Bayesian Sensor Fusion of GNSS and Camera With Outlier Adaptation for Vehicle Positioning

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

TR2022-093 August 06, 2022

Abstract

In this paper we develop a method for vehicle positioning based on global navigation satellite system (GNSS) and camera information. Both GNSS and camera measurements have noise characteristics that vary in time. As a result, the measurements can abruptly change from reliable to unreliable from one time step to another. To adapt to the changing noise levels and hence improve positioning performance, we combine GNSS information with measurements from a forward looking camera, a steering-wheel angle sensor, wheel-speed sensors, and optionally an inertial sensor. We pose the estimation problem in an interacting multiple-model (IMM) setting and use Bayes recursion to choose the best combination of the estimators. 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

International Conference on Information Fusion (FUSION) 2022

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

Bayesian Sensor Fusion of GNSS and Camera With Outlier Adaptation for Vehicle Positioning

Karl Berntorp, Marcus Greiff, and Stefano Di Cairano

Abstract—**In this paper we develop a method for vehicle positioning based on global navigation satellite system (GNSS) and camera information. Both GNSS and camera measurements have noise characteristics that vary in time. As a result, the measurements can abruptly change from reliable to unreliable from one time step to another. To adapt to the changing noise levels and hence improve positioning performance, we combine GNSS information with measurements from a forward looking camera, a steering-wheel angle sensor, wheel-speed sensors, and optionally an inertial sensor. We pose the estimation problem in an interacting multiple-model (IMM) setting and use Bayes recursion to choose the best combination of the estimators. 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.**

I. INTRODUCTION

The need for 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 widely spread. For such applications, high positioning accuracy is needed for safety-critical obstacle and lane-change maneuvering, and to provide comfortable autonomous vehicle control.

Road-vehicle positioning can be approached in numerous ways depending on the sensor suite and the communication interface that is being employed. Using only onboard sensing is convenient as it removes the need for external communication. However, it has the drawback that it can only provide positioning in the local (vehicle) frame, as global positioning information is lacking. For instance, [1] fuses information from several sensors to perform joint road geometry estimation and vehicle tracking. This work was extended in [2], 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 [3], which in addition to the sensors in [2] develop a novel road model with claimed higher prediction accuracy compared to other established road models.

Other positioning methods rely on global navigation satellite system (GNSS) information. GNSSs estimate a receiver's (e.g., located in the vehicle) states from a set of code and carrier-phase measurements, acquired from one or several constellations of satellites and transmitted over one or more frequency bands [4], [5]. The measurement equation, from which the receiver state is inferred, is time-varying, nonlinear in the position of the receiver, and incorporates various biases [6]. For the carrier-phase measurements, some of these biases are integer-valued, commonly referred to as *ambiguities* [7]. Leveraging that these biases are integers, the estimator can significantly improve the estimation accuracy [5], [8]–[10].

Since GNSS measurements are prone to occasional errors and cannot always deliver high-level accuracy, particularly in deep urban settings, one may perform sensor fusion of the position estimate of said GNSS-based estimators with onboard vehicle sensors. 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 this paper, we address the vehicle-positioning estimation problem by fusing GNSS position measurements with a forward looking camera, steering-wheel sensing, wheel-speed sensing, and (optionally) an IMU. In addition to vehicle positioning, our method can also provide estimates of the road geometry. The camera and GNSS measurements are complementary in the sense that provided that lane markings are visible, the camera provides position measurements relative to the road independent of whether the vehicle drives in a rural or urban setting. On the other hand, the GNSS measurements provide global position information irrespective of the lane markings. However, combining GNSS with additional sensing does not circumvent the fact that GNSS measurements are prone to occasional errors (e.g., due to partial occlusion). Hence, to become truly robust to GNSS measurement errors, also the time-varying reliability of the measurements need to be estimated. 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 [14]. However, also 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. To account for this, we model said 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) [15] to handle the nonlinearities in the estimation model. 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.

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

The proposed method differs from [2], [3] in that we incorporate GNSS information to yield performance improvements, and systematically deal with outliers that are likely to be present in the considered vehicle applications. While [11], [12] use GNSS information to make related vehicle estimation problems observable, the IMM method considered in this paper is specifically designed to handle outliers.

