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

Distributed Road-Map Monitoring Using Onboard Sensors

Zhang, Yanyu; Greiff, Marcus; Ren, Wei; Berntorp, Karl

TR2024-093 July 09, 2024

Abstract

Road maps for vehicle control and navigation systems are typically generated by mapping systems that are highly accurate but updated infrequently. However, changes to the roads are made at a higher frequency. Stored road maps may therefore not capture the true road well. To resolve this, we consider online road-map estimation using the type of sensors found in production cars. The map estimation for a given vehicle is based on a global positioning system, camera, steering wheel, and wheel-speed sensors. As each vehicle covers a limited amount of road, we leverage crowdsourced map estimates from multiple vehicles to get a more complete representation of the road map. High-fidelity simulation results indicate a reduction of the estimation error of roughly 15% when using 5 agents compared to the best single agent. Furthermore, we show that the method is capable of updating map segments that have large errors, for example, as may occur during road maintenance.

American Control Conference (ACC) 2024

© 2024 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.

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

Distributed Road-Map Monitoring Using Onboard Sensors

Yanyu Zhang, Marcus Greiff, Wei Ren, Karl Berntorp*

***Abstract*— Road maps for vehicle control and navigation systems are typically generated by mapping systems that are highly accurate but updated infrequently. However, changes to the roads are made at a higher frequency. Stored road maps may therefore not capture the true road well. To resolve this, we consider online road-map estimation using the type of sensors found in production cars. The map estimation for a given vehicle is based on a global positioning system, camera, steering wheel, and wheel-speed sensors. As each vehicle covers a limited amount of road, we leverage crowdsourced map estimates from multiple vehicles to get a more complete representation of the road map. High-fidelity simulation results indicate a reduction of the estimation error of roughly 15% when using 5 agents compared to the best single agent. Furthermore, we show that the method is capable of updating map segments that have large errors, for example, as may occur during road maintenance.**

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 vehicle control.

Road-vehicle positioning is preferably approached by fusion of multiple complementary sensor modalities, because it adds robustness and redundancy [1]. By leveraging prior road maps, for example, from a mobile mapping system (MMS), reliability and accuracy can be improved because it enables positioning relative to a global map. Maps from MMSs are updated infrequently, while higher-frequency changes to the map (e.g., road construction, lane marking 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), radar, and road maps 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], [4], 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).

In [5], we developed a GNSS-based sensor fusion method for joint (single) vehicle positioning and road-map estimation, with the map represented by one or several splines. 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.

Vehicle connectivity and crowdsourcing offer new possibilities for storing and processing data (e.g., [6]). This paper presents a map estimation method using crowdsourced data, for example, by a cloud storage as well as driver-specific data. The method relies on (i) local position-specific map estimates obtained from multiple vehicles and (ii) global map estimates obtained by merging the local map estimates of multiple estimates and a prior map. Locally, a vehicle estimates both its vehicle position and its surrounding map simultaneously, which are subsequently crowdsourced. Globally, the local estimates are fused together to provide an updated global map. The method assumes that the local estimator executed in the vehicles is able to output the first two moments (i.e., mean and covariance) of an estimated distribution. We envision a global map-monitoring solution that can provide more frequent map corrections than what an MMS is capable of. This can have implications related to route planning and high-level navigation systems. Optionally, the global map corrections can be transmitted back to the local vehicles, which can have implications for lower-level control such as motion planning and ADASs. However, we focus on data fusion in this paper, implying that we fuse the estimates globally without updating the internal memory of the local estimators.

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 $\boldsymbol{\Sigma}$. The associated probability density function is denoted by $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Here, $[\boldsymbol{x}]_i$ indicates the ith element of the vector $\boldsymbol{x}$, and matrices are written in capital bold font as $\boldsymbol{X}$ with elements 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 $\boldsymbol{x}_k$ given $\boldsymbol{y}_{0:k}$ from time step 0 to time step k. The concatenation of two vectors $\boldsymbol{x} \in \mathbb{R}^{n_{\boldsymbol{x}}}$ and $\boldsymbol{y} \in \mathbb{R}^{n_{\boldsymbol{y}}}$ is $[\boldsymbol{x}; \boldsymbol{y}] = [\boldsymbol{x}^{\top}, \boldsymbol{y}^{\top}]^{\top} \in \mathbb{R}^{n_{\boldsymbol{x}}+n_{\boldsymbol{y}}}$. Furthermore, $\boldsymbol{I}_n$ is the $n \times n$ identity matrix. We let $f \in \mathcal{C}^n(A, B)$ denote functions $f : A \mapsto B$ whose first n derivatives are continuous. Finally, $\mathrm{vec}(\mathcal{S})$ denotes a vector representation of an arbitrary collection of points $\mathcal{S}$, $\mathrm{vec}(\cdot)$ is the vectorization operator, and $\mathrm{blkdiag}(\boldsymbol{A}, \boldsymbol{B})$ generates a block-diagonal matrix of the inputs $\boldsymbol{A}$ and $\boldsymbol{B}$.

