2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Variational Bayes Kalman Filter for Joint Vehicle Localization and Road Mapping Using Onboard Sensors

Karl Berntorp, Marcus Greiff

Abstract— **This paper addresses the joint vehicle-state and road-map estimation based on global navigation satellite system (GNSS), camera, steering-wheel sensing, wheel-speed sensors, and a prior map. Because prior maps, e.g., generated from mobile mapping systems, are updated infrequently and do not capture high-frequency events such as road construction, we include the map parameters in the estimation problem. 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 combine the sensor information in a noise-adaptive variational Bayes Kalman filter to jointly estimate the vehicle state, the parameter vector of the map, and the measurement noise. Simulation results indicate that the method can accurately adjust the measurement noise to the environmental conditions and thereby correct for errors in the prior map while providing accurate vehicle positioning.**

I. Introduction

Road-vehicle positioning is usually approached by fusion of multiple sensor modalities [1]. Also, by leveraging prior road maps, for example, generated by a mobile mapping system (MMS), reliability and accuracy can be improved as it enables positioning relative to a global map. However, relying blindly on prior maps 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. To this end, many approaches for vehicle positioning include map updating in the estimation problem. Recent works use a combination of global navigation satellite system (GNSS), camera, inertial measurement units (IMUs), and radar to accurately and jointly estimate the vehicle state and road map. For instance, [2] fuses information from several (local) sensors to perform joint road geometry estimation and vehicle tracking. This work was extended in [3], 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 Kalman filter (EKF). An issue with most sensor setups is that the reliability of camera and GNSS measurements is time-varying—for example, because of erroneous detection in the computer-vision algorithm or because of 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 [4]. 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.

In [5], we developed a GNSS-based sensor-fusion method for joint vehicle positioning and road-map estimation, with the map represented by at least one spline. The sensor-fusion method estimates the vehicle state and the spline coefficients in an interacting multiple model (IMM) linear-regression Kalman filter (LRKF), outputting the mean estimate and associated covariance of said quantities. The IMM-LRKF in [5] can deliver accurate vehicle positioning and map updates, down to centimeter-level accuracy, provided that the true underlying measurement noise is captured in the set of LRKFs. However, it can be difficult to know a priori what is a suitable set of noise models to include in the IMM. One option is to include a set large enough to surely include any combination of noise values, but this can lead to the IMM being computationally prohibitive. To solve this problem, this paper adopts a variational-Bayes (VB) methodology [6] for joint vehicle state and road-map estimation. We adapt the VB method in [7] to the LRKF setting and estimate explicitly the measurement covariance within a single LRKF, using our previously developed spline-based road representation. We show that the method can accurately estimate the time-varying noise and thereby provide a more flexible solution than previously presented in [5].

A. 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 $\boldsymbol{\Sigma}$ and $\mathcal{IW}(\nu, \boldsymbol{\Sigma})$ is the inverse-Wishart distribution with degree of freedom ν and scale matrix $\boldsymbol{\Sigma}$. 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 X_{ij}. We let $\hat{\boldsymbol{x}}_{j|m}$ denote the estimate of $\boldsymbol{x}$ at time step j 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 $\boldsymbol{y}_{0:k}$. 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, $\mathbf{1}_{n\times n}$ denotes the $n \times n$ identity matrix, $\mathbf{1}_n$ 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 $\boldsymbol{B}$. The notation $\mathcal{R}(\phi)$ denotes the 2D rotation matrix of angle ϕ.

Mitsubishi Electric Research Labs (MERL), 02139 Cambridge, MA, USA. Email: karl.o.berntorp@ieee.org

Copyright ©2024 EUCA

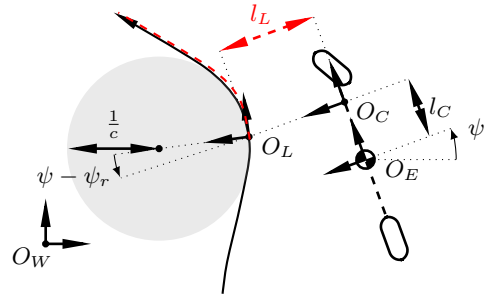

Fig. 1. The relation between the vehicle frame O_E, the camera frame O_C, the road frame of the left lane, $O_{R,l}$, and the world frame O_W. The distance between the vehicle's longitudinal X-axis and the left lane boundary is l_L, and the shaded circle depicts the road curvature (here exaggerated) at the origin of O_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 $O_{R,l}$.

II. Vehicle and Sensor 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 matrix. The road-aligned frame O_L is located on the left lane boundary, separated with a distance l_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 from the camera frame O_C.

