

Bayesian Sensor Fusion for Joint Vehicle Localization and Road Mapping Using Onboard Sensors

Berntorp, Karl; Greiff, Marcus; Di Cairano, Stefano; Miraldo, Pedro

TR2023-074 June 27, 2023

Abstract

We propose a method 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 described by a parameter vector having a Gaussian prior representing the uncertainty of the prior map. 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 an interacting multiple-model (IMM) setting to choose the best combination of the estimators with the vehicle state and the parameter vector of the map as the state vector. In a simulation study, we compare vehicle models with varying complexity, and on a real road segment we show that the proposed method can accurately adjust to changing noise conditions and correct for errors in the prior map.

International Conference on Information Fusion (FUSION) 2023

© 2023 MERL. This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Bayesian Sensor Fusion for Joint Vehicle Localization and Road Mapping Using Onboard Sensors

Karl Berntorp, Marcus Greiff, Stefano Di Cairano, Pedro Miraldo

Abstract—We propose a method 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 described by a parameter vector having a Gaussian prior representing the uncertainty of the prior map. 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 an interacting multiple-model (IMM) setting to choose the best combination of the estimators with the vehicle state and the parameter vector of the map as the state vector. In a simulation study, we compare vehicle models with varying complexity, and on a real road segment we show that the proposed method can accurately adjust to changing noise conditions and correct for errors in the prior map.

I. 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 some established ADAS 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 [1].

Road-vehicle positioning can be approached in numerous ways depending on the available sensor suite and the communication interface that is being employed. Here, we develop a method for the joint vehicle-positioning and road-map estimation problem by fusing 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. In combination with a computer-vision (CV) algorithm, the camera provides measurements of the distance between the lane markings and the vehicle, in addition to measurements of the road geometry [2]. 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 light 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 [3], [4]. While being reliable most of the time, GNSS measurements are prone to occasional errors, which means that both camera and GNSS feature measurements with time-varying reliability.

The time-varying reliability of the measurements is hard to model and predict, because they depend on numerous effects. Hence, to account for said issues, we model the time-varying measurement reliability as a variation on the noise models of the related measurements. We pose the resulting nonlinear estimation problem in an interacting multiple model (IMM) framework, which we combine with linear-regression Kalman filters (LRKFs) [5] to handle the nonlinearities in the estimation model. We represent the map using a spline representation with implicit continuity and we assign Gaussian priors to the spline coefficients. We incorporate the spline coefficients into the state-space formulation and jointly estimate said coefficients with the vehicle state using the IMM-LRKF, thus in effect performing simultaneous localization and mapping with prior map information. Each LRKF executes with its own belief of the measurement noise characteristics, and the estimate from each LRKF is weighted according to how likely it is to best explain the measurements given the state estimates. By formulating the positioning and road-map estimation problem jointly in this way, our method additionally estimates the noise characteristics with marginal detection time, provided the set of LRKFs in the IMM contain the “true” noise covariance.

A. Relation to Previous Work

In [6], we developed a method for vehicle localization using GNSS measurements, wheel-speed sensors, a steering-wheel encoder, a prior map of the road, a camera that measured the distance to the left and right lanes, and optionally an inertial measurement unit. The difference of the current work to [6] is that the current work introduces road map estimation into the problem formulation, whereas [6] focuses only on vehicle state estimation using a *known map*.

Joint vehicle state and road-map estimation has been researched using various sensor constellations and different estimation techniques. For instance, [7] fuses information from several (local) sensors to perform joint road geometry estimation and vehicle tracking. This work was extended in [8], where a forward looking camera and radar, together with an inertial measurement unit (IMU), a steering wheel sensor, wheel speed sensors, and a new road-geometry model are

leveraged in an extended Kalman filter (EKF). A similar work is [9], which in addition to the sensors in [8] 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 focus on the road-geometry estimation problem. 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 and several textbook vehicle-control algorithms employ such representation [10]. However, a spline representation of the road is more general. With the estimated spline-based map, we extract the estimated curvature 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: [11], which uses inertial sensors, wheel-speed sensors, and the steering-wheel angle sensor in combination with GNSS position measurements to perform vehicle-state estimation; [12], which performs tire radii estimation for improving vehicle odometry using GNSS measurements; and [13], which uses GNSS measurements in combination with camera, IMU, and range measurements in a collaborative estimation approach. While [11], [12] use GNSS to make related vehicle estimation problems observable, the IMM method in this paper is specifically designed to handle outliers.

B. Notation:

Throughout, $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicates that the vector $\mathbf{x} \in \mathbb{R}^{n_x}$ is Gaussian distributed with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Matrices are written in capital bold font as \mathbf{X} , and the element on row i and column j of \mathbf{X} is denoted with X_{ij} . We let $\hat{\mathbf{x}}_{j|m}$ denote the estimate of \mathbf{x} at time step j given the measurement sequence $\mathbf{y}_{0:m} = \{\mathbf{y}_0, \dots, \mathbf{y}_m\}$. With $p(\mathbf{x}_k | \mathbf{y}_{0:k})$, we mean the posterior density function of the state \mathbf{x}_k from time step 0 to time step k given $\mathbf{y}_{0:k}$. The concatenation of two vectors $\mathbf{x} \in \mathbb{R}^{n_x}$ and $\mathbf{y} \in \mathbb{R}^{n_y}$ is $[\mathbf{x}; \mathbf{y}] = [\mathbf{x}^\top, \mathbf{y}^\top]^\top \in \mathbb{R}^{n_x+n_y}$. 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, and $(\mathbf{a})(\star)^\top = (\mathbf{a})(\mathbf{a})^\top$ for an expression \mathbf{a} .

II. 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 $O_{R,l}$ 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 .