The work of Y. Zhang was done while he was working at MERL.

M. Greiff, K. Berntorp* are with Mitsubishi Electric Research Labs (MERL), Cambridge, MA, 02139, USA. Email: karl.o.berntorp@ieee.org

Y. Zhang, and W. Ren are with the Department of Electrical and Computer Engineering, University of California, Riverside, CA, 92521, USA.

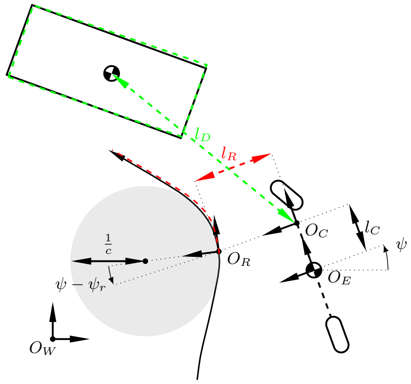

Fig. 1. The relation between the vehicle frame O_E, the camera frame O_C, the road frame of the left lane, $O_{R,l}$, and the world frame O_W. The distance between the vehicle's longitudinal X-axis and the left lane boundary is l_L, and the shaded circle depicts the road curvature (here exaggerated)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

II. Vehicle and Sensor Modeling

Fig. 1 shows the different coordinate frames used in this paper. 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. In [7], 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. Hence, we consider a kinematic model but our method straightforwardly extends to more complicated vehicle models.

The kinematic single-track model has 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,

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

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 process noise, $\boldsymbol{w}_k^{\boldsymbol{z}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}^{\boldsymbol{z}})$, and $\boldsymbol{u}_k$ denotes the control input at time step k.

A. Road Model

There are multiple ways in which a road map can be represented. Several previous works have employed a clothoidal representation [3], [4]. However, 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 such that

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

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

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

The problem with this representation is that the lane boundaries need not be continuous for an integer s unless we impose constraints on $\bar{\boldsymbol{\gamma}}$. 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

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

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 $\boldsymbol{\gamma} = [\boldsymbol{\gamma}_1; ...; \boldsymbol{\gamma}_M]$ and introduce uncertainty in the map by assigning a Gaussian prior on each $\boldsymbol{\gamma}_m$, as $\gamma_{m,0} \sim \mathcal{N}(\boldsymbol{m}^{\gamma}_{m,0}, \boldsymbol{\Sigma}^{\gamma}_{m,0})$ 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,

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

In summary, the prediction model (1) and (7) is

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

where $\boldsymbol{w} = [\boldsymbol{w}^{\boldsymbol{z}}; \boldsymbol{w}^{\boldsymbol{\gamma}}]$ and $\boldsymbol{Q} = \mathrm{blkdiag}(\boldsymbol{Q}^{\boldsymbol{z}}, \boldsymbol{Q}^{\boldsymbol{\gamma}}) \succ \boldsymbol{0}$.

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 [8]–[10]. 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 considered to be 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, we let $\boldsymbol{y}^p \in \mathbb{R}^2$.

The camera and a computer-vision (CV) algorithm provide measurements of the road geometry and the relative vehicle position. 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, see Fig. 1.

To use the polynomial approximation for inference, the measurement equation needs discrete values. Hence, we sample the polynomials uniformly at n_s points over their domain defined in s, $\{s^i_L, s^i_R\}^{n_s}_{i=1}$. 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.

