

Road Geometry Estimation Using a Precise Clothoid Road Model and Observations of Moving Vehicles

Maryam Fatemi, Lars Hammarstrand, Lennart Svensson, Ángel F. García-Fernández*

Abstract—An important part of any advanced driver assistance system is road geometry estimation. In this paper, we develop a Bayesian estimation algorithm using lane marking measurements received from a camera and measurements of the leading vehicles received from a radar-camera fusion system, to estimate the road up to 200 meters ahead in highway scenarios. The filtering algorithm uses a segmented clothoid-based road model. In order to use the heading of leading vehicles we need to detect if each vehicle is keeping lane or changing lane. Hence, we propose to jointly detect the motion state of the leading vehicles and estimate the road geometry using a multiple model filter. Finally the proposed algorithm is compared to an existing method using real data collected from highways. The results indicate that it provides a more accurate road estimation in some scenarios.

I. INTRODUCTION

Nowadays, active safety and autonomous driving systems provide vehicles with advanced functionalities to warn or assist the driver in dangerous situations. Many of the functions within these systems can benefit from accurate road geometry information. Since map data is either not accurate enough or not available for some roads, this information needs to be extracted from noisy observations provided by the on-board sensors. Two useful sources of information are measurements of the lane markings and the moving vehicles.

Road geometry estimation is a well studied subject, examples of which can be found in [1], [2] and [3]. Most of these papers focus on near range road geometry estimation which is enough for some applications. However, in highway scenarios where the host vehicle has a high speed, accurate long distance information can be crucial.

Using the lane marking measurements to estimate the road has been reported in several papers including [4] and [5]. In [1], [6] and [7] the authors use the measurements of moving vehicles to update the road state. The measurements of leading vehicles can be misleading when the vehicle is changing lane/taking an exit. Therefore, in order to use these measurements correctly, one alternative is to design a mechanism that can help us distinguish between keeping lane and changing lane scenarios.

Measurements of the leading vehicles are used to jointly improve the estimation of the road geometry and host vehicle's state in [1], where the algorithm only assumes that these vehicles follow their lane. Not accounting for lane changes

could explain why according to the authors, their algorithm performs best when driving alone. In [6] the authors use the lateral movement of the vehicles to update the road state and they propose using a detection algorithm to distinguish between vehicles which are keeping lane and those which are not. However, according to the authors the detection algorithm suffers from a high false alarm rate. In [7], the heading of the leading vehicles is used and to avoid updating with vehicles that are changing lane/taking an exit, gating [8] is applied. In situations where we are uncertain about the road at far distances, making a hard decision about the motion state of the vehicles can lead to more sensitivity towards outliers.

In this paper we present a Bayesian inference algorithm based on a segmented clothoid-based road model that can estimate the road ahead of the host vehicle up to 200 meters. We use the shape of the lane markings and the heading of the leading vehicles to update the road state. To distinguish between keeping lane/changing lane vehicles, a multiple-model filter has been used. By following this approach we get a more robust description of our uncertainties and avoid the disadvantage of having to deal with false detections while making efficient use of the available information. Two commonly used multiple-model filters are the interacting multiple model (IMM) filter [9] [10] and the generalized pseudo-Bayesian (GPB) filter [9]. Due to computational complexity and storage considerations of the fusion system we are currently working with, GPB1 algorithm is the one that suits our purpose best. Our algorithm is evaluated on real data collected from highways across Europe. The road estimation root mean-squared error (RMSE) is used as the measure of performance.

The paper is organized as follows. We lay out the road estimation problem in Section II. The proposed road model, its resultant state parametrization and process model are explained in Section III. Section IV includes the measurement models. We explain our implementation in Section V. The evaluation of our algorithm is presented in Section VI followed by concluding remarks in Section VII.

II. PROBLEM FORMULATION AND SYSTEM DESCRIPTION

We are interested in estimating the road geometry up to 200 meters ahead of the host vehicle in highway scenarios. The road geometry is defined as the shape of the middle of the host vehicle's lane. The road state, denoted \mathbf{r}_k , contains a parametrization of the road geometry at time k . We assume

* This work was supported by VINNOVA. M. Fatemi, L. Hammarstrand and L. Svensson are with the department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden. {maryam.fatemi, lars.hammarstrand, lennart.svensson}@chalmers.se. Á. F. García-Fernández is with the department of Electrical and Computer Engineering, Curtin University, Perth, Australia. angel.garciafernandez@curtin.edu.au.

that given \mathbf{r}_k , there exists a mapping¹ that can describe the road geometry in the local Cartesian coordinate system (x_k^l, y_k^l) attached to the host vehicle. We assume that the position of the host vehicle (x_k^h, y_k^h) and its orientation ψ_k^h is known in a fixed global Cartesian coordinate system (x^g, y^g) . In this work we consider having only one road, i.e., exits and forks are not accounted for in the designed model. Furthermore, we are working within an asynchronous fusion framework similar to the one described in [11]. This implies that the sensors are asynchronous and upon receiving the measurement of each one we align the time index of the state vector with that sensor and update with the measurement.