In a previous paper, we showed that for estimation purposes under normal driving conditions (i.e., not at-the-limit maneuvers), a kinematic single-track model performs similar to a dynamic equivalent [8]. Hence, in the following we consider a kinematic model, but our method straightforwardly extends to more complicated vehicle models. The kinematic single-track model is based on a bicycle model, with its three states being the global (planar position) and the heading angle, $\boldsymbol{z} = [p^X, p^Y, \psi] \in \mathbb{R}^{n_{\boldsymbol{z}}}$, $n_{\boldsymbol{z}} = 3$. The wheel-speed measurements directly provide the vehicle velocity. In continuous time, the model is

$$\dot{\boldsymbol{z}} = \begin{bmatrix} v^X \cos(\psi + \beta)/\cos(\beta) \\ v^X \sin(\psi + \beta)/\cos(\beta) \\ v^X \tan(\delta_f)/L \end{bmatrix}, \tag{1}$$

where $L = l_f + l_r$, $\beta = \arctan(l_r \tan(\delta)/L)$ is the kinematic body-slip angle, and the velocity is related to the wheel speeds by $v^X = \frac{R_w}{2}(\omega_f + \omega_r)$. After time discretization, we write (1) concisely as

$$\boldsymbol{z}_{k+1} = \boldsymbol{g}(\boldsymbol{z}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k^{\boldsymbol{z}}, \tag{2}$$

with Gaussian zero-mean process noise, $\boldsymbol{w}_k^{\boldsymbol{z}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}^{\boldsymbol{z}})$, to account for generic model errors, and $\boldsymbol{u}_k$ denotes the control input at time step k.

A. Road Model

There are multiple ways a road map can be represented. Several previous works have employed a clothoidal representation [3], [9]. However, as highlighted in [5], it is appealing to consider lower-dimensional Bézier curves forming a spline and implicitly enforce continuity at the endpoints of the curves [5, Proposition 1], as Bézier curves are more expressive than clothoids. Consequently, we consider Bézier curves for the construction of probabilistic road-map distributions.

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*

$$\boldsymbol{b}(\lambda, \mathcal{P}_m) = \sum_{i=0}^{n} \binom{n}{i} (1-\lambda)^{(n-i)} \lambda^i \boldsymbol{c}_{m,i} \tag{3}$$

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,

$$\boldsymbol{r}(s) = \begin{cases} \boldsymbol{r}_m(s - m + 1) & \text{if } s \in (0, M-1] \\ \boldsymbol{r}_1(0) & \text{if } s = 0 \end{cases}, \tag{4}$$

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

$$\bar{\gamma} = [\mathrm{vec}(\mathcal{P}_1); ...; \mathrm{vec}(\mathcal{P}_{M-1}); \mathrm{vec}(\mathcal{W}_1); ...; \mathrm{vec}(\mathcal{W}_{M-1})]. \tag{5}$$

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

To this end, we consider a representation with M generalized endpoints (GEPs), denoted by $\{\boldsymbol{\gamma}_m\}_{m=1}^{M}$, which relate to the set of control points $\{(\mathcal{P}_m, \mathcal{W}_m)\}_{m=1}^{M-1}$ as

$$[\boldsymbol{\gamma}_m]_1 = x_m = [\boldsymbol{c}_{m,0}]_1 = [\boldsymbol{c}_{m-1,n}]_1, \tag{6a}$$
$$[\boldsymbol{\gamma}_m]_2 = y_m = [\boldsymbol{c}_{m,0}]_2 = [\boldsymbol{c}_{m-1,n}]_2, \tag{6b}$$
$$[\boldsymbol{\gamma}_m]_3 = \phi_m = \arctan\left(\frac{[\boldsymbol{c}_{m,0} - \boldsymbol{c}_{m-1,2}]_2}{[\boldsymbol{c}_{m,0} - \boldsymbol{c}_{m-1,2}]_1}\right), \tag{6c}$$
$$[\boldsymbol{\gamma}_m]_4 = r_m = \|\boldsymbol{c}_{m,1} - \boldsymbol{c}_{m-1,n-1}\|_2/2, \tag{6d}$$
$$[\boldsymbol{\gamma}_m]_5 = w_m = w_{m,0} = w_{m-1,1}, \tag{6e}$$

for all $m = 2, .., M-1$, with $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_M$ defined analogously. Expressing the segment $\boldsymbol{r}_m$ in $(\boldsymbol{\gamma}_m, \boldsymbol{\gamma}_{m+1})$ ensures that $\boldsymbol{c} \in \mathcal{C}^1([0, M-1], \mathbb{R}^2)$ and $w \in \mathcal{C}^0([0, M-1], \mathbb{R}_{>0})$, see [5].

Remark 1 *As the center lane $\boldsymbol{c}(s)$ is linear in the map parameters* (5)*, it is possible to formulate constrained linear-regression problems that fit a road-map representation to a collection of data points $\mathcal{D} = \{(\boldsymbol{c}_j, s_j)\}_{j=1}^{J}$ assuming a measurement model $\boldsymbol{c}_j \sim \mathcal{N}(\boldsymbol{c}(s_j), \sigma^2 \boldsymbol{I}_2)$. Such points can*

be sampled from high-definition maps, or simulators. Hence, we map a solution of the regression problem to the GEP-representation by (6) *and use it as a filtering prior.*

In the following, we denote the map parameters in a GEP representation by $\boldsymbol{\gamma} = [\gamma_1; ...; \gamma_M]$ and introduce uncertainty in the map by assigning a Gaussian prior on each $\boldsymbol{\gamma}_m$, $\boldsymbol{\gamma}_{m,0} \sim \mathcal{N}(\boldsymbol{m}^{\gamma}_{m,0}, \boldsymbol{\Sigma}^{\gamma}_{m,0})$ as per Remark 1. The time evolution of $\boldsymbol{\gamma}$ is hard to model from physical reasoning, as road maps are mostly constant over long time spans but change abruptly during road maintenance. Assuming slow changes in the map, we use a nearly-constant position model,

$$\boldsymbol{\gamma}_{k+1} = \boldsymbol{\gamma}_k + \boldsymbol{w}^{\gamma}_k, \quad \boldsymbol{w}^{\gamma}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}^{\gamma}). \tag{7}$$