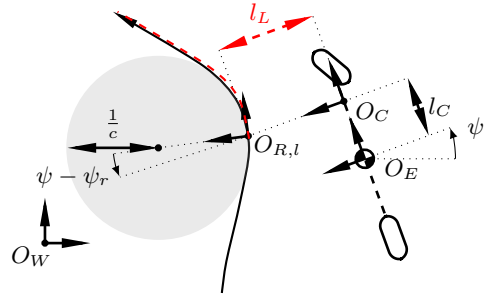


Fig. 1. The relation between the vehicle frame O_E , the camera frame O_C , the road frame $O_{R,l}$, and the world frame O_W . The distance between the vehicle's longitudinal x -axis and the left lane boundary is l_L , and the shaded circle depicts the road curvature at the origin of $O_{R,l}$ (here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

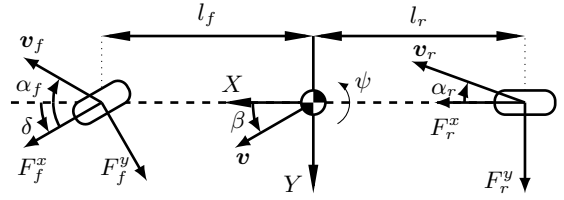


Fig. 2. A schematic of the single-track model and related notation.

A. Vehicle Models

A model based on force balances is generally more accurate than a kinematic model but the differences are small for regular driving [14], and 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 may be difficult, or at least tedious, to estimate in real time [15]. We introduce two assumptions that allows to model the vehicle dynamics by a single-track (i.e., bicycle) model, and have been shown to be valid in normal driving scenarios [10].

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.*

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, ψ is the yaw, $\mathbf{v} = [v^X, v^Y]^\top$ is the velocity vector, and subscripts f, r denote front and rear, respectively. The state is $\mathbf{x} = [p^X, p^Y, v^X, v^Y, \dot{\psi}]^\top \in \mathbb{R}^{n_x}$, $n_x = 5$ where p^X, p^Y is the Cartesian global vehicle position, v^X, v^Y is the longitudinal and lateral velocity of the vehicle in O_E , and $\dot{\psi}$ is the yaw rate. The equations of motion are

$$M(\dot{v}^X - v^Y \dot{\psi}) = F_f^x \cos(\delta) + F_r^x - F_f^y \sin(\delta), \quad (1a)$$

$$M(\dot{v}^Y + v^X \dot{\psi}) = F_f^y \cos(\delta) + F_r^y + F_f^x \sin(\delta), \quad (1b)$$

$$I \ddot{\psi} = l_f (F_f^y \cos(\delta) + F_f^x \sin(\delta)) - l_r F_r^y, \quad (1c)$$

where M is the vehicle mass and I is the inertia. Because we focus on normal driving conditions, the longitudinal and lateral tire forces can be approximated as linear functions of the wheel slip ratio κ and the slip angle α ,

$$F_i^x \approx C_i^x \kappa_i, \quad F_i^y \approx C_i^y \alpha_i, \quad i = f, r, \quad (2)$$

where C_i^x, C_i^y are the longitudinal and lateral stiffness, respectively. The wheel slip is defined following [16], as

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

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

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

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

$$\begin{bmatrix} \dot{p}^X \\ \dot{p}^Y \end{bmatrix} = \mathbf{R}(\psi) \begin{bmatrix} v^X \\ v^Y \end{bmatrix}, \quad \mathbf{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix}. \quad (5)$$

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

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

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

2) *Kinematic Single-Track Model*: The kinematic single-track model is also based on the geometry in Fig. 2 but does not need knowledge of the mass, inertia, and friction parameters. With its three states $\mathbf{z} = [p^X, p^Y, \psi] \in \mathbb{R}^{n_z}$, $n_z = 3$, the kinematic single-track model has lower complexity than the five-state dynamic single-track model (6). Here, instead of being used as inputs to the model, the wheel-speed measurements directly provide the vehicle velocity. In continuous time, the model is

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

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 (7) concisely as

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

with Gaussian zero-mean process noise, $\mathbf{w}_k^z \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^z)$.

B. Road Model

One of the most common ways to represent the road is by defining it locally by a clothoid [8] or sets of clothoids [9] using a curvature, denoted by c . The road curvature can be defined in various ways, but a common definition is illustrated in Fig. 1. The road curvature is often approximated as a linear function, which results in a clothoidal expression of

the road and is commonly used in automotive applications. Furthermore, it approximately agrees with road-construction principles, at least on nonurban roads [17]. Note that the clothoidal approximation is clearly violated in some situations, for example, when a part of the road is a straight line followed by a clothoidal stretch. However, in many situations (e.g., for highway or suburban driving) it is a good local approximation.

Because such curvature representations are restrictive, we work with a spline representation of the road. To this end, we make use of polynomial splines $P_i : [0, 1] \mapsto \mathbb{R}^d$ in $\lambda_i \in [0, 1]$ defined by a parameter vector $\boldsymbol{\alpha}_i$. Nominally, $d = 3$ with the first two elements of P_i representing the Cartesian planar position of the center of the road, and the last dimension defining the road width. A road in the map is comprised of N such splines, where $S : [0, N] \mapsto \mathbb{R}^d$, where

$$S(s; \bar{\boldsymbol{\alpha}}) = P_i(\lambda_i; \boldsymbol{\alpha}_i), \quad i = \lceil s \rceil, \quad \lambda_i = s - i + 1. \quad (9)$$

and $\bar{\boldsymbol{\alpha}} := \{\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_N\}$ parameterizes the road in the map. Here, the i th spline of the road is a Bézier curve of degree \bar{m} ,

$$P_i(\lambda_i; \boldsymbol{\alpha}_i) = \sum_{j=0}^{\bar{m}} \binom{\bar{m}}{j} (1 - \lambda_i)^{\bar{m}-j} (\lambda_i)^j, \quad (10)$$

parameterized in $\bar{m} + 1$ points, with $\boldsymbol{\alpha}_i = \{\mathbf{s}_{i,j} \in \mathbb{R}^d\}_{j=0}^{\bar{m}}$. The derivative of (10) is yet another Bézier curve, satisfying

$$\begin{aligned} \frac{d}{d\lambda} P_i(0; \boldsymbol{\alpha}_i) &= \bar{m}(\mathbf{s}_{i,1} - \mathbf{s}_{i,0}), \\ \frac{d}{d\lambda} P_i(1; \boldsymbol{\alpha}_i) &= \bar{m}(\mathbf{s}_{i,\bar{m}} - \mathbf{s}_{i,\bar{m}-1}). \end{aligned}$$

Consequently, there exist geometric relationships between parameters of consecutive splines that allow the enforcement of continuity of the road. This is summarized in Proposition 1.