We aim at estimating the road geometry accurately using the measurements provided by the on-board sensors. Besides inertial measurement sensors and wheel speed sensors, our vehicle is equipped with a camera and a radar. The camera detects the left and right lane markings and describes each side by a third degree polynomial. These polynomials are given in the local coordinate system (x_k^l, y_k^l) as

$$y_k^l = l_k^0 + l_k^1 x_k^l + l_k^2 (x_k^l)^2 + l_k^3 (x_k^l)^3 \quad (1)$$

$$y_k^r = r_k^0 + r_k^1 x_k^l + r_k^2 (x_k^l)^2 + r_k^3 (x_k^l)^3 \quad (2)$$

where $\mathbf{z}_k^l = [l_k^0 \ l_k^1 \ l_k^2 \ l_k^3]^T$ are the coefficients for the left lane markings and $\mathbf{z}_k^r = [r_k^0 \ r_k^1 \ r_k^2 \ r_k^3]^T$ for the right. These coefficients are accompanied by two confidence values, one for each side telling us how much we can trust the coefficients and up to what distance the polynomial is accurate. We use these confidence values to decide if the lane marking measurements are accurate enough to be used.

The radar detects and tracks moving objects on the road. For each observed vehicle i a measurement \mathbf{z}_k^i is reported by the sensor given as,

$$\mathbf{z}_k^i = [x_k^i, y_k^i, \phi_k^i, v_k^i]^T \quad (3)$$

where (x_k^i, y_k^i) is the position in the local coordinate frame and ϕ_k^i is the heading angle relative to the heading of the host and v_k^i is the speed in that direction. It should be noted that these values are filtered by the sensor supplier and they are not raw measurements.

The sequence of measurements up to and including time k are presented by $\mathbf{Z}_{1:k}$, the aim is to recursively calculate the posterior density of the road state \mathbf{r}_k given these observations $p(\mathbf{r}_k | \mathbf{Z}_{1:k})$.

III. ROAD MODEL

Road constructors usually follow a set of guidelines that are placed to ensure a smooth ride for the users. Among other things, this indicates that the curvature of the road can not change abruptly [12]. Based on this notion we have designed our road model. In this section we describe the clothoid-based road model, the resulting state parametrization and the process model which are illustrated in Fig. 1.

A. State Parametrization

Our proposed road model consists of n connected segments. Each segment is described by a clothoid, i.e., a parametric curve that has a linearly changing curvature $\kappa_r(s)$. Therefore the curvature of segment i is described by

$$\kappa_r^i(s) = \kappa_0^i + \kappa_1^i s \quad (4)$$

where s is the arc length, κ_0^i is the initial curvature and κ_1^i is the curvature change rate. The segments are connected to each other in a manner that ensures G^2 -continuity, i.e., the position, heading and curvature of two segments are equal at their joints. Moreover, at the beginning, all the segment lengths are equal to the default segment length which is $l_s = l_r/n$ where l_r denotes the road length and n is the number of segments.

In order to describe the geometry of the road relative to the host vehicle, we need the curvature at the host vehicle's position denoted by $\kappa_{0,k}$ and the curvature change rate of each segment $\kappa_1^1, \dots, \kappa_1^n$. Furthermore, to complete our description we need the distance between the host vehicle (offset) and the starting point of the road and the heading of the road at that point denoted by y_k^{off} and φ_k , respectively. Note that the road parameters are expressed in the local coordinate system. Finally we can put all of these together to form the road state, which is described as

$$\mathbf{r}_k = [y_k^{\text{off}}, \varphi_k, \kappa_{0,k}, \kappa_{1,1}^1, \dots, \kappa_{1,k}^n]^T. \quad (5)$$

The curvature change rates, $\kappa_{1,1}^1, \dots, \kappa_{1,k}^n$, are related to a fixed part of the road and $y_k^{\text{off}}, \varphi_k$ and $\kappa_{0,k}$ are related to the current position of the host vehicle.

A description of each segment of the road given by \mathbf{r}_k in local Cartesian coordinates is

$$x_r^i(s) = x_0^i + \int_0^s \cos(\varphi_r^i(s)) ds \quad (6)$$

$$y_r^i(s) = y_0^i + \int_0^s \sin(\varphi_r^i(s)) ds \quad (7)$$

where s denotes the arc length into segment i and

$$\varphi_r^i(s) = \varphi_0^i + \kappa_0^i s + \frac{\kappa_1^i}{2} s^2 \quad (8)$$

is the heading of segment i at that arc length. Note that $x_0^i, y_0^i, \varphi_0^i$ and κ_0^i are the position, heading and curvature of the starting point of segment i . Due to G^2 -continuity, these values are the same as the last point of the previous segment. The integrals in (6) and (7) do not have a closed form solution and we solve them by calculating the Taylor series expansion of the integrand over $l_s/2$.

