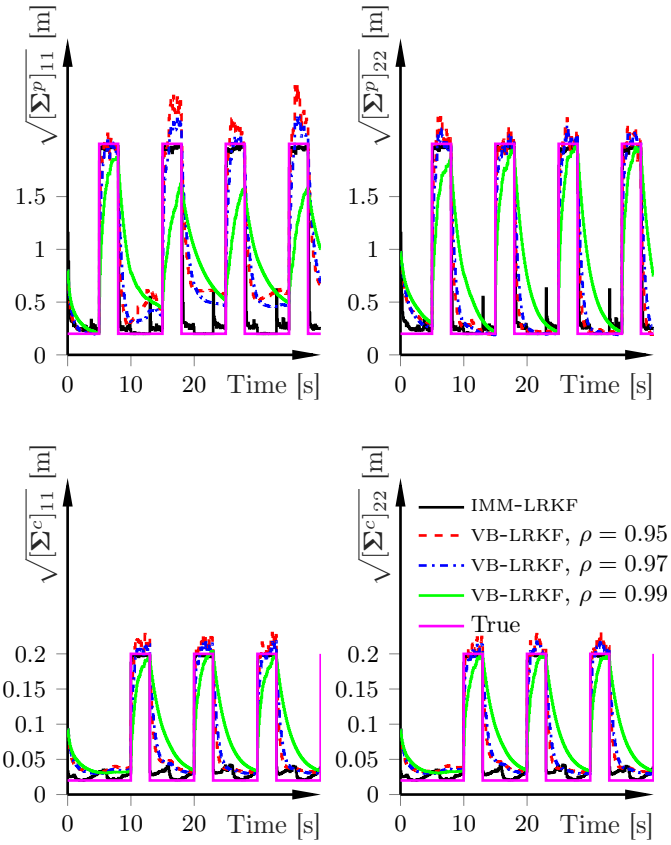

Figure 6: Noise estimates for the GNSS position measurements and measured distances to each road boundary, averaged over the 100 Monte-Carlo runs. The true noise levels in magenta and using the kinematic single-track model in the estimators.

convergence in the noise estimates. While this is partially reflected in the positioning errors, the difference between IMM-LRKF and VB-LRKF with $\rho = 0.95$ and $\rho = 0.97$ is mostly negligible except for a few instances; for inlier measurements, the steady-state error differences are negligible. Using VB-LRKF with a high forgetting factor $\rho = 0.99$ generally gives too slow convergence in the noise estimates, which is reflected in the positioning errors.

4.1.3. Comparison With and Without Map Updates

To conclude the evaluation of the positioning performance, we evaluate the impact of including map updates. We do this in terms of the RMSE of the lateral position in the vehicle frame, as the accuracy of the lateral positioning is largely determined by the camera measurements and the accuracy of the map estimates. In contrast, the longitudinal positioning is largely determined by the GNSS measurements, which are not dependent on the map accuracy.

Fig. 7 shows the results with and without map updates for VB-LRKF using $\rho = 0.97$ and for IMM-LRKF. First, IMM-LRKF with map updates gives the smallest position RMSE, especially when outliers are present. Second, IMM-LRKF without map updates in contrast gives the largest posi-

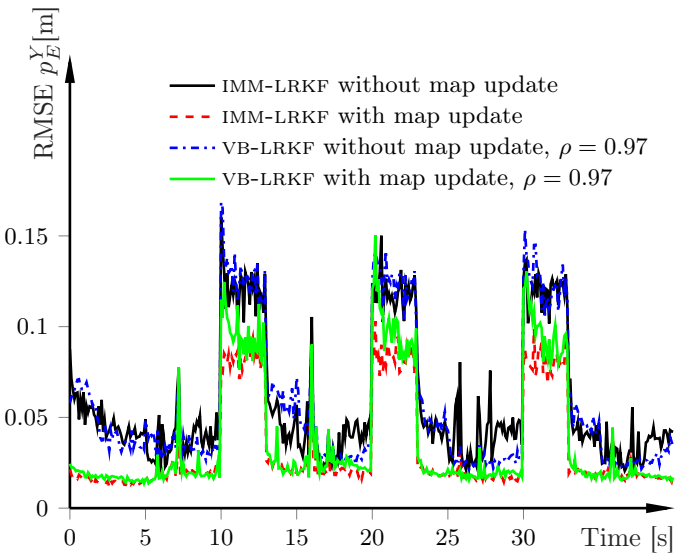

Figure 7: Position RMSE in the vehicle frame with and without map update (Algorithms 2 and 3 without map states) for an initial prior map standard deviation of 0.1m (see Remark 2).