Proposition 1 Consider two consecutive polynomials (10). A sufficient condition for $\frac{d^n}{d\lambda^n} P_i(1; \boldsymbol{\alpha}_i) = \frac{d^n}{d\lambda^n} P_{i+1}(0; \boldsymbol{\alpha}_{i+1})$ is

$$\mathbf{s}_{i,\bar{m}} = \mathbf{s}_{i+1,0} = \mathbf{s}_{i,\bar{m}-k} + \mathbf{s}_{i+1,k}, \quad \forall k = 1, \dots, n. \quad (11)$$

Proposition 1 provides a way to find a parametrization of the Bézier curves that implicitly enforces continuity at the end-points of the polynomial segments. If we seek a parametrization that is continuous of order n , we can simply take splines of degree $\bar{m} = 2n + 1$ with $2(n + 1)$ control points, and let

$$\mathbf{s}_{i,k} = \mathbf{s}_{i,0} + \mathbf{v}_{i,k}, \quad \forall k = 1, \dots, n, \quad (12a)$$

$$\mathbf{s}_{i,k} = \mathbf{s}_{i,\bar{m}} - \mathbf{v}_{i,k}, \quad \forall k = n + 1, \dots, 2n + 1, \quad (12b)$$

for any set of vectors $\{\mathbf{v}_{i,k}\}_{k=1}^n$. The sufficient conditions for continuity according to Proposition 1 are thus satisfied if

$$\begin{cases} \mathbf{s}_{i,\bar{m}} = \mathbf{s}_{i+1,0} \\ \mathbf{v}_{i,k} = -\mathbf{v}_{i+1,\bar{m}-k} \end{cases} \quad \forall k = 1, \dots, n, i = 1, \dots, N. \quad (13)$$

This geometric relationship is illustrated in Fig. 3, showing the center of a lane in Cartesian coordinates (the first two dimensions of S). If this curve is defined in terms of the end-points, $\mathbf{s}_{i,0}$, and directions, $\mathbf{v}_{i,0}$, then continuity can be enforced implicitly. Here, $n = 1$ is sufficient to guarantee continuous lane boundaries, and we therefore define a new

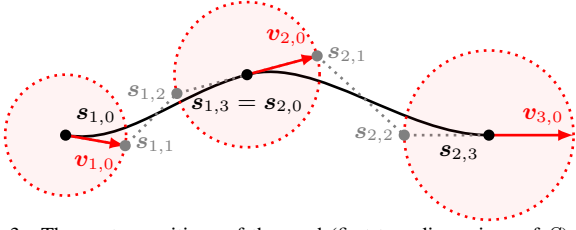


Fig. 3. The center positions of the road (first two dimensions of S). In this example, there are $N = 2$ polynomial segments of degree $\bar{m} = 3$, and continuity is enforced in the positions and velocities ($n = 1$). No matter how the parameters $\bar{\gamma} = [s_{1,0}; v_{1,0}; s_{2,0}; v_{2,0}; s_{3,0}; v_{3,0}]$ are realized, when mapped to the control points through (13), the path and its derivative are continuous in the path variable s by Proposition 1.

parameter vector $\bar{\gamma} = [\gamma_1; \dots; \gamma_{N+1}]$ where $\gamma_i = [s_{1,0}; v_{i,0}]$. When evaluating the road boundaries, $\bar{\gamma}$ is first mapped to the parameter vector $\bar{\alpha}$ using (13) before evaluating (9). Next, we introduce uncertainty in the map by assigning a Gaussian prior on each γ_i , as $\gamma_i \sim \mathcal{N}(\mu_i^\gamma, \Sigma_i^\gamma)$, but the time evolution of these parameters is hard to model from physical reasoning. Assuming slow changes in the map, we let

$$\bar{\gamma}_{k+1} = \bar{\gamma}_k + w_k^\gamma, \quad w_k^\gamma \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^\gamma). \quad (14)$$

The process noise in the map is small, with $\mathbf{Q}^\gamma \approx \mathbf{0}$. Thus, there is effectively no prediction performed for the map state $\bar{\gamma}$ and all changes are attributed to the measurement update.

Remark 1 For a more compact notation, in the following we use \bar{x} to mean either $\bar{x} = [x; \bar{\gamma}]$ or $\bar{x} = [z; \bar{\gamma}]$. Similarly $\bar{f}(\bar{x}; \mathbf{u})$ denotes the functional relationships in (6) or (8) combined with (14), often omitting \mathbf{u} for brevity. Also, we let the full process noise be denoted by $\bar{w} \sim \mathcal{N}(\mathbf{0}, \bar{\mathbf{Q}})$.

C. Measurement Model

We consider the GNSS position measurements \mathbf{y}_k^p generated by an estimator using code and carrier-phase measurements, for example, by the methods in [18]–[20]. 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 μ_k^p and covariance \mathbf{R}_k^p are considered to be time varying, resulting in $\mathbf{y}_k^p \sim \mathcal{N}(\mu_k^p, \mathbf{R}_k^p)$. For simplicity but without loss of generality, we let $\mathbf{y}^p \in \mathbb{R}^2$.

When using the dynamic single-track model, we can utilize an IMU measuring the vehicle body frame longitudinal, $a_k^X = \dot{v}_k^X - v_k^Y \psi_k$, and lateral, $a_k^Y = \dot{v}_k^Y + v_k^X \psi_k$, 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 $\mathbf{y}_k^a = [a_k^X, a_k^Y, \dot{\psi}_k]^\top$. 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 [15], [21], [22]. For now, we assume that the bias has been predetermined offline but refer to some of our previous work [15], [21] for IMU bias estimation in similar applications. Also, note that the kinematic single-track model (8) does not utilize an IMU.