Remark 2 *As 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 observable.*

III. Distributed Map Updates by Estimate Fusion

We propose a distributed approach to road mapping where to increase the map accuracy we fuse the mean and covariance of the GEP map parameters from different vehicles that independently estimate segments of the same map. Each vehicle estimates the vehicle state $\boldsymbol{z}$ and map parameters $\boldsymbol{\gamma}$ using the IMM-LRKF with GEPs developed in [5], [7], which Sec. III-A briefly summarizes.

A. Joint Vehicle State and Map Estimation for Single Vehicle

The estimation model consisting of (2), (8), and (10) 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 and therefore have to be numerically approximated if used.[1] We therefore consider LRKFs, which we embed in an IMM framework.

1) Linear-Regression Kalman Filter: For each LRKF, we approximate the posterior density by its first two moments,

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

Given (11) at time step k, the state-prediction distribution at time step $k+1$ is approximated by

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

by direct evaluation of the associated moment integrals. Using the LRKF framework [11], we transform the coordinates $\boldsymbol{\xi}_k = \boldsymbol{L}^{-1}_{k|k}(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k})$ using the Cholesky factorization of the covariance matrix $\boldsymbol{P}_{k|k} = \boldsymbol{L}_{k|k}\boldsymbol{L}^\top_{k|k}$, and approximate the transformed integrals by evaluating the nonlinearity (8) in a set of integration points $\{(\omega^i, \boldsymbol{\xi}^i)\}^{N_{\mathrm{int}}}_{i=1}$, as

$$\hat{\boldsymbol{x}}^i_{k+1|k} = \boldsymbol{f}\left(\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{L}_{k|k}\boldsymbol{\xi}^i\right), \tag{13}$$

leading to the predicted sufficient statistics as the weighted average of the mean and covariance. For the measurement update, the joint density is approximated using the same integration techniques, resulting in a similar expression for the measurement update (see [5]) and associated expressions $\hat{\boldsymbol{y}}$, $\boldsymbol{P}^{\boldsymbol{xy}}$, and $\boldsymbol{P}^{\boldsymbol{y}}$. This leads to the usual Kalman filter update, where the update is done with respect to the vehicle state and the map parameters associated with the Bézier curves currently being measured by the camera.

2) Interacting Multiple-Model LRKF: The reliability of both the GNSS measurements and camera-based measurements vary in time. We therefore implement the LRKF in an IMM framework [12], [13], where we have a set of N_{mod} 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,

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k, \qquad \boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k), \tag{14a}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{e}_k(\theta_k), \;\; \boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}^{\theta_k}_k), \tag{14b}$$

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

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

[1] For instance, given (p^X, p^Y, ψ) and $\bar{\boldsymbol{\gamma}}$, 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 $O_{R,l}$ in the global frame, before evaluating l_L. As such, l_L is a function of $\bar{\boldsymbol{x}}$, but this function is not differentiable.

Next, the IMM runs a filter bank of N_{mod} LRKFs to find the estimate of $\boldsymbol{x}_k$, where the jth LRKF executes using $\boldsymbol{R}^j$. The state posterior is expressed using the law of total probability as a Gaussian mixture of N_{mod} components,

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k}) \approx \sum_{j=1}^{N_{\text{mod}}} q_k^j \mathcal{N}(\boldsymbol{x}_k^j|\hat{\boldsymbol{x}}^j_{k|k-1}, \boldsymbol{P}^j_{k|k-1}), \tag{15}$$

where

$$q_k^j \propto \mathcal{N}\Big(\boldsymbol{y}_k|\hat{\boldsymbol{y}}^j_{k|k-1}, \boldsymbol{P}^{\boldsymbol{y},j}_{k|k-1}\Big)\bar{q}_k^j, \ \ \forall j \in [1, N_{\text{mod}}]. \tag{16}$$

The mean $\hat{\boldsymbol{y}}^j_{k|k-1}$ and covariance $\boldsymbol{P}^{\boldsymbol{y},j}_{k|k-1}$ are determined by the corresponding LRKF. The state estimate is the weighted average of the involved LRKFS.