B. Process Model

The parameters of the state vector describe the road ahead at each time step and as we drive along the road these parameters will change accordingly. The process model describes these changes and consists of two parts. The first part compensates for the host vehicle movement and the second part describes the evolution of the road.

¹The details of this mapping are explained in Section III-A.

The first three elements of our state vector namely y_k^{off} , φ_k and $\kappa_{0,k}$ describe the road at the current position of the host vehicle. So in order to calculate their time evolution we need to compensate for the movement of the host vehicle. Knowing how much we have moved along the road, we can locate the new starting point of the road and derive its offset, heading and curvature.

We receive the position and orientation of the host vehicle in a fixed global coordinate system from the fusion system. These parameters, at previous and current time stamp, are denoted by $(x_{k-1}^h, y_{k-1}^h, \psi_{k-1}^h)$ and (x_k^h, y_k^h, ψ_k^h) , respectively. We denote the starting position of the road at the previous time stamp by $(x_{k-1}^{g,r}, y_{k-1}^{g,r})$ and the current time stamp by $(x_k^{g,r}, y_k^{g,r})$. These coordinates are illustrated in Fig. 1. The traveled distance along the road is expressed as

$$d_k = \sqrt{(x_k^{g,r} - x_{k-1}^{g,r})^2 + (y_k^{g,r} - y_{k-1}^{g,r})^2}. \quad (9)$$

Note that $(x_k^{g,r}, y_k^{g,r})$ is the intersection of two lines one originating from $(x_{k-1}^{g,r}, y_{k-1}^{g,r})$ in the direction of φ_{k-1} and the other originating from (x_k^h, y_k^h) in the direction of $\psi_k^h + \pi/2$. Additionally, the process models of y_k^{off} , φ_k and $\kappa_{0,k}$ are calculated by

$$\begin{aligned} y_{k+1}^{\text{off}} &= \begin{bmatrix} -\sin(\psi_k^h) & \cos(\psi_k^h) \end{bmatrix} \begin{bmatrix} x_k^{g,r} - x_k^h \\ y_k^{g,r} - y_k^h \end{bmatrix} \\ &+ \nu_k^y \\ \varphi_{k+1} &= \varphi_r(d_k) - (\psi_k^h - \psi_{k-1}^h) + \nu_k^\varphi \\ \kappa_{0,k+1} &= \kappa_r(d_k) + \nu_k^\kappa \end{aligned} \quad (10)$$

where $\kappa_r(\cdot)$ and $\varphi_r(\cdot)$ are described by (4) and (8), respectively. Moreover, ν_k^κ , ν_k^y and ν_k^φ are Gaussian process noise terms which account for the uncertainties that exist in the calculations of the host vehicle's movement.

As stated previously, the road model consists of n segments. When the host vehicle is driving within a segment, d_k is subtracted from the first segment and added to the last, additionally, the curvature change rates of all the segments remain the same.

$$\kappa_{1,k+1}^i = \kappa_{1,k}^i \quad \forall i = 1 \dots n \quad (11)$$

When the host vehicle reaches the end of the first segment, this segment is removed and a new segment is added at the end. The segment lengths are adjusted such that the road has the same total length. As the first segment is removed, the curvature rates of the remaining segments shift to the left

$$\kappa_{1,k+1}^i = \kappa_{1,k}^{i+1} \quad \forall i = 1 \dots n-1. \quad (12)$$

The curvature change rate of the new segment is calculated according to

$$\kappa_{1,k}^n = \nu_k^{\kappa_1} \quad (13)$$

where the uncertainty is modeled by $\nu_k^{\kappa_1} \sim \mathcal{N}(0, \sigma_{\kappa_1}^2)$.

IV. MEASUREMENT MODELS

We use two sets of measurements to estimate the road geometry, namely the lane marking measurements and measurements of the moving vehicles. Although we receive filtered tracks from the sensors, since we are not given any information about their covariances we treat them as raw measurements. In this section we describe the corresponding measurement models.

A. Lane Markings

The information about the lane markings are provided by the camera as coefficients of a third-degree polynomial. To more conveniently relate this information to the road state, we sample each polynomial by four equally spaced points up to the length where the sensor states that it is accurate. These points which completely describe the third degree polynomials, are collected in a vector called \mathbf{p}_k .