The camera in combination with a CV algorithm 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, 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 over their domain defined in s . This results in the measurement equation

$$\mathbf{h}^c = [l_L; l_R; f_L(s_L^1); \dots; f_L(s_L^{n_s}); f_R(s_R^1); \dots; f_R(s_R^{n_s})]. \quad (15)$$

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

$$\mathbf{y}_k = \mathbf{h}(\bar{\mathbf{x}}_k, \mathbf{u}_k) + \mathbf{e}_k \in \mathbb{R}^{n_y}, \quad (16)$$

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

Remark 2 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 (16) renders $\bar{\mathbf{x}}_k$ locally observable.

III. BAYESIAN SENSOR FUSION OF GNSS AND CAMERA

Irrespective of using the dynamic or kinematic single-track model, the resulting estimation model contains multiple nonlinearities, both as a result of the vehicle model but also as a result of the map model. Hence, an analytic solution to the estimation problem does not exist. In fact, not even the Jacobians of the measurement equation are known in closed form, but have to be numerically approximated if used.¹ As such, employing an extended Kalman filter (EKF) may work, but will be computationally burdensome and potentially inaccurate. We therefore consider derivative-free estimation methods. Due to the rich set of measurements, the estimation problem is assumed to be unimodal and a particle filter (PF) is therefore deemed unnecessary for the task at hand. A convenient middle-ground between complexity and accuracy is to use linear-regression Kalman filters (LRKFs), which we employ in this work embedded in an IMM framework.

A. Linear-Regression Kalman Filter

For each LRKF, we approximate the posterior density by its first two moments, leading to

$$p(\bar{\mathbf{x}}_k | \mathbf{y}_{0:k}) \approx \mathcal{N}(\hat{\bar{\mathbf{x}}}_{k|k}, \mathbf{P}_{k|k}). \quad (17)$$

¹For instance, given (p^X, p^Y, ψ) and $\bar{\gamma}$, the distance l_L in (15) is found by applying a univariate Newton method to compute a path length s_L^* corresponding to the origin of $O_{R,L}$ in the global frame, before evaluating l_L . As such, l_L is a function of $\bar{\mathbf{x}}$, but this function is not differentiable.

Given the assumed Gaussian filtering posterior (17) at time step k , the distribution of the state prediction at time step $k+1$ is approximated by a Gaussian,

$$p(\bar{\mathbf{x}}_{k+1}|\bar{\mathbf{x}}_k, \mathbf{y}_{0:k}) \approx \mathcal{N}(\bar{\mathbf{x}}_{k+1}|\hat{\bar{\mathbf{x}}}_{k+1|k}, \mathbf{P}_{k+1|k}), \quad (18)$$

by direct evaluation of the associated moment integrals

$$\hat{\bar{\mathbf{x}}}_{k+1|k} = \int \bar{\mathbf{f}}(\bar{\mathbf{x}}_k) p(\bar{\mathbf{x}}_k|\mathbf{y}_{0:k}) d\bar{\mathbf{x}}_k, \quad (19a)$$

$$\mathbf{P}_{k+1|k} = \int (\bar{\mathbf{f}}(\bar{\mathbf{x}}_k) - \hat{\bar{\mathbf{x}}}_{k+1|k})(\star)^\top p(\bar{\mathbf{x}}_k|\mathbf{y}_{0:k}) d\bar{\mathbf{x}}_k + \bar{\mathbf{Q}}_k, \quad (19b)$$

simplified by the additive $\bar{\mathbf{w}}_k$. By insertion of the approximation in (17), this becomes equivalent to evaluating two Gaussian integrals. For a general $\bar{\mathbf{f}}$, no closed-form solutions exist, but numerical integration methods also known as *cubature rules* can be employed [23]. To facilitate this, we transform the coordinates $\boldsymbol{\xi}_k = \mathbf{L}_{k|k}^{-1}(\bar{\mathbf{x}}_k - \hat{\bar{\mathbf{x}}}_{k|k})$, using the Cholesky factorization of the covariance matrix $\mathbf{P}_{k|k} = \mathbf{L}_{k|k} \mathbf{L}_{k|k}^\top$. The LRKFs approximate the transformed integrals by evaluating the nonlinearity $\bar{\mathbf{f}}$ in a set of integration points $\mathcal{P} = \{\omega^i, \boldsymbol{\xi}^i\}_{i=1}^{|\mathcal{P}|}$, where $|\mathcal{P}|$ is the total number of points used. Hence, for each such point $\boldsymbol{\xi}^i$ and the estimate $\hat{\bar{\mathbf{x}}}_{k|k}$,

$$\hat{\bar{\mathbf{x}}}_{k+1|k}^i = \bar{\mathbf{f}}(\hat{\bar{\mathbf{x}}}_{k|k} + \mathbf{L}_{k|k} \boldsymbol{\xi}^i), \quad (20)$$

and approximate the moment integrals in (19) as

$$\hat{\bar{\mathbf{x}}}_{k+1|k} \approx \sum_{i=1}^{|\mathcal{P}|} \omega^i \hat{\bar{\mathbf{x}}}_{k+1|k}^i, \quad (21a)$$

$$\mathbf{P}_{k+1|k} \approx \sum_{i=1}^{|\mathcal{P}|} \omega^i (\hat{\bar{\mathbf{x}}}_{k+1|k}^i - \hat{\bar{\mathbf{x}}}_{k+1|k})(\star)^\top. \quad (21b)$$

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

$$p([\bar{\mathbf{x}}_{k+1}; \mathbf{y}_{k+1}]|\bar{\mathbf{x}}_k, \mathbf{y}_{0:k}) \approx \mathcal{N}\left(\begin{bmatrix} \hat{\bar{\mathbf{x}}}_{k+1|k} \\ \hat{\mathbf{y}}_{k+1|k} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{k+1|k} & \mathbf{P}_{k+1|k}^{\bar{\mathbf{x}}\mathbf{y}} \\ \mathbf{P}_{k+1|k}^{\mathbf{y}\bar{\mathbf{x}}} & \mathbf{P}_{k+1|k}^{\mathbf{y}\mathbf{y}} \end{bmatrix}\right) \quad (22)$$

with moment integrals (dropping the subindex for brevity),

$$\hat{\mathbf{y}} = \int \mathbf{h}(\bar{\mathbf{x}}) \mathcal{N}(\bar{\mathbf{x}}|\hat{\bar{\mathbf{x}}}, \mathbf{P}) d\bar{\mathbf{x}}, \quad (23a)$$