B. Distributed Map Fusion

There are myriad methods that may be used for decentralized fusion of the local road maps when represented in the implicitly continuous form with Gaussian distributions over the map parameters. This includes covariance intersection (CI) [14], inverse CI (ICI) [15], ellipsoidal intersection (EI) [16], and many others. In this paper, we primarily consider ICI, as the method comes with proofs of consistency and is generally less conservative than CI [14]. We denote an estimate of the map in the ith vehicle as $\hat{\gamma}^i_{k|k} \sim \mathcal{N}(\boldsymbol{m}^i_k, \boldsymbol{\Sigma}^i_k)$, which can be extracted from the corresponding state estimate. The fusion of N such PDFs is denoted by

$$\mathcal{N}(\boldsymbol{m}^f_k, \boldsymbol{\Sigma}^f_k) \leftarrow \texttt{fuse}(\{\mathcal{N}(\boldsymbol{m}^i_k, \boldsymbol{\Sigma}^i_k)\}_{i=1}^N). \tag{17}$$

While methods such as CI can be formulated in a central manner, that is, fusing all of the estimates at once, ICI cannot. Consequently, to make use of ICI and facilitate a distributed implementation, (17) is implemented in a nested manner, operating on pairs of information items in the spirit of [15]. Furthermore, we do so independently over the generalized endpoints in the implicitly continuous map representation, as a block-structured covariance matrix is explicitly enforced in the IMM-LRKF implementation [5]. For a map with M endpoints, we let $\mathcal{N}(\boldsymbol{m}^i_{m,k}, \boldsymbol{\Sigma}^i_{m,k})$ denote the estimate associated with the mth endpoint in the ith vehicle, and we apply the fusion method outlined in Algorithm 1.

Algorithm 1 Fusion by pairwise application of a fusion rule.

```
$\mathcal{N}(\boldsymbol{m}^f_k, \boldsymbol{\Sigma}^f_k) \leftarrow \mathcal{N}(\boldsymbol{m}^1_k, \boldsymbol{\Sigma}^1_k)$
for $i = 2, \dots, N$ do
    for $m = 1, \dots, M$ do
        $\mathcal{N}(\boldsymbol{m}^f_{m,k}, \boldsymbol{\Sigma}^f_{m,k}) \leftarrow$
            $\texttt{fuse}(\{\mathcal{N}(\boldsymbol{m}^i_{m,k}, \boldsymbol{\Sigma}^i_{m,k}), \mathcal{N}(\boldsymbol{m}^f_{m,k}, \boldsymbol{\Sigma}^f_{m,k})\})$
    end for
end for
return $\mathcal{N}(\boldsymbol{m}^f_k, \boldsymbol{\Sigma}^f_k)$
```

Algorithm 1 can be implemented in a distributed manner and effectively implies that $N-1$ of the vehicles receive a fused PDF, apply the fuse operation with respect to its own estimate, before passing along the resulting PDF to the next vehicle. In the following, we implement ICI for the fuse operation of two densities $\mathcal{N}(\boldsymbol{m}^1_k, \boldsymbol{\Sigma}^1_k)$ and $\mathcal{N}(\boldsymbol{m}^2_k, \boldsymbol{\Sigma}^2_k)$

$$\boldsymbol{M}_k = (w_1\boldsymbol{\Sigma}^1_k + w_2\boldsymbol{\Sigma}^2_k)^{-1}, \tag{18a}$$

$$(\boldsymbol{\Sigma}^f_k)^{-1} = (\boldsymbol{\Sigma}^1_k)^{-1} + (\boldsymbol{\Sigma}^2_k)^{-1} - \boldsymbol{M}_k, \tag{18b}$$

$$\boldsymbol{m}^f_k = \boldsymbol{\Sigma}^f_k\Big(\sum_{i=1}^{2}[(\boldsymbol{\Sigma}^i_k)^{-1} - w_i\boldsymbol{M}_k]\boldsymbol{m}^i_k\Big), \tag{18c}$$

where the weights $w_1, w_2 > 0$ sum to 1.

IV. Simulation Study

In this section, we present results from a simulation study utilizing CarSim [17] and its sensor package.