The measurement model describes the relationship between the sampled points and the state vector. The geometry of the left and right lane markings is described by translating the road half a lane width to the left and to the right. The translation is carried out by calculating the parallel clothoid of each segment similar to [5]. The following constraints are used to make sure that the translated clothoid of each segment (to either right or left), is parallel to the segment

$$\begin{aligned} \tilde{\varphi}^0 &= \varphi^0 \\ \tilde{s}^i &= s^i - \frac{\omega_k^L}{2} \Delta\varphi^i \\ \tilde{\kappa}_0^i &= \frac{1}{\frac{1}{\kappa_0^i} + \frac{\omega_k^L}{2}} \\ \Delta\varphi^i &= \Delta\tilde{\varphi}^i \end{aligned} \quad (14)$$

where $\Delta\varphi^i = \kappa_0^i s^i + \frac{\kappa_1^i}{2} (s^i)^2$ is the difference of the heading between the start and the end of segment i . The lane width at time k is denoted by ω_k^L the calculation of which is explained in Section V-A. $\tilde{\varphi}^0$ and φ^0 denote the heading at the beginning of the segment of the translated and original clothoid, respectively. Furthermore, \tilde{s}^i , $\tilde{\kappa}_0^i$ and $\tilde{\kappa}_1^i$ denote the segment length, the curvature at the current position of the host vehicle and the curvature change rate of segment i , respectively, of the translated clothoid. Accordingly, the measurement model is described by

$$\mathbf{p}_k = g(\mathbf{r}_k, \mathbf{s}_k, \omega_k^L) + \mathbf{w}_k^l(\mathbf{s}_k) \quad (15)$$

where $g(\cdot)$ is a nonlinear function performing the sampling and the translation described by (14). Furthermore, \mathbf{s}_k is a vector which contains the sampling distances in arc length and $\mathbf{w}_k^l \sim \mathcal{N}(0, \mathbf{R}^l)$ describes the uncertainty around measurement points.

B. Moving Vehicles

We use the heading of the moving vehicles to update the road state using the assumption that if the vehicle is following its lane, its heading and the heading of the road at the position of the vehicle should be approximately the same. Since this assumption is not always valid, e.g., when

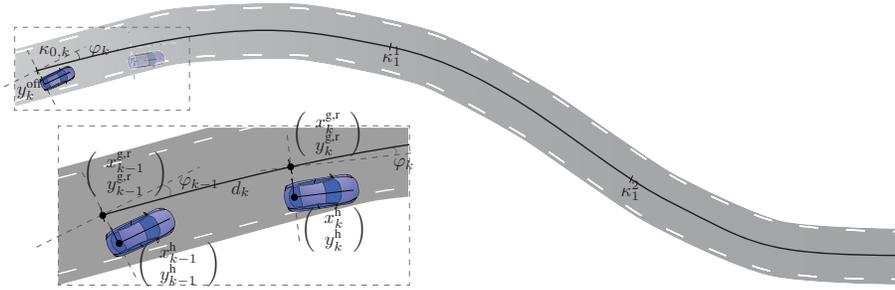


Fig. 1. Illustration of the road model, state parametrization and process model.

the vehicle changes lane or takes an exit, we propose two measurement models each describing these two scenarios. Assuming that vehicle i keeps its lane, its heading denoted by ϕ_k^i at position $[x_k^i, y_k^i]^T$ is approximately equal to the heading of the road at that distance. This is expressed as

$$\phi_k^i = \varphi_r(s(x_k^i, y_k^i)) + \eta_k^{i,0} \quad (16)$$

where $s(\cdot)$ denotes the arc length associated with the position of vehicle i and $\eta_k^{i,0} \sim \mathcal{N}(0, \sigma_{\eta_0}^2)$.

For the case where the vehicle is changing lane/taking an exit, we propose to use the same model structure but with a higher noise level. This is described by

$$\phi_k^i = \varphi_r(s(x_k^i, y_k^i)) + \eta_k^{i,1} \quad (17)$$

where $\eta_k^{i,1} \sim \mathcal{N}(0, \sigma_{\eta_1}^2)$ and since the vehicle is not parallel to the road anymore, $\sigma_{\eta_1}^2$ has a large value. Based on these two noise models, we have two modes for each vehicle. One mode states that the vehicle follows its lane while the other states that it is changing lane/taking an exit. We denote the modes of a single vehicle at time k by $m_k \in \{0, 1\}$ where $m_k = 0$ states that the vehicle is following its lane.

Finally, to model how a vehicle transitions between these two modes, we use a transition probability matrix where each element represents a probability of the form $\Pr\{m_k | m_{k-1}\}$. The transitional probabilities which have been calculated by the method mentioned in [13] are

$$\mathbf{\Pi} = \begin{pmatrix} 0.982 & 0.018 \\ 0.008 & 0.992 \end{pmatrix}$$

At time k we get observations on n_k leading vehicles. As each of these vehicles can have two modes, we can form $N = 2^{n_k}$ mode hypotheses. We denote the q^{th} hypothesis at time k by

$$\mathbf{m}_k^q = [m_k^{q,1}, m_k^{q,2}, \dots, m_k^{q,n_k}]$$

The resulting n_k -dimensional measurement model for a given hypothesis is expressed as

$$\phi_k = h(\mathbf{r}_k) + \mathbf{w}_k^o(\mathbf{m}_k^q) \quad (18)$$

where $\phi_k \in \mathbb{R}^{n_k}$ is a vector that includes the heading measurements of all the vehicles at time k , $h(\mathbf{r}_k)$ is a linear function described according to (16) and (17) for each vehicle and $\mathbf{w}_k^o(\mathbf{m}_k^q) \sim \mathcal{N}(0, \mathbf{R}_q)$. Since we assume the measurements to be independent, the measurement noise

covariance is a diagonal matrix denoted by $\mathbf{R}_q \in \mathbb{R}^{n_k \times n_k}$ where each entry of the diagonal is equal to either $\sigma_{\eta_0}^2$ or $\sigma_{\eta_1}^2$ depending on \mathbf{m}_k^q .