$$\mathbf{P}^{\bar{\mathbf{x}}\mathbf{y}} = \int (\hat{\bar{\mathbf{x}}} - \bar{\mathbf{x}})(\hat{\mathbf{y}} - \mathbf{h}(\bar{\mathbf{x}}))^\top \mathcal{N}(\bar{\mathbf{x}}|\hat{\bar{\mathbf{x}}}, \mathbf{P}) d\bar{\mathbf{x}}, \quad (23b)$$

$$\mathbf{P}^{\mathbf{y}\mathbf{y}} = \int (\hat{\mathbf{y}} - \mathbf{h}(\bar{\mathbf{x}}))(\star)^\top \mathcal{N}(\bar{\mathbf{x}}|\hat{\bar{\mathbf{x}}}, \mathbf{P}) d\bar{\mathbf{x}} + \mathbf{R}. \quad (23c)$$

Eq. (23) implies integrating over $\bar{\gamma}$, including all subintervals of \mathcal{S} , and is computationally prohibitive for realistic implementations. To circumvent this, we introduce Assumption 3.

Assumption 3 *The spline coefficients for any two subintervals γ_i and γ_j are independent for $i \neq j$.*

Given Assumption 3, the integration can be done with respect to each individual polynomial in the spline. When performing

the moment evaluations, it is possible that the parameters of two adjacent splines 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 (22) on the new measurement \mathbf{y}_{k+1} amounts to the usual Kalman filter update

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}^{\bar{\mathbf{x}}\mathbf{y}} (\mathbf{P}_{k+1|k}^{\mathbf{y}\mathbf{y}})^{-1}, \quad (24a)$$

$$\hat{\bar{\mathbf{x}}}_{k+1|k+1} = \hat{\bar{\mathbf{x}}}_{k+1|k} + \mathbf{K}_{k+1|k} (\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1|k}), \quad (24b)$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1|k} \mathbf{P}_{k+1|k}^{\bar{\mathbf{x}}\bar{\mathbf{x}}}, \quad (24c)$$

where the update is done with respect to the vehicle state and the currently relevant map parameters. For the numerical examples, we use an LRFK with the point set defined by the spherical cubature rule, resulting in the cubature KF (CKF) [24]. Other integration rules, such as the unscented transform, can be used without modifications to the underlying method.

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

B. Interacting Multiple-Model LRFK

LRKFs usually assume a known process noise and measurement covariance matrix. However, the reliability of both the GNSS measurements and camera-based measurements varies in time. To account for this, we implement the LRFK in an IMM framework [22], [25], in which we have a set of m 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,

$$\bar{\mathbf{x}}_{k+1} = \bar{\mathbf{f}}(\bar{\mathbf{x}}_k; \mathbf{u}_k) + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k), \quad (25a)$$

$$\mathbf{y}_k = \mathbf{h}(\bar{\mathbf{x}}_k, \mathbf{u}_k) + \mathbf{e}_k(\theta_k), \quad \mathbf{e}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k^{\theta_k}), \quad (25b)$$

where the mode parameter $\theta_k \in [1, m] \subset \mathbb{N}$ evolves according to a finite-state Markov chain with transition probability matrix $\boldsymbol{\Pi} \in [0, 1]^{m \times m}$. For every possible θ_k , we assign a unique measurement noise covariance matrix from $\{\mathbf{R}^{\theta_k} \in \mathbb{R}^{n_e \times n_e} | \mathbf{R}^{\theta_k} = (\mathbf{R}^{\theta_k})^\top, \mathbf{R}^{\theta_k} \succ \mathbf{0}\}_{\theta_k=1}^m$.

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

$$\bar{q}_k^i = \sum_{j=1}^m \Pi_{ij} q_{k-1}^j, \quad (26a)$$

$$\hat{\bar{\mathbf{x}}}_{k-1|k-1}^i = \sum_{j=1}^m \Pi_{ij} \frac{q_{k-1}^j}{\bar{q}_k^i} \hat{\bar{\mathbf{x}}}_{k-1|k-1}^j, \quad (26b)$$

$$\mathbf{P}_{k-1|k-1}^i = \sum_{j=1}^m \Pi_{ij} \frac{q_{k-1}^j}{\bar{q}_k^i} \left(\mathbf{P}_{k-1|k-1}^j + (\hat{\bar{\mathbf{x}}}_{k-1|k-1}^j - \hat{\bar{\mathbf{x}}}_{k-1|k-1}^i)(\star)^\top \right). \quad (26c)$$

Next, we execute a filter bank of m LRFs to find the estimate of $\bar{\mathbf{x}}_k$, where the j th LRF executes using \mathbf{R}^j . The state posterior is expressed using the law of total probability as a Gaussian mixture of m components,

$$\begin{aligned} p(\bar{\mathbf{x}}_k | \mathbf{y}_{0:k}) &= \sum_{j=1}^m p(\bar{\mathbf{x}}_k^j | \mathbf{y}_{0:k}) = \sum_{j=1}^m \frac{p(\mathbf{y}_k | \bar{\mathbf{x}}_k^j) p(\bar{\mathbf{x}}_k | \mathbf{y}_{0:k-1})}{p(\mathbf{y}_k | \mathbf{y}_{0:k-1})} \\ &\approx \sum_{j=1}^m q_k^j \mathcal{N}(\bar{\mathbf{x}}_k^j | \hat{\bar{\mathbf{x}}}_{k|k-1}^j, \mathbf{P}_{k|k-1}^j), \end{aligned} \quad (27)$$

where

$$q_k^j \propto p(\mathbf{y}_k | \bar{\mathbf{x}}_k^j) = \mathcal{N}(\mathbf{y}_k | \hat{\mathbf{y}}_{k|k-1}^j, \mathbf{P}_{k|k-1}^{yy,j}) \bar{q}_k^j, \quad \forall j \in [1, m], \quad (28)$$

and $p(\mathbf{y}_k | \mathbf{y}_{0:k-1})$ is a normalization constant. The mean $\hat{\mathbf{y}}_{k|k-1}^j$ and covariance prediction $\mathbf{P}_{k|k-1}^{yy,j}$ are determined by the corresponding LRF. The state estimate is

$$\hat{\bar{\mathbf{x}}}_{k|k} = \sum_{j=1}^m q_k^j \hat{\bar{\mathbf{x}}}_{k|k}^j, \quad (29a)$$

$$\mathbf{P}_{k|k} = \sum_{j=1}^m q_k^j \left(\mathbf{P}_{k|k}^j + (\hat{\bar{\mathbf{x}}}_{k|k}^j - \hat{\bar{\mathbf{x}}}_{k|k})(\star)^\top \right). \quad (29b)$$

Algorithm 1 summarizes the proposed method for adaptive sensor fusion of GNSS and camera measurements.