As an example, if the process noise in the map is small, $\boldsymbol{Q}^{\gamma} \approx \boldsymbol{0}$, there are negligible changes to the map state $\boldsymbol{\gamma}$ in the time update and most changes are due to the measurement update. In summary, the prediction model consisting of (1) and (7) is

$$\boldsymbol{x}_{k+1} = \underbrace{[\boldsymbol{g}(\boldsymbol{z}_k, \boldsymbol{u}_k); \boldsymbol{\gamma}_k]}_{\triangleq \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)} + \boldsymbol{w}_k, \quad \boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}), \tag{8}$$

with $\boldsymbol{x} = [\boldsymbol{z}; \boldsymbol{\gamma}]$, $\boldsymbol{w} = [\boldsymbol{w}^{\boldsymbol{z}}; \boldsymbol{w}^{\boldsymbol{\gamma}}]$ and $\boldsymbol{Q} = \mathrm{blkdiag}(\boldsymbol{Q}^{\boldsymbol{z}}, \boldsymbol{Q}^{\boldsymbol{\gamma}})$.

B. Measurement Model

We consider the GNSS position measurements $\boldsymbol{y}^p_k$ generated by an estimator using code and carrier-phase measurements, for example, by the methods in [10]–[12]. 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 $\boldsymbol{\mu}^p_k$ and covariance $\boldsymbol{R}^p_k$ are time varying, resulting in $\boldsymbol{y}^p_k \sim \mathcal{N}(\boldsymbol{\mu}^p_k, \boldsymbol{R}^p_k)$. For simplicity but without loss of generality, $\boldsymbol{y}^p \in \mathbb{R}^2$.

The camera in combination with a computer-vision (CV) algorithm (e.g., [13]) provides measurements of the road geometry and the relative vehicle position. We assume intermediary processing such that we obtain 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.

To use the polynomial approximation for inference, the measurement equation needs particular values at each time step and not functional expressions. Hence, we sample the polynomials from the CV algorithm uniformly at n_s points over their domain defined in s, $\{s^i_L, s^i_R\}_{i=1}^{n_s}$. This gives

$$\boldsymbol{h}^c = [l_L, l_R, f_L(s^1_l), \cdots, f_L(s^{n_s}_L), f_R(s^1_R), \cdots, f_R(s^{n_s}_R)]^{\top}. \tag{9}$$

The camera measurements $\boldsymbol{y}^c_k$ are assumed Gaussian distributed according to $\boldsymbol{y}^c_k \sim \mathcal{N}(\boldsymbol{\mu}^c_k, \boldsymbol{R}^c_k)$, where, similarly to the GNSS measurements, both the mean and covariance are time varying. The complete measurement model is

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{e}_k \in \mathbb{R}^{n_{\boldsymbol{y}}}, \tag{10}$$

where $\boldsymbol{y}_k = [\boldsymbol{y}^p_k; \boldsymbol{y}^c_k] \in \mathbb{R}^{4+2n_s}$ and $\boldsymbol{e}_k$ is zero-mean Gaussian distributed with a block-diagonal covariance matrix.

As noted in [5], because the GNSS provides global position measurements of the vehicle and the camera provides map measurements relative to the vehicle, the model (8) in combination with (10) renders $\boldsymbol{x}_k$ locally weakly observable. Note that the Jacobians of the measurement equation are not known in closed form, but have to be numerically approximated if used.[1]

III. Bayesian Sensor Fusion of GNSS and Camera

The 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, we consider derivative-free nonlinear KFs.

A. Linear-Regression Kalman Filter

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

$$p(\boldsymbol{x}_k | \boldsymbol{y}_{0:k}) \approx \mathcal{N}\big(\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}\big). \tag{11}$$

Given the assumed Gaussian filtering posterior (11) at time step k, the distribution of the state prediction at time step $k+1$ is approximated by another Gaussian,

$$p(\boldsymbol{x}_{k+1} | \boldsymbol{x}_k, \boldsymbol{y}_{0:k}) \approx \mathcal{N}\left(\boldsymbol{x}_{k+1} | \hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{P}_{k+1|k}\right), \tag{12}$$

by direct evaluation of the associated moment integrals

$$\hat{\boldsymbol{x}}_{k+1|k} = \int \boldsymbol{f}(\boldsymbol{x}_k) p(\boldsymbol{x}_k | \boldsymbol{y}_{0:k}) \, \mathrm{d}\boldsymbol{x}_k, \tag{13a}$$

$$\boldsymbol{P}_{k+1|k} = \int \big(\boldsymbol{f}(\boldsymbol{x}_k) - \hat{\boldsymbol{x}}_{k+1|k}\big)(\star)^{\top} p(\boldsymbol{x}_k | \boldsymbol{y}_{0:k}) \, \mathrm{d}\boldsymbol{x}_k + \boldsymbol{Q}_k, \tag{13b}$$