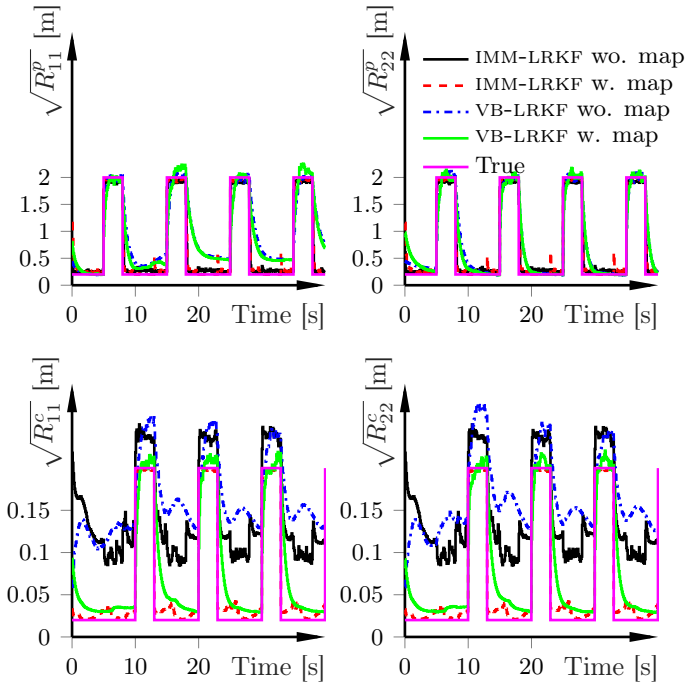

Figure 8: Average noise estimates for the GNSS position measurements and measured distances to each road boundary with map update and without map update (Algorithms 2 and 3 without map states) for an initial map standard deviation of 0.1m (see Remark 2). Ground truth in magenta. VB-LRKF uses $\rho = 0.97$.

tion RMSE, particularly for inlier measurements. Third, VB-LRKF with map updates performs close to IMM-LRKF, but with a slightly larger RMSE. Comparing VB-LRKF and IMM-LRKF without map updates, VB-LRKF gives slightly smaller errors and the estimation is smoother overall.

Fig. 8 displays the corresponding noise estimates. To account for the unmodeled errors in the map, when executing VB-LRKF (Algorithm 3) without map updates, the camera noise estimates are inflated. While this is intuitive since the unmodeled map uncertainty is injected into the noise estimates, it leads to degraded performance compared to including map updates. This is even more pronounced for IMM-LRKF. With a finer gridding of the noise models in IMM-LRKF (i.e., using more models), this behavior can be mitigated. Still, it indicates some of the defincies with using IMM-LRKF when using an incorrect map.

Finally, to visualize the variability of estimation performance between the different configurations, Fig. 9 displays the probability of the time-averaged lateral position RMSE for the filters in Fig. 7. The two approaches with map updates perform similarly with small variability. The difference in performance between the filters with and without map updates is notable.

4.2. Real-Time Feasibility for Embedded Implementation

To conclude the validation, we assess the real-time computational feasibility of the proposed methods in a dSPACE MicroAutoBox-II rapid prototyping unit. We utilized an implementation in Simulink using the MATLAB function block with automatic C-code generation, and the setup is

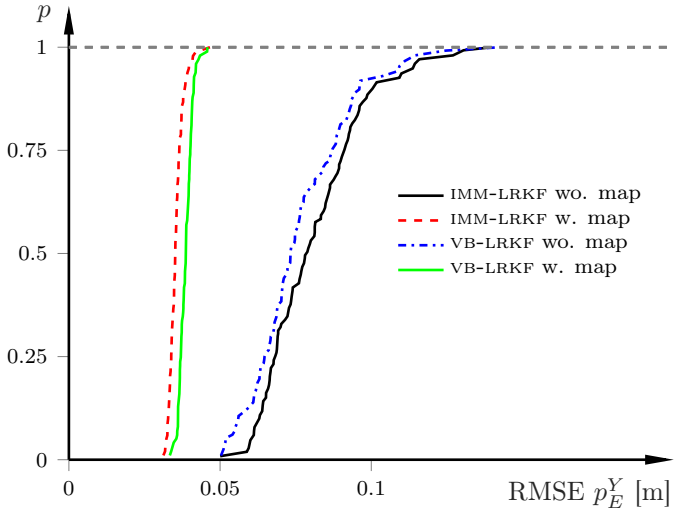

Figure 9: Probability of time-averaged position RMSE (in the vehicle frame) for IMM-LRKF and VB-LRKF with and without map updates, computed as cumulative distribution functions over the 100 Monte-Carlo runs of the whole data set. The simulations correspond to Figs. 7 and 8. VB-LRKF uses $\rho = 0.97$.