In the following, we consider a five-lane highway, with 14 vehicles in different lanes starting from different initial positions. We use the Cubature rule for assigning integration points in the LRKFs [18] the common weighting heuristic

$$w_i = \text{Tr}(\boldsymbol{\Sigma}^i_k)^{-1}/(\text{Tr}(\boldsymbol{\Sigma}^1_k)^{-1} + \text{Tr}(\boldsymbol{\Sigma}^2_k)^{-1}), \ \ i = 1, 2 \tag{19}$$

for the ICI, which, in the context of Algorithm 1, is implemented such that different weights are computed for the fusion of different generalized endpoints in the map. The sensor measurements, that is, camera and GNSS, are logged for five of the vehicles and the estimation is done offline. The exact models used in CarSim [17] are of higher fidelity than those assumed in Sec. II. The true map used in CarSim is sampled in a set of points, corresponding to the output of an MMS, and the implicitly continuous map is regressed to this data (see Remark 1). We regress two maps; one with very little noise associated with the HD map represented as $\boldsymbol{\gamma}^t \sim \mathcal{N}(\boldsymbol{m}^t, \boldsymbol{\Sigma}^t)$, which serves as ground truth ($\sigma = 1$cm), and one with greater variance associated with the HD map ($\sigma = 10$cm), which serves as a prior for the road-map estimation, $\hat{\gamma}_{0|0} \sim \mathcal{N}(\boldsymbol{m}_0, \boldsymbol{\Sigma}_0)$, common for all cars. Hence, when assessing the estimation performance, we compare the root mean-square error (RMSE) of the map parameters of an estimate $\hat{\gamma}^i_{k|k} \sim \mathcal{N}(\boldsymbol{m}^i_k, \boldsymbol{\Sigma}^i_k)$ and the true map $\boldsymbol{\gamma}^t$. We do this in a set of N_{MC} Monte-Carlo (MC) simulations, where the super-index $(\cdot)^{(n)}$ denotes the nth simulation.

A. Simulation With Nominal Prior Map

For the first simulation, five vehicles are initialized from different positions and different lanes on the same map. Each vehicle runs the IMM-LRKF described in Sec. III and communicates its estimate to a neighbor (or a cloud) in real-time. Specifically, the first two moments of the map parameter distribution estimated in each vehicle is communicated, and a fused estimate is computed using (18) and the weight rule defined in (19) in a distributed manner (see Algorithm 1).

In each simulation, the measurements are generated in CarSim, but additional noise is added to simulate outlier events. Specifically, noise is added to the GNSS measurements with covariance $\boldsymbol{R}^p$ and to the camera measurements with covariance $\boldsymbol{R}^c = \text{diag}(\boldsymbol{R}^c_{\text{lane}}, \boldsymbol{R}^c_{\text{poly}})$, distinguishing between the orthogonal lane measurements and the measurements of the polynomial approximation of the lane boundary ahead. The IMM uses seven models for estimation.

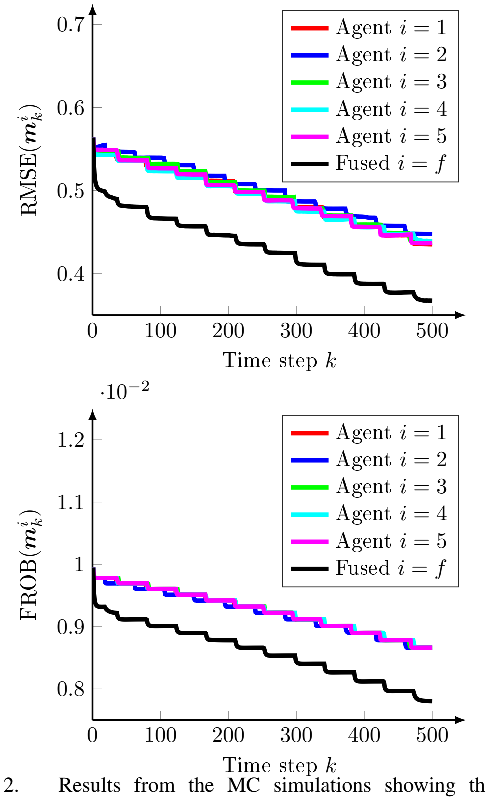

Fig. 2. Results from the MC simulations showing the RMSE and Frobenious error of the map parameters estimated locally in each vehicle $i = 1, \ldots, 5$ as well as the fused map (black).