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 $\boldsymbol{\Sigma}$. Matrices are written in capital bold font as $\boldsymbol{X}$, and the element on row i and column j of $\boldsymbol{X}$ is denoted with X_{ij}. We let $\hat{\boldsymbol{x}}_{j|m}$ denote the estimate of $\boldsymbol{x}$ at time step j given the measurement sequence $\boldsymbol{y}_{0:m} = \{\boldsymbol{y}_0, \dots \boldsymbol{y}_m\}$. With $p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k})$, we mean the posterior density function of the state $\boldsymbol{x}_k$ from time step 0 to time step k given $\boldsymbol{y}_{0:k}$. The concatenation of two vectors $\boldsymbol{x} \in \mathbb{R}^{n_{\boldsymbol{x}}}$ and $\boldsymbol{y} \in \mathbb{R}^{n_{\boldsymbol{y}}}$ is $(\boldsymbol{x}; \boldsymbol{y}) = (\boldsymbol{x}^\top, \boldsymbol{y}^\top)^\top \in \mathbb{R}^{n+m}$. Furthermore, $\mathbf{1}_{n\times n}$ denotes the $n \times n$ identity matrix, $\mathbf{1}_n$ is a column vector of n elements equal to one, and $(\boldsymbol{a})(\star)^\top = (\boldsymbol{a})(\boldsymbol{a})^\top$ for an expression $\boldsymbol{a}$.

Outline: Sec. II outlines the assumptions, the sensors, the dynamic models, and the associated measurement models our method relies on. Sec. III presents the proposed method, which is evaluated in Sec. IV, and finally, Sec. V closes the paper.

II. Modeling

In this section we present the motion and measurement models employed in the estimation. We assume to have access to steering-wheel and wheel encoders. Based on these inputs, we present two vehicle models of varying complexity. Ultimately, the goal of many vehicle estimations is to be employed in closed loop with controllers enabling ADAS and AD features. However, to accurately determine future vehicle decisions, not only the ego vehicle motion but also the knowledge of its surroundings is of utmost importance [16], [17]. To this end, we also present a representation of the road geometry as a dynamical system, dependent on the chosen vehicle model. Depending on the amount onboard sensing assumed available and what is known about the vehicle parameters, models of different complexity can be used. Here, we formulate two different vehicle models, which are subsequently used to formulate the road-geometry model.

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 (heading) angle ψ describes the rotation of the vehicle frame O_E relative to the world frame O_W by the standard planar rotation matrix. Similarly, the road-aligned frame O_R is located on the left lane boundary, separated with a distance l_R from the camera frame O_C, which is rigidly connected to O_E with distance l_C.

A. Vehicle Models

There are three main categories of vehicle models; point-mass models, which represent the vehicle as a particle; kinematic models, which account for the geometry of the vehicle; and dynamic models, which account for the force balances, including the tire models, to more accurately capture the vehicle motion under (highly) dynamic maneuvers. Although a model based on force balances is generally more accurate than a kinematic model, the differences are small for regular driving [18]. Furthermore, a dynamic model depends on more parameters, such as the wheel radii, tire stiffness, and vehicle mass and inertia, which typically are unknown/uncertain and may be difficult, or at least tedious, to estimate [19]. To get a tractable estimation problem using only standard sensors, we make certain assumptions. Such assumptions are consistent with those that allow to model the vehicle dynamics by a single-track (i.e., bicycle) model, and have been shown to be valid in normal driving scenarios, see, for example, [20].

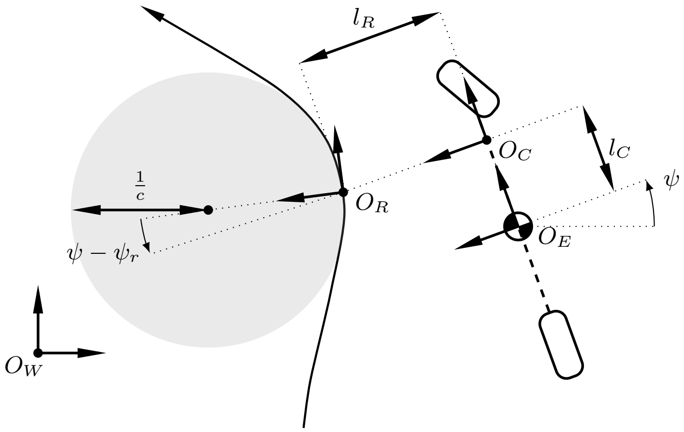

Fig. 1. The relation between the vehicle frame O_E, the camera frame O_C, the road frame O_R, and the world frame O_W. The distance between the vehicle's longitudinal x-axis and the lane boundary is l_R, and the shaded circle depicts the road curvature (here exaggerated) at the origin of O_R.