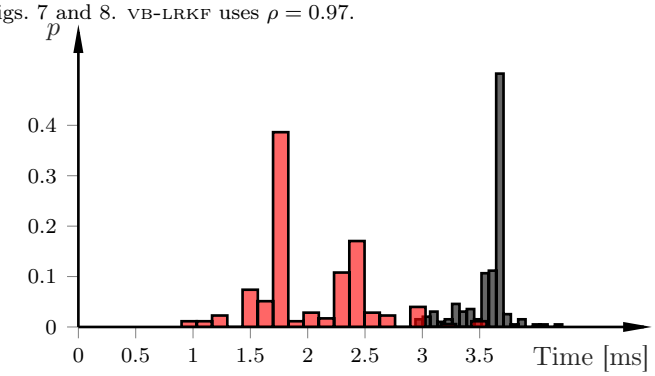

Figure 10: Histogram of the computation times using IMM-LRKF (gray) and VB-LRKF (red) without map updates (i.e., vehicle state estimation alone).

otherwise the same as in the preceding section. The HIL setup is analogous to [40, 12].

Fig. 10 displays the distribution of the computation times per time step over a 20s simulation for IMM-LRKF (gray) and VB-LRKF (red), respectively, without map updates. The computation times include both a prediction step and an update step (i.e., one full iteration of Algorithm 2 and 3). The computational budget is 10ms, as the method updates with 100Hz. Both methods are well within the allotted computational budget. However, on average, VB-LRKF is roughly a factor of 2 faster than IMM-LRKF.

Fig. 11 shows barplots of the corresponding results when incorporating map updates. To generate the results, we varied the maximum number of iterations allowed in VB-LRKF. Furthermore, we reduced the sampling rate in both estimators to 10Hz. We see that all filter variants stay within the maximum allotted computation time of 100ms. IMM-LRKF is more predictable in that it exhibits a lower variation of computation times. Nevertheless, all versions of VB-LRKF have a substantially smaller average computation time than IMM-LRKF, about a factor of 3-5 on average. Note that all versions of VB-LRKF have simi-

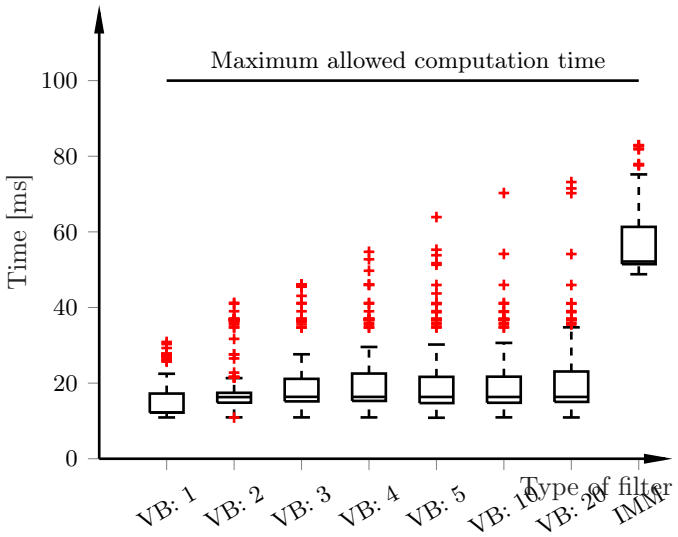

Figure 11: Barplots of the computation times with map updates using IMM-LRKF with six models and VB-LRKF for a varying number of maximum number of iterations.

lar mean and 75% quantile computation times, while the maximum computation time differs more. This implies that a vast amount of the fixed-point iterations converge quickly, while there are a few occasions when convergence requires more computations. However, from our experience the main steps impacting estimation performance are the first $3-4$ iterations. Hence, to improve the worst-case computation times, it is possible to set the maximum iterations to a relative small number, without sacrificing estimation performance significantly.