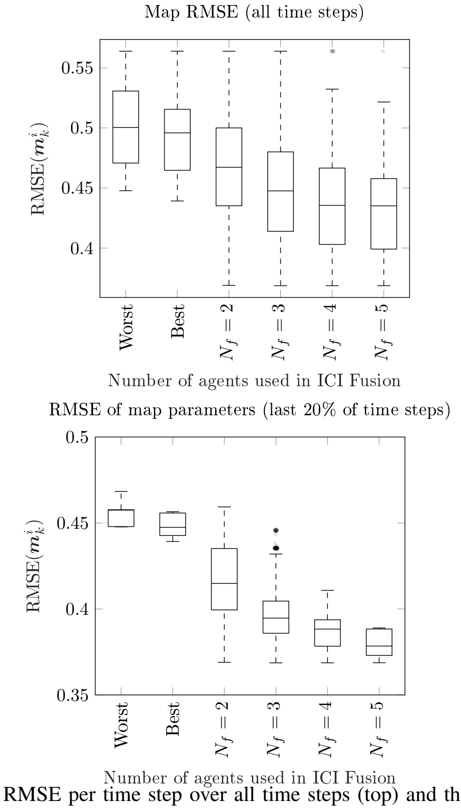

Fig. 3. RMSE per time step over all time steps (top) and the last 20% of time steps (bottom).

For every 300m, there is 240m of road without outliers, 30m of camera outliers and another 30m of GNSS outliers. These segments are fixed to specific intervals in the path variable s of the map, corresponding to a tunnel or lack of lane markings that therefore equally affect all vehicles passing a specific portion of the map. Note that none of these simulation models exists in the set of estimation models defining the IMM-LRKFs, and that we generally model slightly higher measurement noises in the filters. This is because the measurements generated by CarSim include additional errors than the the simulation models.

Fig. 2 compares the resulting estimated map for each vehicle with the fused estimates in terms of RMSE. In the first subplot, we note a quick initial decrease in the RMSE, which is due to the vehicles residing in different spatial locations. Thus, the map is rapidly improved in their respective local surroundings. Both the RMSE and the Frobenious error also decrease with time, indicating that we find a map that is indeed closer to the true map used in CarSim. There is a significant improvement in using multiple vehicles and the fusion rule in (18) with the weight update in (19), compared with each individual vehicle.

B. Map Evaluation with Different Number of Vehicles

To further highlight the benefits of cooperative mapping, we explore how the RMSEs of the fused estimates depend on the number of vehicles included in the fusion. Specifically, we study the RMSE over some time interval $k \in [k_{\min}, k_{\max}]$. Fig. 3 shows the results for the worst vehicle (in expectation) and the best vehicle (in expectation) along with the fused results. Again, we consider a maximum of $N = 5$ vehicles, which means that there are $\binom{N}{N_f} \in \{10, 10, 5, 1\}$ unique ways of choosing $N_f \in \{2, 3, 4, 5\}$ combinations of vehicles from the set of all vehicles. In the MC study, which includes $N_{MC} = 50$ simulations, the statistics are aggregated over all these combinations; that is, for the case $N_f = 3$, we use 10×50 estimates per time step to compute the RMSE.

In this study, we observe a significant improvement in RMSE of the map parameters as a function of the number of vehicles used in the fusion, but diminishing returns as the number of vehicles increases. When considering all of the time steps, the difference in using one or many vehicles is clear but less pronounced, as we are effectively considering all of the time steps at once in the previous example. A vast majority of these estimation errors will simply be those induced by the prior. More interestingly, when considering the last 20% of the time steps, there is a clear improvement in RMSE of the map parameters, both with respect to the best and the worst performing single vehicle.

C. Simulation with Erroneous Prior Map

It is relevant to analyze what effects systematic errors in the map prior have on the estimation performance, and if the proposed method can adapt when using information from multiple vehicles. To this end, we consider a simpler setting with a single-lane road where an IMM-LRKF is run over a stretch of $K = 400$ time steps. In total, we let 20 vehicles pass the same portion of the map. After each simulation, the terminal map estimate of the most recent simulation is used as the prior map in the next. Here, we introduce systematic