simplified by the assumption of additive noise $\boldsymbol{w}_k$. Using the approximation (11), (13) reduces to two Gaussian integrals. For a general $\boldsymbol{f}$, no closed-form solutions exist, so we leverage numerical integration methods also known as *cubature rules* [14]. To this end, we transform the coordinates using the Cholesky factors of the covariance matrix $\boldsymbol{P}_{k|k} = \boldsymbol{L}_{k|k}\boldsymbol{L}^{\top}_{k|k}$, such that in the transformed coordinates, the integration is over a unit Gaussian. The LRKFs approximate the transformed integrals by evaluating the nonlinearity $\boldsymbol{f}$ 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 $\boldsymbol{\eta}^i$,

$$\hat{\boldsymbol{x}}^i_{k+1|k} = \boldsymbol{f}\left(\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{L}_{k|k}\, \boldsymbol{\eta}^i\right), \tag{14}$$

and subsequently approximate the moment integrals in (13),

$$\hat{\boldsymbol{x}}_{k+1|k} \approx \sum_{i=1}^{|\mathcal{I}|} \varpi^i \hat{\boldsymbol{x}}^i_{k+1|k}, \tag{15a}$$

$$\boldsymbol{P}_{k+1|k} \approx \sum_{i=1}^{|\mathcal{I}|} \varpi^i (\hat{\boldsymbol{x}}^i_{k+1|k} - \hat{\boldsymbol{x}}_{k+1|k})(\star)^{\top}. \tag{15b}$$

[1] For instance, given (p^X, p^Y, ψ) and $\boldsymbol{\gamma}$, the distance l_L in (9) is found by applying a univariate Newton method to compute a path length $s^{\star}_L$ corresponding to the origin of O_L in the global frame, before evaluating l_L. Thus, l_L is a function of $\boldsymbol{x}$, but this function is not differentiable.

We can use any cubature rule $\mathcal{I}$. In the implementations, we use the spherical Cubature rule [15] because it is simple and scale favorably when compared to Gauss-Hermite methods.

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

$$p\big([\boldsymbol{x}_{k+1};\boldsymbol{y}_{k+1}]|\boldsymbol{x}_k,\boldsymbol{y}_{0:k}\big) \approx \mathcal{N}\left(\begin{bmatrix}\hat{\boldsymbol{x}}_{k+1|k}\\ \hat{\boldsymbol{y}}_{k+1|k}\end{bmatrix}, \begin{bmatrix}\boldsymbol{P}_{k+1|k} & \boldsymbol{P}^{\boldsymbol{xy}}_{k+1|k}\\ \boldsymbol{P}^{\boldsymbol{yx}}_{k+1|k} & \boldsymbol{P}^{\boldsymbol{yy}}_{k+1|k}\end{bmatrix}\right) \tag{16}$$

with moment integrals (dropping time indexing for brevity),

$$\hat{\boldsymbol{y}} = \int \boldsymbol{h}(\boldsymbol{x})\mathcal{N}(\boldsymbol{x}|\hat{\boldsymbol{x}},\boldsymbol{P})\mathrm{d}\boldsymbol{x}, \tag{17a}$$

$$\boldsymbol{P}^{\boldsymbol{xy}} = \int (\hat{\boldsymbol{x}}-\boldsymbol{x})(\hat{\boldsymbol{y}}-\boldsymbol{h}(\boldsymbol{x}))^\top \mathcal{N}(\boldsymbol{x}|\hat{\boldsymbol{x}},\boldsymbol{P})\mathrm{d}\boldsymbol{x}, \tag{17b}$$

$$\boldsymbol{P}^{\boldsymbol{yy}} = \int (\hat{\boldsymbol{y}}-\boldsymbol{h}(\boldsymbol{x}))(\star)^\top \mathcal{N}(\boldsymbol{x}|\hat{\boldsymbol{x}},\boldsymbol{P})\mathrm{d}\boldsymbol{x} + \boldsymbol{R}. \tag{17c}$$

Eq. (17) implies integrating over all of the GEPs $\boldsymbol{\gamma}$, which is computationally prohibitive for realistic implementations. To circumvent this problem, we introduce Assumption 1.

Assumption 1 *$\boldsymbol{\gamma}_i$ and $\boldsymbol{\gamma}_j$, are independent for $i \neq j$.*

Given Assumption 1, the integration can be done 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 (16) on the new measurement $\boldsymbol{y}_{k+1}$ amounts to

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}^{\boldsymbol{xy}}_{k+1|k}(\boldsymbol{P}^{\boldsymbol{yy}}_{k+1|k})^{-1}, \tag{18a}$$

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{K}_{k+1|k}(\boldsymbol{y}_{k+1} - \hat{\boldsymbol{y}}_{k+1|k}), \tag{18b}$$

$$\boldsymbol{P}_{k+1|k+1} = \boldsymbol{P}_{k+1|k} - \boldsymbol{K}_{k+1|k}\boldsymbol{P}^{\boldsymbol{y}\bar{\boldsymbol{x}}}_{k+1|k}, \tag{18c}$$

which is done with respect to the vehicle state and the currently relevant map parameters. Specifically, any GEP required in the process of evaluating (10) is included explicitly in the domain over which the integral is computed.