5. Conclusions

We presented a framework for joint vehicle positioning and map estimation based on GEPs. The map parametrization gives a flexible way to model road maps and does not exhibit some of the limitations that clothoidal modeling frameworks have. Coupled with our two noise-adaptive nonlinear KFs, the framework allows real-time execution on automotive-grade computing power. The validation on a dSPACE MicroAutoBox-II indicates that the presented methods are suitable for implementation on automotive embedded platforms. Furthermore, our study shows that for the considered assumptions and maneuvers, a kinematic single-track model gives only minor performance reductions than a dynamic counterpart, while avoiding the need to estimate additional parameters.

From experience, capping the maximum allowed number of iterations low (say, 3–4 iterations) does not impact estimation performance in VB-LRKF but has a large impact in terms of meeting the computation budget. While IMM-LRKF has higher estimation accuracy potential, it has the inherent drawback that reasonable guesses on the noise models need to be known a priori. In this regard, VB-LRKF is more flexible.

References

[1] , 1994. Vägutformning 94 version S-2. Technical Report. Swedish National Road Administration.

[2] Arasaratnam, I., 2009. Cubature Kalman filtering theory & applications. Ph.D. thesis. McMaster University.

[3] Beal, M., 2003. Variational algorithms for approximate Bayesian inference. Ph.D. thesis. University of London.

[4] Berntorp, K., 2008. ESP for Suppression of Jackknifing in an Articulated Bus. Master's Thesis ISRN LUTFD2/TFRT--5831--SE. Department of Automatic Control, Lund University, Sweden.

[5] Berntorp, K., 2014. Particle Filtering and Optimal Control for Vehicles and Robots. Ph.D. thesis. Department of Automatic Control, Lund University, Sweden.

[6] Berntorp, K., 2016. Joint wheel-slip and vehicle-motion estimation based on inertial, GPS, and wheel-speed sensors. IEEE Trans. Control Syst. Technol. 24, 1020–1027.

[7] Berntorp, K., Di Cairano, S., 2018. Tire-stiffness and vehicle-state estimation based on noise-adaptive particle filtering. IEEE Trans. Control Syst. Technol. 27, 1100–1114.

[8] Berntorp, K., Di Cairano, S., 2019. Noise-statistics learning of automotive-grade sensors using adaptive marginalized particle filtering. J. Dynamic Syst., Measurement, and Control 141, 061009–10.

[9] Berntorp, K., Greiff, M., Di Cairano, S., 2022. Bayesian sensor fusion of GNSS and camera with outlier adaptation for vehicle positioning, in: Int. Conf. Information Fusion, Linköping, Sweden.

[10] Berntorp, K., Greiff, M., Di Cairano, S., Miraldo, P., 2023. Bayesian sensor fusion for joint vehicle localization and road mapping using onboard sensors, in: Int. Conf. Information Fusion, Charleston, SC.

[11] Berntorp, K., Hoang, T., Quirynen, R., Di Cairano, S., 2018. Control architecture design of autonomous vehicles, in: Conf. Control Technol. and Applications, Copenhagen, Denmark. Invited paper.

[12] Berntorp, K., Quirynen, R., Uno, T., Di Cairano, S., 2019. Trajectory tracking for autonomous vehicles on varying road surfaces by friction-adaptive nonlinear model predictive control. Veh. Syst. Dyn. 58, 705–725.

[13] Berntorp, K., Weiss, A., Di Cairano, S., 2020. Integer ambiguity resolution by mixture Kalman filter for improved GNSS precision. IEEE Trans. Aerosp. Electron. Syst. 56, 3170–3181.

[14] Blom, H., Bar-Shalom, Y., 1988. The interacting multiple model algorithm for systems with Markovian switching coefficients. IEEE Trans. Autom. Control .

[15] Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press.

[16] Carvalho, A., Lefévre, S., Schildbach, G., Kong, J., Borrelli, F., 2015a. Automated driving: The role of forecasts and uncertainty—A control perspective. European J. Control 24, 14–32.

[17] Carvalho, A., Lefévre, S., Schildbach, G., Kong, J., Borrelli, F., 2015b. Automated driving: The role of forecasts and uncertainty - a control perspective. Eur. J. Control 24, 14–32.

[18] De Boor, C., 2001. A practical guide to splines: Revised Version. volume 27. Springer-Verlag, New York.

[19] Di Cairano, S., Tseng, H., Bernardini, D., Bemporad, A., 2013. Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain. IEEE Trans. Control Syst. Technol. 21, 1236–1248.