Algorithm 1 Pseudo-code of the proposed IMM algorithm

Initialize: $\{\xi^i, \omega^i\}_{i=1}^{|\mathcal{P}|}$, $\{\hat{\mathbf{x}}_{-1|-1}^j, \mathbf{P}_{-1|-1}^j, \mathbf{R}^j, q_{-1}^j\}_{j=1}^m$, Π

- 1: **for** $k = 0, 1, \dots$ **do**
- 2: **for** $i \in \{1, \dots, m\}$ **do**
- 3: Mix estimates according to (26).
- 4: **end for**
- 5: **for** $j \in \{1, \dots, m\}$ **do**
- 6: **for** $i \in \{1, \dots, |\mathcal{P}|\}$ **do**
- 7: Determine $\hat{\mathbf{x}}_{k|k-1}^i$ according to (20).
- 8: **end for**
- 9: Determine $\hat{\mathbf{x}}_{k|k-1}^j, \mathbf{P}_{k|k-1}^j$ according to (21a).
- 10: **for** $i \in \{1, \dots, |\mathcal{P}|\}$ **do**
- 11: Determine $\hat{\mathbf{y}}_{k|k-1}^i$ akin to (20)
- 12: **end for**
- 13: Determine $\hat{\mathbf{y}}_{k|k-1}^j, \mathbf{P}_{k|k-1}^{yy,j}, \mathbf{P}_{k|k-1}^{\bar{\mathbf{x}}y,j}$ akin to (21a).
- 14: Determine $\hat{\bar{\mathbf{x}}}_{k|k}^j, \mathbf{P}_{k|k}^j$ using (24).
- 15: Determine q_k^j according to (28).
- 16: **end for**
- 17: Determine $\hat{\bar{\mathbf{x}}}_{k|k}, \mathbf{P}_{k|k}$ according to (29).
- 18: **end for**

IV. SIMULATION STUDY

We validate the proposed method in a Monte-Carlo simulation study. We consider a car modeled by the dynamic single-track model (6) in closed loop with a reference tracking controller driving on a one-lane road. The route is extracted using the open-source routing machine (OSRM) tool [26]. The

map is represented by a sequence of points. To generate our spline-based map, we perform a rudimentary regression for $\bar{\gamma}$.

The IMU measurement noise is comparable to the noise for a low-cost IMU, and the GNSS position measurements nominally provide Gaussian zero-mean measurements with standard deviation 0.2m in both X and Y direction. Furthermore, the camera measurements provide lane measurements that nominally are Gaussian distributed according to $\mathbf{y}_k^c \sim \mathcal{N}(\mathbf{h}^c(\bar{\mathbf{x}}_k), \mathbf{R}_c)$, where \mathbf{R}_c is a diagonal matrix with elements $[0.02\mathbf{1}_2; 0.11\mathbf{8}]$. The transition probability matrix Π is set to have diagonal elements 0.9 with identical offdiagonal elements. From our experience, the design of the mixing matrix is not critical to performance.

When executing Algorithm 1 using the kinematic single-track model (8), only GNSS and camera measurements are used. For each of the 100 Monte-Carlo runs, we generate the initial state by sampling it from some initial 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. To simulate outliers, we consider six different models:

- 1) $\mathbf{R}^p = \mathbf{R}_{nom}^p, \mathbf{R}^c = 4^2 \mathbf{R}_{nom}^c$;
- 2) $\mathbf{R}^p = 10^2 \mathbf{R}_{nom}^p, \mathbf{R}^c = \mathbf{R}_{nom}^c$;
- 3) $\mathbf{R}^p = 10^2 \mathbf{R}_{nom}^p, \mathbf{R}^c = 2^2 \mathbf{R}_{nom}^c$;
- 4) $\mathbf{R}_{gps} = 10^2 \mathbf{R}_{gps,nom}, \mathbf{R}_c = 5^2 \mathbf{R}_{c,nom}$;
- 5) $\mathbf{R}_{gps} = 1^2 \mathbf{R}_{gps,nom}, \mathbf{R}_c = 5^2 \mathbf{R}_{c,nom}$;
- 6) $\mathbf{R}_{gps} = \mathbf{R}_{gps,nom}, \mathbf{R}_c = 7^2 \mathbf{R}_{c,nom}$;

where \mathbf{R}_{nom}^c denotes the nominal camera covariance matrix and \mathbf{R}_{nom}^p is the nominal GNSS covariance matrix. GNSS outliers occur every tenth second starting at 5s and lasting for three seconds, and to simulate GNSS outliers, we generate measurements according to $\mathbf{R}^p = 10^2 \mathbf{R}_{nom}^p$. Similarly, we have camera outliers occurring every tenth second starting at 10s and lasting for three seconds, with camera measurements generated according to $\mathbf{R}^c = 10^2 \mathbf{R}_{nom}^c$. Hence, the true model is not contained in the filter bank and consequently, the optimal output should be a combination of the filters in the IMM.²

A. Results for One Realization

For illustration purposes, we first present results using the dynamic single-track model from a single realization where the map prior uncertainty has a 0.05m standard deviation for each of the lane markings, over the entire simulation horizon. Fig. 4 displays snapshots of excerpts of the map at different time instants. As measurements are gathered, the map estimate is improved. Furthermore, the uncertainty in the map is decreased. In particular, note the decrease of the uncertainty and associated correction of the map ahead of the vehicle due to the forward-looking camera.