B. Variational-Bayes LRKF

To account for the time-variability in the measurement-noise characteristics, we implement the LRKF in a VB setting [7], [16]. VB methods rely on a free-form approximation of the joint posterior of $\boldsymbol{x}_k$ and $\boldsymbol{R}_k$,

$$p(\boldsymbol{x}_k,\boldsymbol{R}_k|\boldsymbol{y}_{0:k}) \approx \underbrace{p(\boldsymbol{x}_k,|\boldsymbol{y}_{0:k})}_{Q_{\boldsymbol{x}}(\boldsymbol{x}_k)}\underbrace{p(\boldsymbol{R}_k|\boldsymbol{y}_{0:k})}_{Q_{\boldsymbol{R}}(\boldsymbol{R}_k)}, \tag{19}$$

where the objective is to determine $Q_{\boldsymbol{x}}(\boldsymbol{x}_k)$ and $Q_{\boldsymbol{R}}(\boldsymbol{R}_k)$. The VB approximation is formed by minimizing the Kullback-Leibler (KL) divergence between an approximate distribution in the form $Q_{\boldsymbol{x}}(\boldsymbol{x}_k)Q_{\boldsymbol{R}}(\boldsymbol{R}_k)$ and the true distribution $p(\boldsymbol{x}_k,\boldsymbol{R}_k|\boldsymbol{y}_{0:k})$. This amounts to

$$\min_{Q_{\boldsymbol{x}},Q_{\boldsymbol{R}}} \mathrm{KL}(Q_{\boldsymbol{x}}(\boldsymbol{x}_k)Q_{\boldsymbol{R}}(\boldsymbol{R}_k)||p(\boldsymbol{x}_k,\boldsymbol{R}_k|\boldsymbol{y}_{0:k})) = \int Q_{\boldsymbol{x}}(\boldsymbol{x}_k)Q_{\boldsymbol{R}}(\boldsymbol{R}_k)\log\left(\frac{Q_{\boldsymbol{x}}(\boldsymbol{x}_k)Q_{\boldsymbol{R}}(\boldsymbol{R}_k)}{p(\boldsymbol{x}_k,\boldsymbol{R}_k|\boldsymbol{y}_{0:k})}\right)\mathrm{d}\boldsymbol{x}_k\mathrm{d}\boldsymbol{R}_k, \tag{20}$$

where the minimizers are

$$Q_{\boldsymbol{x}}(\boldsymbol{x}_k) \propto \exp\left(\int \log p(\boldsymbol{y}_k,\boldsymbol{x}_k,\boldsymbol{R}_k|\boldsymbol{y}_{0:k-1})Q_{\boldsymbol{R}}(\boldsymbol{R}_k)\mathrm{d}\boldsymbol{R}_k\right), \tag{21a}$$

$$Q_{\boldsymbol{R}}(\boldsymbol{R}_k) \propto \exp\left(\int \log p(\boldsymbol{y}_k,\boldsymbol{x}_k,\boldsymbol{R}_k|\boldsymbol{y}_{0:k-1})Q_{\boldsymbol{x}}(\boldsymbol{x}_k)\mathrm{d}\boldsymbol{x}_k\right). \tag{21b}$$

Eq. (21) cannot be solved directly as there is coupling between $Q_{\boldsymbol{x}}$ and $Q_{\boldsymbol{R}}$. However, when the state posterior is Gaussian assumed, $Q_{\boldsymbol{x}}(\boldsymbol{x}_k) = \mathcal{N}(\boldsymbol{x}_k|\hat{\boldsymbol{x}}_{k|k},\boldsymbol{P}_{k|k})$ and the noise distribution is inverse-Wishart assumed,

$$Q_{\boldsymbol{R}}(\boldsymbol{R}_k) = \mathcal{IW}(\boldsymbol{R}_k|\nu_k,\boldsymbol{V}_k), \tag{22}$$

the integrals in (21) can be made explicitly. Hence [7],

$$\begin{aligned}
\hat{\boldsymbol{y}}_k &= \int \boldsymbol{h}(\boldsymbol{x}_k)\mathcal{N}(\boldsymbol{x}_k|\hat{\boldsymbol{x}}^-_k,\boldsymbol{P}^-_k)\mathrm{d}\boldsymbol{x}_k,\\
\boldsymbol{P}^{\boldsymbol{xy}}_k &= \int (\hat{\boldsymbol{x}}^-_k-\boldsymbol{x}_k)(\hat{\boldsymbol{y}}_k-\boldsymbol{h}(\boldsymbol{x}_k))^\top \mathcal{N}(\boldsymbol{x}_k|\hat{\boldsymbol{x}}^-_k,\boldsymbol{P}^-_k)\mathrm{d}\boldsymbol{x}_k,\\
\boldsymbol{S}_k &= \int (\hat{\boldsymbol{y}}_k-\boldsymbol{h}(\boldsymbol{x}_k))(\star)^\top \mathcal{N}(\boldsymbol{x}_k|\hat{\boldsymbol{x}}^-_k,\boldsymbol{P}^-_k)\mathrm{d}\boldsymbol{x}_k\\
&\quad + (\nu_k - n_{\boldsymbol{y}} - 1)^{-1}\boldsymbol{V}_k = \boldsymbol{T} + (\nu_k - n_{\boldsymbol{y}} - 1)^{-1}\boldsymbol{V}_k,\\
\boldsymbol{K}_k &= \boldsymbol{P}^{\boldsymbol{xy}}_k\boldsymbol{S}_k^{-1},\\
\hat{\boldsymbol{x}}_k &= \hat{\boldsymbol{x}}^-_k + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k),\\
\boldsymbol{P}_k &= \boldsymbol{P}^-_k - \boldsymbol{K}_k\boldsymbol{S}_k\boldsymbol{K}_k^\top,\\
\nu_k &= \nu^-_k + 1,\\
\boldsymbol{V}_k &= \boldsymbol{V}^-_k + \int (\boldsymbol{y}_k-\boldsymbol{h}(\boldsymbol{x}_k))(\star)^\top \mathcal{N}(\boldsymbol{x}_k|\hat{\boldsymbol{x}}_k,\boldsymbol{P}_k)\mathrm{d}\boldsymbol{x}_k,
\end{aligned} \tag{23}$$