[20] Dickmanns, E.D., 2007. Dynamic Vision for Perception and Control of Motion. Springer-Verlag, Berlin, Heidelberg.

[21] Eidehall, A., Pohl, J., Gustafsson, F., 2007. Joint road geometry estimation and vehicle tracking. Control Eng. Pract. 15, 1484–1494.

[22] Garcia-Fernandez, A.F., Hammarstrand, L., Fatemi, M., Svensson, L., 2014. Bayesian road estimation using onboard sensors. IEEE Trans. Intell. Transp. Syst. 15, 1676–1689.

[23] Gillespie, T., 1992. Fundamentals of vehicle dynamics. Society of Automotive Engineers, Inc.

[24] Greiff, M., Berntorp, K., 2020. Optimal measurement projections with adaptive mixture Kalman filtering for GNSS positioning, in: Amer. Control Conf.
[25] Greiff, M., Berntorp, K., Di Cairano, S., Kim, K., 2021. Mixed-integer linear regression Kalman filters for GNSS positioning, in: Conf. Control Techn. Applications, San Diego, CA.
[26] Gustafsson, F., 2009. Automotive safety systems. IEEE Signal Process. Mag. 26, 32–47.
[27] Gustafsson, F., 2010. Statistical Sensor Fusion. Utbildningshuset/Studentlitteratur, Lund, Sweden.
[28] Karlsson, R., Gustafsson, F., 2017. The future of automotive localization algorithms: Available, reliable, and scalable localization: Anywhere and anytime. IEEE Signal. Process. Mag. 34, 60–69.
[29] Kiencke, U., Nielsen, L., 2005. Automotive Control Systems—For Engine, Driveline and Vehicle. 2nd edition ed., Springer-Verlag, Berlin Heidelberg.
[30] Lundquist, C., Karlsson, R., Özkan, E., Gustafsson, F., 2014. Tire radii estimation using a marginalized particle filter. IEEE Transactions on Intelligent Transportation Systems 15, 663–672.
[31] Lundquist, C., Schön, T.B., 2011. Joint ego-motion and road geometry estimation. Information Fusion 12, 253–263.
[32] Mbalawata, I.S., Särkkä, S., Vihola, M., Haario, H., 2015. Adaptive Metropolis algorithm using variational Bayesian adaptive Kalman filter. Computational Statistics & Data Analysis 83, 101–115.
[33] OSRM, . Homepage of the open source road map project. URL: http://project-osrm.org/. last accessed 08-19-2021.
[34] Rajamani, R., 2006. Vehicle Dynamics and Control. Springer-Verlag.
[35] Sahmoudi, M., Landry, R., 2009. A nonlinear filtering approach for robust multi-GNSS RTK positioning in presence of multipath and ionospheric delays. IEEE J. Selected Topics Signal Process. 3, 764–776.
[36] Särkkä, S., Hartikainen, J., 2013. Non-linear noise adaptive Kalman filtering via variational Bayes, in: IEEE Int. Workshop Machine Learning for Signal Processing, Southampton, UK.
[37] Särkkä, S., Nummenmaa, A., 2009. Recursive noise adaptive Kalman filtering by variational Bayesian approximations. IEEE Trans. Automat. Contr. 54, 596–600.
[38] Sato, M.A., 2001. Online model selection based on the variational Bayes. Neural computation 13, 1649–1681.
[39] Schindler, E., 2007. Fahrdynamik: Grundlagen Des Lenkverhaltens Und Ihre Anwendung Für Fahrzeugregelsysteme. Expert-Verlag, Renningen, Germany.
[40] Sean Vaskov, Rien Quirynen, M.M., Berntorp, K., 2024. Friction-adaptive stochastic nonlinear model predictive control for autonomous vehicles. Vehicle System Dynamics 62, 347–371. doi:10.1080/00423114.2023.2219791.
[41] Steinbring, J., Hanebeck, U.D., 2013. SSKF: The smart sampling Kalman filter, in: Int. Conf. Information Fusion, Istanbul, Turkey.
[42] Teunissen, P., 1997. A canonical theory for short GPS baselines. Part III: the geometry of the ambiguity search space. J. Geodesy 71, 486–501.
[43] Wen, W., Bai, X., Zhang, G., Chen, S., Yuan, F., Hsu, L.T., 2020. Multi-agent collaborative GNSS/camera/INS integration aided by inter-ranging for vehicular navigation in urban areas. IEEE Access .