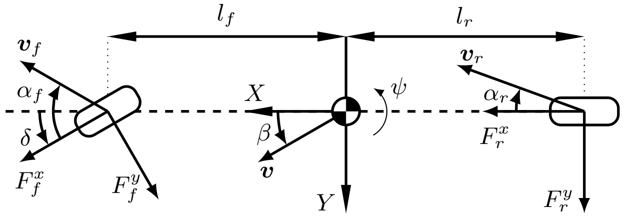

Fig. 2. A schematics of the single-track model and related notation.

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.*

Under these assumptions, we can use a planar single-track model depicted in Fig. 2.

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, $\boldsymbol{v}$ is the velocity vector, and subscripts f, r denote front and rear, respectively. With the state vector $\boldsymbol{x} = (p^X, p^Y, v^X, v^Y, \dot{\psi}) \in \mathbb{R}^{n_{\boldsymbol{x}}}$, $n_{\boldsymbol{x}} = 5$ 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

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

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

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

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 λ and the slip angle α,

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

where C_i^x, C_i^y are the longitudinal and lateral stiffness, respectively. The wheel slip is defined following [21], as

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

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

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

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

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

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

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

with input vector $\boldsymbol{u} = (\delta, \omega_f, \omega_r)$, and Gaussian zero-mean process noise, $\boldsymbol{w}_k^{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}^{\boldsymbol{x}})$ that accounts for model mismatch. The resulting model is accurate even for highly dynamic maneuvers. As will be shown later, it is also possible to exploit (6) in the road-prediction model. However, (6) requires the mass, inertia, and tire stiffness parameters to be known, which may be a drawback in practical applications.

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 $\boldsymbol{z} = (p^X, p^Y, \psi) \in \mathbb{R}^{n_{\boldsymbol{z}}}$, $n_{\boldsymbol{z}} = 3$, 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

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

where $L = l_f + l_r$, $\beta = \arctan(l_r \tan(\delta)/L)$ is the kinematic body-slop angle, and the velocity is related to the wheel speeds by $v^X = \frac{R_w}{2}(\omega_f + \omega_r)$. After time discretization, (7) is

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

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

B. Road-Geometry Models

When describing the road geometry, a critical component is the curvature, denoted by c. The road curvature can be defined in various ways. We define it as the curvature of the lane markings to the left of the ego vehicle (see Fig. 1). The road curvature is often approximated as a linear function

$$c(x_R) = c_0 + c_1 x_R, \tag{9}$$

where x_R is the position along the road in a road-aligned coordinate frame O_R, related to O_E through O_C (see Fig. 1).

Eq. (9) results in a clothoid expression of the road and is commonly used in automotive applications, and approximately agrees with road-construction principles [22]. Note that (9) is clearly violated in some situations, for example, when a part of the road is a straight line followed by a clothoidal stretch. However, for many situations (e.g., for highway or suburban driving) (9) is a good local approximation of the curvature.

Using (9), simple closed-form expressions of the road path can be found using small angle approximations. The lateral position of the road relative to the E frame can be written

$$y_R^E = l_R + x_R^E \tan(\delta_r) + \frac{c_0}{2} {x_R^E}^2 + \frac{c_1}{6} {x_R^E}^3, \tag{10}$$

which is a good polynomial approximation of the road, at least for prediction lengths less than 60–80m [1]. In (10), we let $\delta_r = \psi_R - \psi$ denote the difference between the road heading angle ψ_R and the vehicle heading angle ψ.

A commonly used model for the road curvature is

$$\dot{c}_0 = v^X c_1, \tag{11a}$$

$$\dot{c}_1 = 0. \tag{11b}$$

However, (11) does not utilize the vehicle model when predicting the road curvature.

In [2], a new road dynamic model was derived that uses the road geometry to calculate c_0 and δ_r. By defining

$$\delta_R = \delta_r - \beta, \tag{12}$$

that is, the difference between the road angle relative to the vehicle and the vehicle velocity vector, the road geometry model has the state vector $\boldsymbol{r} = (\delta_R; c_0; w) \in \mathbb{R}^{n_{\boldsymbol{r}}}$, $n_{\boldsymbol{r}} = 3$, where w is the lane width. Choosing between the dynamic and kinematic single-track model will not only have implications on the vehicle motion, but also on the road-prediction model.