where $(\star)^-_k = (\star)_{k|k-1}$ and $\boldsymbol{R}_k = (\nu_k - n_{\boldsymbol{y}} - 1)^{-1}\boldsymbol{V}_k$. The first six equations in (23) are the usual KF equations and the integrals involved can be approximated as in Sec. III-A. The solution to (23) is found by fixed-point iterations akin to expectation-maximization (EM) methods and (locally) converge asymptotically under some assumptions [17], [18].

The prediction step of the sufficient statistics is in this paper chosen consistent with [7]:

$$\nu^-_k = \rho(\nu_{k-1} - n_{\boldsymbol{y}} - 1) + n_{\boldsymbol{y}} + 1, \tag{24}$$

$$\boldsymbol{V}^-_k = \rho\boldsymbol{V}_{k-1}, \tag{25}$$

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

Algorithm 1 summarizes the proposed VB-LRKF.

IV. Simulation Study

We validate the proposed method in a Monte-Carlo simulation study. 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 [5]. The route is extracted using the open-source routing machine (OSRM) tool [19], and the map is represented by a sequence of points. To generate our spline-based map, we select the sequence of points as control and end points, respectively.

The GNSS position measurements nominally provide Gaussian zero-mean measurements with standard deviation 0.2m in both X and Y direction, $\boldsymbol{R}^p_{\mathrm{nom}} = \mathrm{diag}(0.2^2, 0.2^2)$.

Algorithm 1 Pseudo-code of the proposed VB-LRKF

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

for $k = 0, 1, \ldots$ **do**
  for $i \in \{1, \ldots, |\mathcal{I}|\}$ **do**
    Determine $\hat{\boldsymbol{x}}^i_{k|k-1}$ according to (14).
  end for
  Determine $\hat{\boldsymbol{x}}_{k|k-1}$, $\boldsymbol{P}_{k|k-1}$ according to (15).
  Set $\nu_k^- = \rho(\nu_{k-1} - n_{\boldsymbol{y}} - 1) + n_{\boldsymbol{y}} + 1$, $\boldsymbol{V}_k^- = \rho \boldsymbol{V}_{k-1}$.
  for $i \in \{1, \ldots, |\mathcal{I}|\}$ **do**
    Determine $\hat{\boldsymbol{y}}^i_{k|k-1}$ akin to (14)
  end for
  Determine $\hat{\boldsymbol{y}}_{k|k-1}$, $\boldsymbol{T}$, $\boldsymbol{P}^{\boldsymbol{xy}}_{k|k-1}$ akin to (15).
  Set $\hat{\boldsymbol{x}}_k^{(0)} = \hat{\boldsymbol{x}}_{k|k-1}$, $\boldsymbol{P}_k^{(0)} = \boldsymbol{P}_{k|k-1}$, $\nu_k = 1 + \nu_k^-$, $\boldsymbol{V}_k^{(0)} = \boldsymbol{V}_k^-$, $j = 0$.
  while `not_converged` **do**

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

    $j = j + 1$
  end while
  Set $\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_k^{(j)}$, $\boldsymbol{P}_{k|k} = \boldsymbol{P}_k^{(j)}$, $\boldsymbol{V}_k = \boldsymbol{V}_k^{(j)}$.
end for

Furthermore, the camera measurements provide lane measurements that nominally are Gaussian distributed according to $\boldsymbol{y}_k^c \sim \mathcal{N}(\boldsymbol{h}^c(\boldsymbol{x}_k), \boldsymbol{R}_c)$, where $\boldsymbol{R}^c_{\text{nom}} = \text{diag}(0.02\mathbf{1}_{10})$. 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 larger than the added random noise. We run Algorithm 1 for 100 Monte-Carlo runs for different forgetting factors., with each simulation 20s long. We generate the initial state by sampling it from a Gaussian distribution with 5m initial standard deviation on the position. All measurements arrive with sampling rate 10Hz but the prediction step is performed at 100Hz, that is, when executing Algorithm 1 at 100Hz, the measurement update step and weight update 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) $\boldsymbol{R}^p = \boldsymbol{R}^p_{nom}, \boldsymbol{R}^c = \boldsymbol{R}^c_{nom}$;
2) $\boldsymbol{R}^p = 10^2 \boldsymbol{R}^p_{nom}, \boldsymbol{R}^c = \boldsymbol{R}^c_{nom}$;
3) $\boldsymbol{R}_p = \boldsymbol{R}^p_{nom}, \boldsymbol{R}_c = 5^2 \boldsymbol{R}_{c,nom}$.