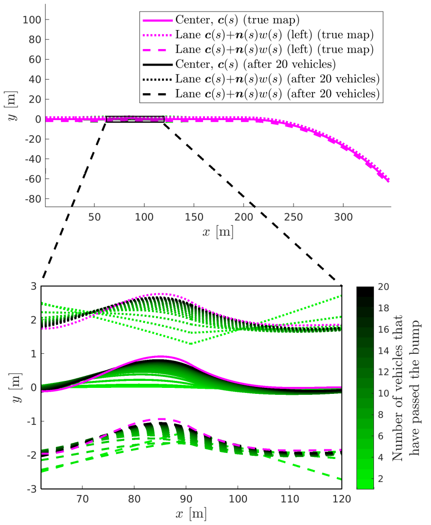

Fig. 4. Representation of the maps as parametric curves corresponding to the center of the lane $\boldsymbol{c}(s)$ and the left and right lane boundaries. A subsection of the map (top) and a zoom on the location where the errors are introduced (bottom). The true map $\boldsymbol{\gamma}^t$ (magenta) differs significantly from the map prior $\hat{\gamma}_{0|0}$ (green) but when driving a large number of cars over the bump (here 20), the resulting map evaluated in $\hat{\gamma}_{K|K}$ (black) approaches the true map (magenta).

errors by modifying the true map $\boldsymbol{\gamma}^t \sim \mathcal{N}(\boldsymbol{m}^t, \boldsymbol{\Sigma}^t)$, such that the resulting center lane deviates from the center lane computed with the filter prior by as much as 1m laterally. This error is introduced in the MMS points prior to a regression in line with Remark 1. In addition to this, we introduce systematic errors in the lane width of 2m in two consecutive GEPs,

$$[\hat{\gamma}_{0|0}]_{15} \leftarrow [\hat{\gamma}_{0|0}]_{15} - 2, \tag{20a}$$

$$[\hat{\gamma}_{0|0}]_{20} \leftarrow [\hat{\gamma}_{0|0}]_{20} + 2. \tag{20b}$$

Due to these substantial errors, we introduce additional process noise in the map. For illustration purposes, we inflate the map prior covariance slightly prior to each simulation. Note that the same trends hold even if this inflation is not done, but more simulations are required.

Fig. 4 illustrates the result, with the true map (magenta); the prior of the first vehicle (green); and the estimate of the 20th vehicle (black). The bump in the true map is clearly visible in the center lane (full, green vs magenta), and the modifications to the lane width prior in (20) are visible in the lane boundaries (dashed/dotted, green vs magenta). This is a significant and systematic error in the map estimates, corresponding to scenarios such as sudden changes due to road construction. Despite these large errors, the algorithm converges to a map that is reasonably close to the true map after 20 vehicles have passed the portion of the map. This highlights the efficacy of the proposed method to poorly specified MMS maps, and demonstrates that the IMM-LRKF with the proposed map representation is robust.

V. Conclusions

We proposed a distributed localization and road mapping framework that utilizes crowdsourced data. We provided a detailed formulation of a road model constructed with Bézier curves in an implicitly continuous GEP representation, explained how this is amenable to IMM-LRKF estimation, and showed how the resulting local estimates can be used for distributed map fusion. The Monte-Carlo analysis implies that our algorithm not only achieves higher mapping accuracy under normal noise levels but also effectively handles large road modifications caused by external effects, such as road maintenance. Furthermore, the impact of the number of agents on map improvement shows a 14% reduction of RMSE using 5 agents.

References

[1] F. Gustafsson, "Automotive safety systems," *IEEE Signal Processing magazine*, 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] 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.

[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. Menner, Z. Ma, K. Berntorp, and S. Di Cairano, "Location and driver-specific vehicle adaptation using crowdsourced data," in *Eur. Control Conf.*, Jul. 2022.

[7] 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.

[8] 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.

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

[10] 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.

[11] J. Steinbring and U. D. Hanebeck, "S 2 KF: The smart sampling Kalman filter," in *Int. Conf. Information Fusion*, Istanbul, Turkey, Jul. 2013.

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

[13] F. Gustafsson, *Statistical Sensor Fusion*. Lund, Sweden: Utbildningshuset/Studentlitteratur, 2010.

[14] L. Chen, P. Arambel, and R. Mehra, "Estimation under unknown correlation: covariance intersection revisited," *IEEE Transactions on Automatic Control*, vol. 47, no. 11, pp. 1879–1882, 2002.

[15] B. Noack, J. Sijs, and U. D. Hanebeck, "Inverse covariance intersection: New insights and properties," in *2017 20th International Conference on Information Fusion (Fusion)*, 2017, pp. 1–8.

[16] ——, "Algebraic analysis of data fusion with ellipsoidal intersection," in *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2016, pp. 365–370.

[17] "CarSim," Mechanical Simulation version 2021.0. [Online]. Available: https://www.carsim.com,

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