²For a perfect map, i.e., for the localization problem, we have already verified that the method finds the correct model [6] so in this paper we focus on other aspects.

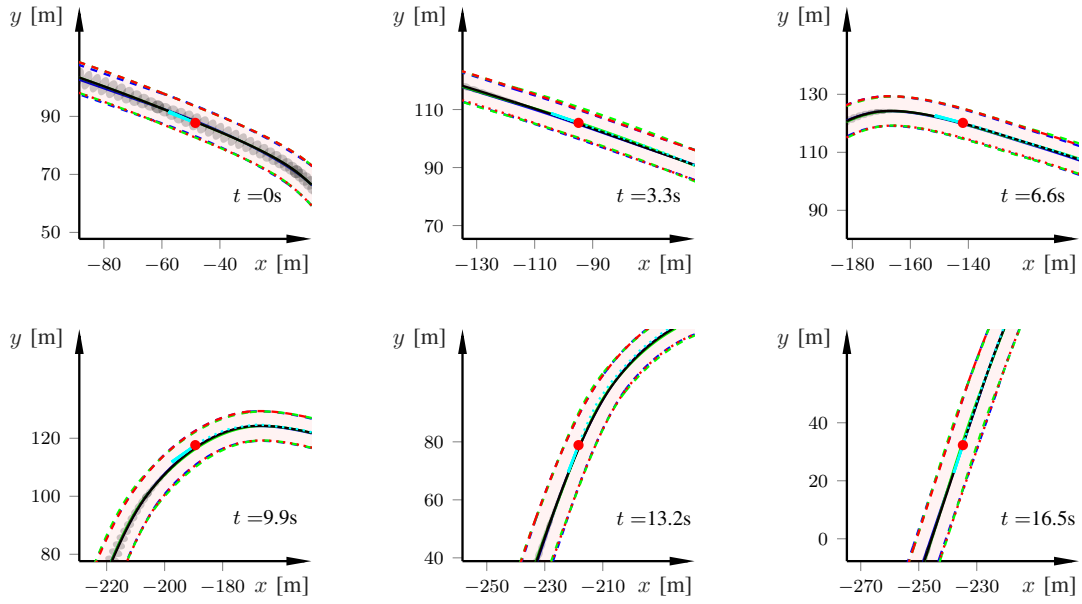


Fig. 4. Snapshots of the map for different time steps including the true map in blue, the prior mean map (green), and the posterior in red with the map uncertainty illustrated as the shaded circles. Note that the uncertainty is magnified for visualization purposes.

B. 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. IV-A. We compare Algorithm 1 with our previously proposed IMM-LRKf [6], which assumes a perfect map. We have executed the methods for two different map priors: (i), zero standard deviation on the spline coefficients (i.e., a perfect map); (ii) a covariance on the spline coefficients corresponding to a 0.05m standard deviation. For (ii), at the beginning of each Monte-Carlo run, we sample the true and prior maps for each LRKF in the IMM according to this distribution.

Fig. 5 shows the X and Y RMSE in the vehicle frame O_E for the different models in the IMM, together with the estimated output of Algorithm 1 (Line 17). The effects of outliers are seen by the sudden increase in positioning error for the different models. The position estimate of the weighted average shows the smallest position error for most of the time except for some transients, that is, it correctly weighs together the different estimators in the IMM, irrespective which motion model that is used for estimation. This is clearly seen between 5–8s when there are GNSS outliers, when the models having small GNSS measurement covariance gives an estimation error of more than 2m in the X -direction. The IMM-LRKf discards those models when producing the estimates. The lateral error fluctuates less and is generally smaller than the longitudinal positioning error, which is expected since the camera has higher resolution in that direction. Furthermore, the RMSEs exhibit larger variations when using the kinematic single-track model. This is not surprising as the synthetic data are generated using the dynamic single-track model, and to account for the model mismatch the process noise has to be increased when using the kinematic single-track model.

Fig. 6 compares the results when updating the map as in Algorithm 1 and when not performing the map update, as in

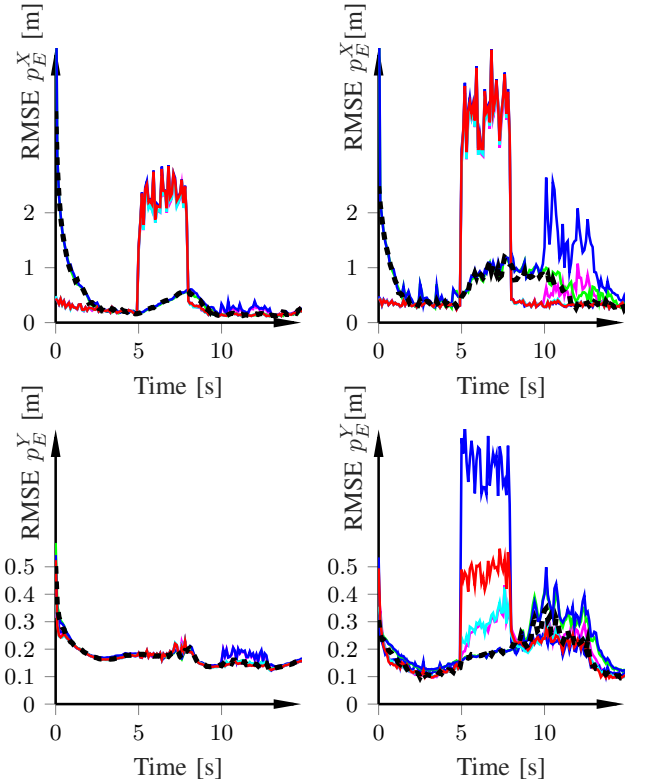


Fig. 5. RMSE of the seven (six models plus weighted average) compared combinations for 100 Monte-Carlo runs with 0.05m initial map covariance using the dynamic single-track model (left) and the kinematic single-track model (right) in the estimator. The estimator output is in black dashed. The estimator chooses the best available filter except for during the transients.

[6]. Fig. 6 displays the resulting RMSE when the prior map is in perfect alignment with the true map, and with the 0.05m initial uncertainty. Clearly, incorporating the map update into the estimation problem improves performance.