GNSS outliers occur every tenth second starting at 5s and last three seconds. Similarly, we have camera outliers occurring every tenth second that last three seconds, starting at 10s.

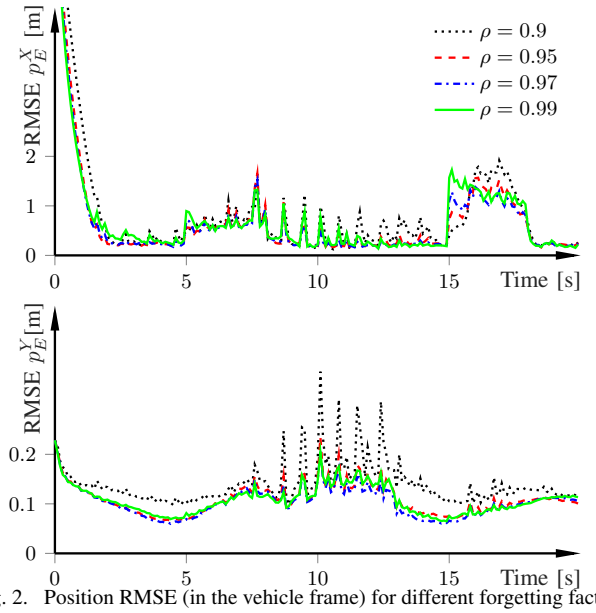

Fig. 2. Position RMSE (in the vehicle frame) for different forgetting factors over 100 Monte-Carlo runs with 0.1m initial map standard deviation.

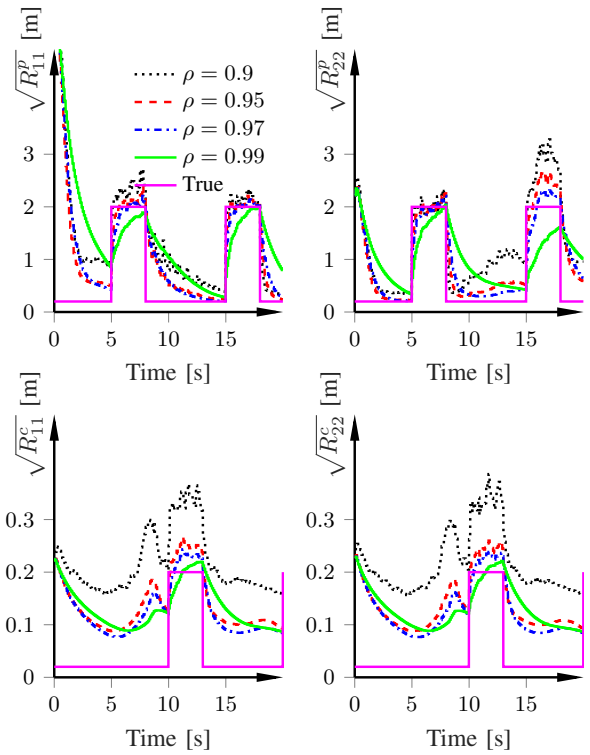

Fig. 3. Noise estimates for the GNSS position measurements and measured distances to each road boundary, averaged over the 100 Monte-Carlo runs.

Fig. 2 shows the corresponding position RMSEs, and Fig. 3 shows the corresponding noise estimates averaged over the 100 Monte-Carlo runs. The choice of forgetting factor is a trade-off between how fast to adjust to changing measurement quality and estimate smoothness. For instance, $\rho = 0.9$ gives quick adaptation of the estimates in case of outliers, but in steady-state, the estimation accuracy is generally decreased. At the same time, a high forgetting factor ($\rho = 0.99$) gives smooth estimates, but in case of

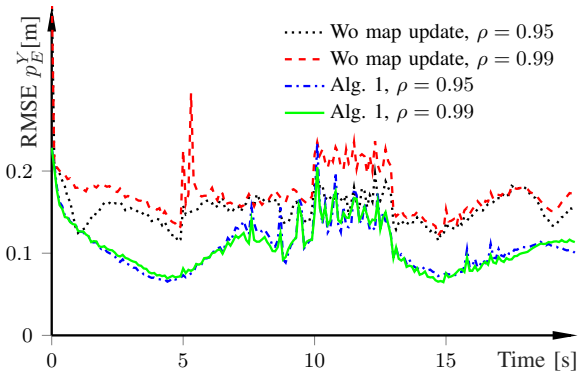

Fig. 4. Position RMSE in the vehicle frame with map update and without map update for an initial map standard deviation of 0.1m.