V. IMPLEMENTATION

We follow a Bayesian approach to form the posterior density of the state vector \mathbf{r}_k given measurements of the lane markings and the moving objects. This approach consists of two steps, prediction and measurement update. As the process model is nonlinear, we use a square root cubature Kalman filter (sqCKF) [14] to approximate the predicted density using the process model described in Section III-B. In this section we explain how we have implemented the measurement update step to form the posterior density. The pseudo-code of the complete filtering algorithm is presented in Algorithm 1.

```

if The filter has been initialized then
    Prediction (see Section III-B);
    if Lane marking measurements received then
        Update with the lane marking measurements
        (see Section IV-A);
    end
    if Other vehicles measurements received then
        Update with other vehicles (see Section IV-B);
    end
else
    Initialize the filter (see Section VI);
end

```

Algorithm 1: Pseudo-code of the filtering algorithm.

A. Lane Markings

We use (15) to do the measurement update and since this model is nonlinear we use the sqCKF. The square root implementation is chosen due to its robustness. The lane width at time k in (15) is calculated as

$$\omega_k^L = \sum_{j=k-L}^k \frac{l_j^0 - r_j^0}{L} \quad (19)$$

where l_k^0 and r_k^0 are the zero order coefficients of the lane marking polynomials on the left and right, respectively and L is the window size over which we perform the averaging.

If a lane change is detected, before performing the update we translate the predicted road state accordingly (to the left or right) using (15). We detect a lane change by monitoring the offset measurements (l_k^0, r_k^0) where a jump comparable to a lane width indicates a lane change.

B. Moving Vehicles

In this part, we only use mature tracks, i.e. we wait a bit before we start using the leading vehicles observations to update our road estimates. The posterior density that results from such measurements, more specifically the heading of other vehicles on the road, is expressed as

$$p(\mathbf{r}_k | \mathbf{Z}_{1:k}) = \sum_{q=1}^N \Pr\{\mathbf{m}_k^q | \mathbf{Z}_{1:k}\} p(\mathbf{r}_k | \mathbf{m}_k^q, \mathbf{Z}_{1:k}) \quad (20)$$

As can be seen, we need to calculate a multiple-model posterior density for which there exists different approximations e.g. the IMM filters [9] [10] and the GPB filters [9]. We are working within a real-time asynchronous fusion system which imposes some constraints on the choice of the multiple-model algorithm. More specifically, in order to use the IMM filters or the GPB filters with depth more than one, we need to propagate the means and covariances of all the models throughout the whole system. Therefore, to avoid the resulting storage and computational complexity demand, we choose the GPB1 algorithm where we can approximate the posterior density with a single Gaussian. Since we have used the sqCKF for lane marking measurement update, we have implemented the square root GPB1. This implementation increases the robustness of our filter.

We can see in (20) that calculating the multiple-model posterior density boils down to calculating the posterior density resulting from each hypothesis $p(\mathbf{r}_k | \mathbf{m}_k^q, \mathbf{Z}_{1:k})$ together with its weight $\Pr\{\mathbf{m}_k^q | \mathbf{Z}_{1:k}\}$. Given the hypothesis and the heading measurements, the posterior density is described by

$$p(\mathbf{r}_k | \mathbf{m}_k^q, \mathbf{Z}_{1:k}) \propto p(\mathbf{r}_k | \mathbf{Z}_{1:k-1}) p(\phi_k | \mathbf{m}_k^q, \mathbf{r}_k) \quad (21)$$

where the prediction density is calculated using the process model described in Section III-B and the likelihood function is explained in (18). The probability of each hypothesis is expressed as

$$\begin{aligned} \Pr\{\mathbf{m}_k^q | \mathbf{Z}_{1:k}\} &\propto p(\phi_k | \mathbf{m}_k^q, \mathbf{Z}_{1:k-1}) \\ &\times \Pr\{\mathbf{m}_k^q | \mathbf{Z}_{1:k-1}\} \end{aligned} \quad (22)$$

where the first term is the distribution of the innovation

$$p(\phi_k | \mathbf{m}_k^q, \mathbf{Z}_{1:k-1}) = \mathcal{N}(\phi_k; \hat{\phi}_{k|k-1}, \mathbf{S}_q) \quad (23)$$