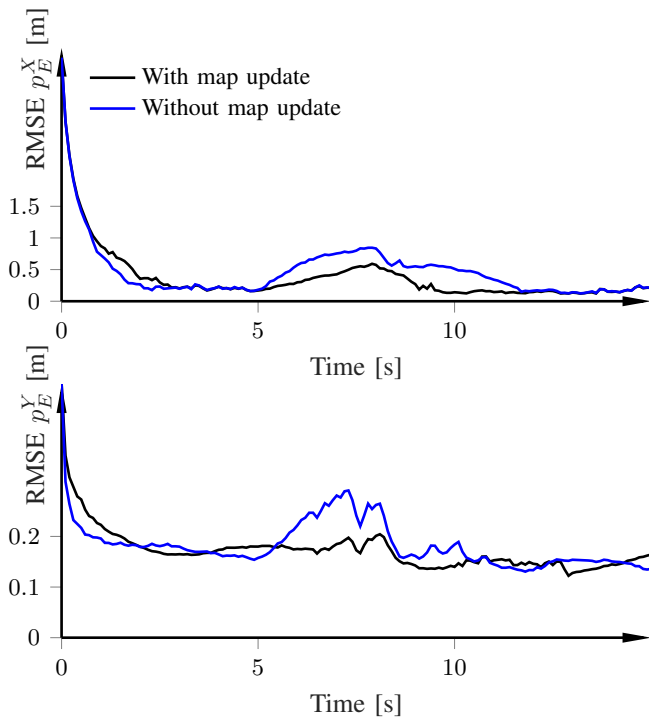


Fig. 6. RMSE with map update (black) and without map update (blue, our previous work [6]).

V. CONCLUSION

By using the IMM framework we can adapt to the time-varying measurement reliability for both the GNSS and camera measurements, and the simulation results indicate that our method indeed chooses the model combination yielding the smallest error. As long as the support of the noise covariances in the models covers the true noise, the proposed method is a viable alternative to other noise-adaptive filters such as particle filters or filters based on variational Bayes. Unsurprisingly, when using the dynamic single-track model, slightly more reliable estimates are provided. However, in practice the dynamic single-track model needs knowledge of parameters that are uncertain and also needs to be estimated online. Hence, using the kinematic single-track model is a sub-optimal but attractive option due to its simplicity. The spline-based map representation is highly flexible and more realistic than the traditional curvature-based approaches, since it does not restrict itself to certain types of roads. In future work, we plan to experimentally verify the proposed method.

REFERENCES

- [1] R. Karlsson and F. Gustafsson, "The future of automotive localization algorithms: Available, reliable, and scalable localization: Anywhere and anytime," *IEEE Signal. Process. Mag.*, vol. 34, no. 2, pp. 60–69, 2017.
- [2] E. D. Dickmanns, *Dynamic Vision for Perception and Control of Motion*. Berlin, Heidelberg: Springer-Verlag, 2007.
- [3] P. Teunissen, "A canonical theory for short GPS baselines. Part III: the geometry of the ambiguity search space," *J. Geodesy*, vol. 71, no. 8, pp. 486–501, 1997.
- [4] M. Sahnoudi 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] J. Steinbring and U. Hanebeck, "LRKF revisited—the smart sampling Kalman filter (S2KF)," *J. Adv. Information Fusion*, vol. 9, no. 2, pp. 106–123, 2014.
- [6] 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.
- [7] 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.
- [8] C. Lundquist and T. B. Schön, "Joint ego-motion and road geometry estimation," *Information Fusion*, vol. 12, no. 4, pp. 253–263, 2011.
- [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] R. Rajamani, *Vehicle Dynamics and Control*. Springer-Verlag, 2006.
- [11] K. Berntorp, "Joint wheel-slip and vehicle-motion estimation based on inertial, GPS, and wheel-speed sensors," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 3, pp. 1020–1027, 2016.
- [12] C. Lundquist, R. Karlsson, E. Özkan, and F. Gustafsson, "Tire radii estimation using a marginalized particle filter," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 663–672, 2014.
- [13] W. Wen, X. Bai, G. Zhang, S. Chen, F. Yuan, and L.-T. Hsu, "Multi-agent collaborative GNSS/camera/ins integration aided by inter-ranging for vehicular navigation in urban areas," *IEEE Access*, 2020.
- [14] A. Carvalho, S. Lefèvre, G. Schildbach, J. Kong, and F. Borrelli, "Automated driving: The role of forecasts and uncertainty - a control perspective," *Eur. J. Control*, vol. 24, pp. 14–32, 2015.
- [15] K. Berntorp and S. Di Cairano, "Tire-stiffness and vehicle-state estimation based on noise-adaptive particle filtering," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 1100–1114, 2018.
- [16] E. Schindler, *Fahrdynamik: Grundlagen Des Lenkverhaltens Und Ihre Anwendung Für Fahrzeugregelsysteme*. Renningen, Germany: Expert-Verlag, 2007.
- [17] "Vägutformning 94 version s-2," Swedish National Road Administration, Tech. Rep., 1994.
- [18] 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.
- [19] M. Greiff and K. Berntorp, "Optimal measurement projections with adaptive mixture Kalman filtering for GNSS positioning," in *Amer. Control Conf.*, 2020.
- [20] 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.
- [21] K. Berntorp and S. Di Cairano, "Noise-statistics learning of automotive-grade sensors using adaptive marginalized particle filtering," *J. Dynamic Syst., Measurement, and Control*, vol. 141, no. 6, pp. 061 009–10, 2019.
- [22] F. Gustafsson, *Statistical Sensor Fusion*. Lund, Sweden: Utbildningshuset/Studentlitteratur, 2010.
- [23] J. Steinbring and U. D. Hanebeck, "S 2 KF: The smart sampling Kalman filter," in *Int. Conf. Information Fusion*, Istanbul, Turkey, Jul. 2013.
- [24] I. Arasaratnam, "Cubature Kalman filtering theory & applications," Ph.D. dissertation, McMaster University, 2009.
- [25] H. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Autom. Control*, 1988.
- [26] OSRM, "Homepage of the open source road map project," last accessed 08-19-2021. [Online]. Available: <http://project-osrm.org/>