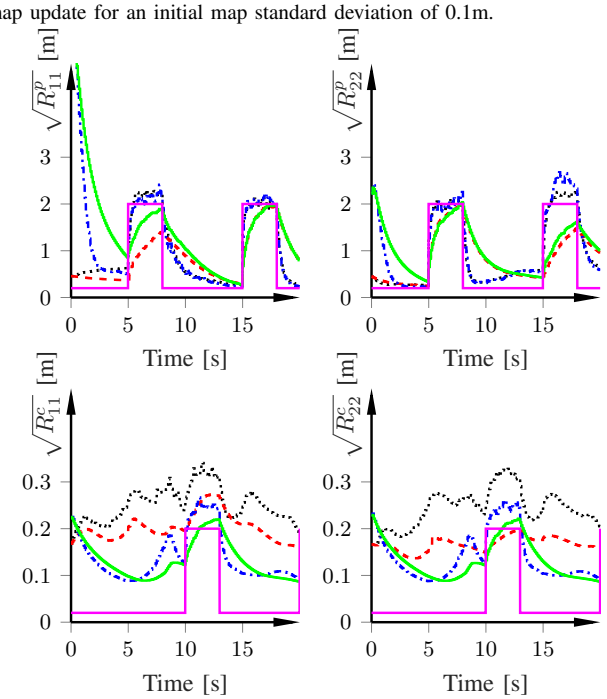

Fig. 5. Average noise estimates for the GNSS position measurements and measured distances to each road boundary with map update and without map update (Algorithm 1 without map states) for an initial map standard deviation of 0.1m. Same notation as in Fig. 4.

outliers the adjustment is slower (c.f. at around 15s).

To conclude the evaluation, we compare 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. 4 shows the results of Algorithm 1 with and without map updates for two different forgetting factors. Irrespective of the forgetting factor, there is roughly a 0.1m improvement, which corresponds to the uncertainty in the map. Hence, when updating the map, most of the errors are corrected and the algorithm exhibits improved positioning.

Fig. 5 displays the corresponding noise estimates. To account for the unmodeled errors in the map, when executing Algorithm 1 without map updates, the camera noise estimates are inflated. While this is intuitive since the unmodeled map uncertainty is injected into the noise estiamtes, it leads to degraded performance compared to including map updates.

V. Conclusion

The proposed spline-based method for map modeling is flexible since it can handle a more diverse set of roads than traditional curvature-based road modeling. This combined with the VB framework allows us to adapt to the time-varying measurement reliability for both the GNSS and camera measurements, as well as uncertianties in the map. The proposed method is a viable alternative to other noise-adaptive filters such as particle filters or filters based on IMM. In future work, we plan to perform hardware-in-the-loop simulations for assessing real-time feasibility, experimentally verify the proposed method, and compare to our previously developed method [5].

References

[1] F. Gustafsson, "Automotive safety systems," *IEEE Signal Process. Mag.*, vol. 26, no. 4, pp. 32–47, July 2009.

[2] A. Eidehall, J. Pohl, and F. Gustafsson, "Joint road geometry estimation and vehicle tracking," *Control Eng. Pract.*, vol. 15, no. 12, pp. 1484–1494, 2007.

[3] C. Lundquist and T. B. Schön, "Joint ego-motion and road geometry estimation," *Information Fusion*, vol. 12, no. 4, pp. 253–263, 2011.

[4] M. Sahmoudi and R. Landry, "A nonlinear filtering approach for robust multi-GNSS RTK positioning in presence of multipath and ionospheric delays," *IEEE J. Selected Topics Signal Process.*, vol. 3, no. 5, pp. 764–776, 2009.

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

[6] M. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, University of London, 2003.

[7] S. Särkkä and J. Hartikainen, "Non-linear noise adaptive Kalman filtering via variational Bayes," in *IEEE Int. Workshop Machine Learning for Signal Processing*, Southampton, UK, Sep. 2013.

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

[9] A. F. Garcia-Fernandez, L. Hammarstrand, M. Fatemi, and L. Svensson, "Bayesian road estimation using onboard sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1676–1689, 2014.

[10] K. Berntorp, A. Weiss, and S. Di Cairano, "Integer ambiguity resolution by mixture Kalman filter for improved GNSS precision," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 4, pp. 3170–3181, 2020.

[11] M. Greiff and K. Berntorp, "Optimal measurement projections with adaptive mixture Kalman filtering for GNSS positioning," in *Amer. Control Conf.*, 2020.

[12] M. Greiff, K. Berntorp, S. Di Cairano, and K. Kim, "Mixed-integer linear regression Kalman filters for GNSS positioning," in *Conf. Control Techn. Applications*, San Diego, CA, Aug. 2021.

[13] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "Polylanenet: Lane estimation via deep polynomial regression," in *Int. Conf. Pattern Recognition*, 2021.

[14] J. Steinbring and U. D. Hanebeck, "SSKF: The smart sampling Kalman filter," in *Int. Conf. Information Fusion*, Istanbul, Turkey, Jul. 2013.

[15] I. Arasaratnam, "Cubature Kalman filtering theory & applications," Ph.D. dissertation, McMaster University, 2009.

[16] S. Särkkä and A. Nummenmaa, "Recursive noise adaptive Kalman filtering by variational Bayesian approximations," *IEEE Trans. Automat. Contr.*, vol. 54, no. 3, pp. 596–600, 2009.

[17] I. S. Mbalawata, S. Särkkä, M. Vihola, and H. Haario, "Adaptive Metropolis algorithm using variational Bayesian adaptive Kalman filter," *Computational Statistics & Data Analysis*, vol. 83, pp. 101–115, 2015.

[18] M.-A. Sato, "Online model selection based on the variational Bayes," *Neural computation*, vol. 13, no. 7, pp. 1649–1681, 2001.

[19] OSRM, "Homepage of the open source road map project," last accessed 08-19-2021. [Online]. Available: http://project-osrm.org/