and the second term is the predicted weight. These weights are calculated as

$$\begin{aligned} \Pr\{\mathbf{m}_k^q | \mathbf{Z}_{1:k-1}\} &= \prod_{j=1}^{n_k} \sum_{m_{k-1}^j=0}^1 (\Pr\{m_k^{q,j} | m_{k-1}^j\}) \\ &\times \Pr\{m_{k-1}^j | \mathbf{Z}_{1:k-1}\} \end{aligned} \quad (24)$$

where the first term comes from the transition probability matrix $\mathbf{\Pi}$ which is explained in Section IV-B. To make the picture complete we need to calculate the following

$$\Pr\{m_{k-1}^j = 0 | \mathbf{Z}_{1:k-1}\} = \sum_{q \in \mathcal{H}} \Pr\{\mathbf{m}_{k-1}^q | \mathbf{Z}_{1:k-1}\}$$

$$\Pr\{m_{k-1}^j = 1 | \mathbf{Z}_{1:k-1}\} = 1 - \Pr\{m_{k-1}^j = 0 | \mathbf{Z}_{1:k-1}\}$$

where \mathcal{H} is the set of all hypotheses in which we assume that the j^{th} vehicle follows its lane.

As stated before, we approximate the multi-modal posterior in (20) as a single Gaussian. The first two moments of this approximated density are the estimated road state and its covariance. Note that we are working with the square root covariance rather than the covariance itself. We use the following notations to present the derivation of the estimated road state and its square root covariance

$$\begin{aligned} p(\mathbf{r}_k | \mathbf{m}_k^q, \mathbf{Z}_{1:k}) &= \mathcal{N}(\mathbf{r}_k; \hat{\mathbf{r}}_{k|k}^q, \hat{\mathbf{P}}_{k|k}^q) \\ \Pr\{\mathbf{m}_k^q | \mathbf{Z}_{1:k}\} &= \omega_q \end{aligned}$$

where $\hat{\mathbf{P}}_{k|k}^q \in \mathbb{R}^{n \times n}$ and n is the dimension of the state vector. The estimated road state is expressed as

$$\hat{\mathbf{r}}_{k|k} = \sum_{q=1}^N \omega_q \hat{\mathbf{r}}_{k|k}^q.$$

We derive the square root covariance of the estimate by using techniques similar to the one presented in [15]. The estimated covariance of the Gaussian mixture of (20) is expressed as

$$\begin{aligned} \hat{\mathbf{P}}_{k|k} &= \sum_q \omega_q \hat{\mathbf{P}}_{k|k}^q \\ &+ \sum_q \omega_q (\hat{\mathbf{r}}_{k|k}^q - \hat{\mathbf{r}}_{k|k}) (\hat{\mathbf{r}}_{k|k}^q - \hat{\mathbf{r}}_{k|k})^T \end{aligned} \quad (25)$$

By factoring $\hat{\mathbf{P}}_{k|k}^q = \mathbf{A}_{k|k}^q \mathbf{A}_{k|k}^{qT}$, where T denotes matrix transpose, the square root covariance is found by performing two QR decompositions. The first decomposition gives us the square root matrix resulting from the first term in (25)

$$\mathbf{M}_k = qr\{[\sqrt{\omega_1} \mathbf{A}_{k|k}^1 \quad \dots \quad \sqrt{\omega_N} \mathbf{A}_{k|k}^N]^T\}$$

where $\mathbf{M}_k \in \mathbb{R}^{n \times n}$ and $qr\{\cdot\}$ denotes performing the QR decomposition and extracting the first n rows and columns of the transpose of the resulting R matrix. Following the same notation the square root of the covariance matrix is

$$\mathbf{A}_{k|k} = qr\left\{ \left[\begin{array}{c} \sqrt{\omega_1} \Delta \hat{\mathbf{r}}_{k|k}^1 \quad \dots \quad \sqrt{\omega_N} \Delta \hat{\mathbf{r}}_{k|k}^N \\ \mathbf{M}_k \end{array} \right] \right\}$$

where $\Delta \hat{\mathbf{r}}_{k|k}^i = (\hat{\mathbf{r}}_{k|k}^i - \hat{\mathbf{r}}_{k|k})$. Since we approximated the posterior density in (20) by a single Gaussian distribution, it is completely described by its mean $\hat{\mathbf{r}}_{k|k}$ and square root covariance $\mathbf{A}_{k|k}$.

VI. EVALUATION

In this section we evaluate the performance of our filter on real data collected from highways. We are mainly interested in calculating the road estimation error, i.e., we compare the true road sampled every 20 meters up to 200 meters to the estimated road sampled at the same distances. We use the

road estimation RMSE as the measure of performance. The true road is calculated from the host vehicle's path in the global coordinate system and the lane marking coefficients l_k^0 and r_k^0 . We calculate the host path by the dead-reckoning method, i.e., using the internal sensor measurements. The precision of this method and the assumptions upon which we find this method reasonable are discussed in detail in [7].

We evaluate the performance in two different scenarios. The first, only using lane marking measurements to estimate the road and the second using measurements of both lane markings and moving vehicles. By comparing the results in the two scenarios, we show the benefit of using moving vehicle measurements. Furthermore, for the case where we use moving vehicle measurements, we compare the results of two different measurement update methods. The first is the sqCKF-GPB1 filter described in Section V-B and the second is the sqCKF-gating used in [7]. Gating is an outlier detection method where a measurement is only accepted as a valid measurement if the Mahalanobis distance between the measurement and the predicted measurement is below a threshold. The sqCKF-GPB1 and sqCKF-gating use the same state parametrization and process model. Their key difference in the measurement update is how they treat the measurements of other vehicles.

Both filters are initiated by the first measurements of the left and right lane markings. The settings for the process and measurement noise are expressed in terms of their standard deviations. Both filters use the same setting for the process noise, i.e., $\sigma_k^y = 0.4m$, $\sigma_k^\varphi = 0.5^\circ$, $\sigma_k^{\kappa} = 10^{-5}m^{-1}$, $\sigma_k^{\kappa_1} = 2e10-6m^{-2}$, as well as for the measurement noise of the lane marking samples. The noise of the lane marking samples are $0.01m$ and $0.025m + s_k/10$ for the x and y coordinates, respectively, where s_k is the sampling distance in arc length. For the leading vehicles, the measurement noise is set to $\sigma_{\eta^0} = 0.5^\circ$ (keep lane) and $\sigma_{\eta^1} = 3^\circ$ (changing lane), for the sqCKF-GPB1. These values are increased by the rate of $1^\circ/100m$ based on the longitudinal distance to the host vehicle. The sqCKF-gating algorithm uses σ_{η^0} as the measurement noise and the gate size is set to three. These values are set by reasoning and trial and error.

Fig. 2 compares the RMSE error of the three different cases. For this evaluation we chose a rather windy highway, the profile of which is depicted in Fig. 3, and where there were many vehicles present around the host vehicle. As can be seen, using the heading of the leading vehicles improves the performance significantly. The improvement happens for the distances over 50-60 meters and that is exactly the distances up to which we typically have lane marking measurements. Additionally, the sqCKF-GPB1 performs better than sqCKF-gating.

In Fig. 4 we have compared the road estimation RMSE averaged over ten log files which constitute a total of 30 minutes of driving (62.4 km). The data of these log files contain measurements recorded from different highway scenarios, i.e., windy, straight, flat and hilly highways. It should be mentioned that these log files include both busy

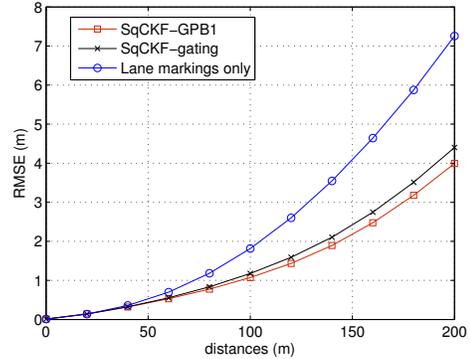


Fig. 2. Comparison of road estimation RMSE using only lane marking measurements and using both lane marking measurements and the heading of the leading vehicles. When using both measurements the sqCKF-gating and the sqCKF-GPB1 are used.

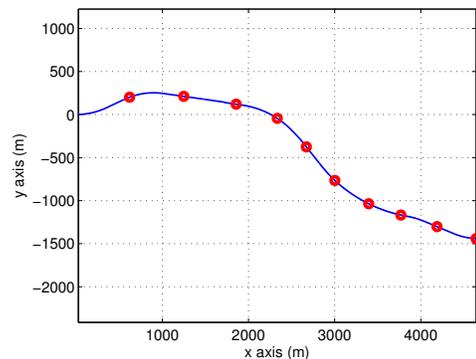


Fig. 3. True road in the global coordinate system. The dots denote the position of the host vehicle on the road every 20 seconds.

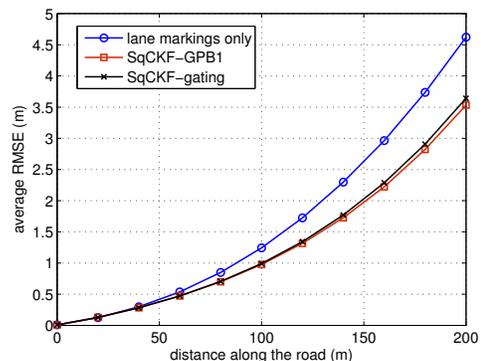


Fig. 4. Comparison of road estimation RMSE averaged over 30 minutes of driving (62.4 km).

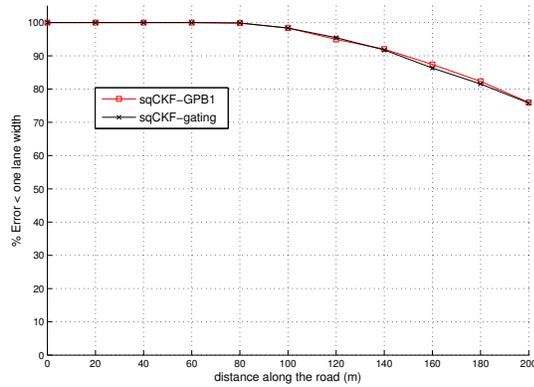


Fig. 5. Percentage of the time that the road estimation RMSE is below one lane width for sqCKF-GPB1 and sqCKF-gating.

and empty highways. More specifically, in some of the log files there are many leading vehicles present while in others there are only one or two vehicles besides the host. As such, the data that has been used for evaluation includes many different real-world highway situations. The result clearly indicates that using the heading of other vehicles improves the performance significantly, additionally, the sqCKF-GPB1 filter performance is only slightly better than the sqCKF-gating algorithm. We believe the reason for this is that detecting a lane change at far distances on a rather empty and windy road is a difficult problem for both algorithms.

Finally, Fig. 5 depicts the percentage of the times that the road estimation RMSE is lower than one lane width for sqCKF-GPB1 and sqCKF-gating. This percentage is calculated from the same data used to generate Fig. 4. We can see that the RMSE at 200m from the host vehicle is less than one lane width 76% of the times for both algorithms.

VII. CONCLUSION

In this paper, we develop a Bayesian inference algorithm to estimate the road ahead of the host vehicle up to 200 meters, using a segmented clothoid-based road model. We use observations of the shape of the lane markings and the heading of the leading vehicles to update the road state. We use the GPB1 algorithm to distinguish between the vehicles which are keeping their lane and those that do not. We evaluate our algorithm on real data collected in highways across Europe. The results show that using the heading of leading vehicles improves the road estimation error significantly. Additionally, the sqCKF-GPB1 performs better than the sqCKF-gating in some scenarios.

Use of other sources of information such as the measurements of the guardrails could improve the performance of the road geometry estimation. We will address this issue in our future research.

VIII. ACKNOWLEDGMENTS

We would like to thank Volvo Car Group for providing us with the data used in the evaluations.

REFERENCES

- [1] C. Lundquist and T. B. Schön, “Joint ego-motion and road geometry estimation,” *Information Fusion*, vol. 12, no. 4, pp. 253–263, 2011.
- [2] K. Kaliyaperumal, S. Lakshmanan, and K. Kluge, “An algorithm for detecting roads and obstacles in radar images,” *IEEE Transactions on Vehicular Technology*, vol. 50, no. 1, pp. 170–182, Jan. 2001.
- [3] Z. Zomotor and U. Franke, “Sensor fusion for improved vision based lane recognition and object tracking with range-finders,” in *Proceedings of IEEE Conference on Intelligent Transportation System*, 1997, pp. 595–600.
- [4] R. Chapuis, R. Aufrere, and F. Chausse, “Accurate road following and reconstruction by computer vision,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 4, pp. 261–270, Dec. 2002.
- [5] C. Gackstatter, S. Thomas, and G. Klinker, “Fusion of clothoid segments for a more accurate and updated prediction of the road geometry,” in *Intelligent Transportation Systems (ITSC), 13th International IEEE Conference on*, Funchal, Sep. 2010, pp. 1691 – 1696.
- [6] T. B. Schön, A. Eidehall, and F. Gustafsson, “Lane departure detection for improved road geometry estimation,” in *Intelligent Vehicle Symposium*, Tokyo, Japan, June 2006.
- [7] A. F. García-Fernández, L. Hammarstrand, M. Fatemi, and L. Svensson, “Bayesian road estimation using on-board sensors,” *IEEE Transaction on intelligent transportation systems*, 2014.
- [8] Y. Bar-Shalom and E. Tse, “Tracking in a cluttered environment with probabilistic data association,” *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.
- [9] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. Academic Press, 1988.
- [10] H. A. P. Blom and Y. Bar-Shalom, “The interacting multiple model algorithm for systems with markovian switching coefficients,” *IEEE Transaction on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.
- [11] F. Bengtsson and L. Danielsson, “A design architecture for sensor data fusion systems with application to automotive safety,” in *15th world congress on Intelligent Transport Systems*, 2008.
- [12] A. Gern, U. Franke, and P. Levi, “Robust vehicle tracking fusing radar and vision,” in *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2001, pp. 323–328.
- [13] A. K. Kristian Weiss, Nico Kaempchen, “Multiple-model tracking for the detection of lane change maneuvers,” in *Intelligent Vehicles Symposium, IEEE*, July 2004, pp. 937 – 942.
- [14] I. Arasaratnam and S. Haykin, “Cubature Kalman filters,” *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, June 2009.
- [15] R. van der Merwe and E. A. Wan, “The square-root unscented Kalman filter for state and parameter-estimation,” in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, Salt Lake City, UT, May 2001, pp. 3461 – 3464.