1) Road Model with the Dynamic Single-Track Model: We briefly present the road-geometry model in [2], referring to it for additional details. A differential equation for δ_R is

$$\dot{\delta}_R = c_0 v^X - (\dot{\psi} + \dot{\beta}), \tag{13}$$

which follows from differentiating (12) and from geometrical relationships. From a approximation (i.e., $\beta \approx v^X/v^Y$), by substituting $\dot{\beta}$ with (1b) divided with v^X,

$$\dot{\delta}_R = c_0 v^X - \left(\frac{C_r^y l_r - C_f^y l_f}{(m v^X)^2}\right)\dot{\psi} - \left(\frac{C_f^y + C_r^y}{m (v^X)^2}\right) v^Y \\ + \left(\frac{C_f^y + \dot{v}^X m}{m v^X}\right)\delta, \tag{14}$$

which gives the dynamic equation for δ_R. Differentiating (13),

$$\ddot{\delta}_R = \dot{c}_0 v^X + c_0 \dot{v}^X - (\ddot{\psi} + \ddot{\beta}) \iff \dot{c}_0 = \frac{\ddot{\delta}_R + \ddot{\psi} + \ddot{\beta} - c_0 \dot{v}^X}{v^X}, \tag{15}$$

which gives a dynamic model for c_0. The premise of the dynamic single track model in (6) is that the vehicle operates in the linear region of the tire-force curve and that δ is small. It is therefore reasonable to set $\ddot{\delta}_R \approx 0$ and use small-angle approximations in (1). Hence,

$$\begin{aligned} \dot{c}_0 = \frac{1}{(I m^2 v^X)^4} & ((C_r^y)^2 (I + l_r^2 m)(-\dot{\psi} l_r + v^Y) \\ & + (C_f^y)^2 (I + l_f^2 m)(\dot{\psi} l_f + v^Y - \delta v^Y) \\ & + C_r^y I m(-3\dot{\psi}\dot{v}^X l_r + 3 v^Y \dot{v}^X + \dot{\psi}(v^X)^2) \\ & + \dot{v}^X I m^2 v^X (2\frac{v^Y \dot{v}^X}{v^X} + v^X(\dot{\psi} - c_0 v^X)) \\ & + C_f^y (C_r^y (I - l_r l_f m)(\dot{\psi} L - 2\dot{\psi} l_r + 2 v^Y - \delta v^X) \\ & + I m(3\dot{\psi}\dot{v}^X l_f + (3 v^Y - 2\delta \dot{v}^X) + (\dot{\delta} + \dot{\psi})(v^X)^2))), \end{aligned} \tag{16}$$

which follows by differentiating (1b) and inserting that together with (1c) into the second expression in (15), and using $\ddot{v}^X \approx 0$.[1] For the maneuvers in this paper, the difference between (16) and the exact, much longer expression that does not use any small-angle approximations, is negligible. After discretization with sampling period T_s and modeling the lane width w as a random walk, the resulting road-geometry model using the dynamic single-track model is

$$\boldsymbol{r}_{k+1} = \boldsymbol{a}(\boldsymbol{x}_k, \boldsymbol{r}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k^r, \tag{17}$$

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

2) Road Model with the Kinematic Single-Track Model: When using the kinematic single-track model, by inserting $\beta = \arctan(l_r \tan(\delta)/L)$ into (13),

$$\begin{aligned} \dot{\delta}_R &= c_0 v^X - (\dot{\psi} + \frac{\mathrm{d}}{\mathrm{d}t}\arctan(l_r \tan(\delta)/L)) \\ &= c_0 v^X - \dot{\psi} - \frac{L l_r}{L^2 \cos^2(\delta) + l_r^2 \sin^2(\delta)}\dot{\delta} \\ &\approx c_0 v^X - \dot{\psi} - \frac{l_r}{L}\dot{\delta}. \end{aligned} \tag{18}$$

From (15) and by again assuming $\ddot{\delta}_R \approx 0$,

$$\dot{c}_0 \approx \frac{\ddot{\psi} + \ddot{\beta} - c_0 \dot{v}^X}{v^X}, \tag{19}$$

where $\ddot{\psi}$ can be approximated using the yaw-rate measurement and, by using a small-angle approximation, $\ddot{\beta} \approx l_r/L\ddot{\delta}$. Clearly, this way of constructing the road prediction model is less precise than when utilizing the dynamic single-track model. On the other hand, the kinematic prediction model does not use knowledge of additional parameters, and for moderate driving maneuvers the discrepancies between the two models will be small. After discretization with a forward Euler scheme with sampling period T_s, the road-geometry model using the kinematic single-track model is

$$\boldsymbol{r}_{k+1} = \boldsymbol{b}(\boldsymbol{z}_k, \boldsymbol{r}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k^r, \tag{20}$$

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

Remark 1 *For a more compact notation, in the following we use $\bar{\boldsymbol{x}}$ to mean either $\bar{\boldsymbol{x}} = (\boldsymbol{x}; \boldsymbol{r})$ or $\bar{\boldsymbol{x}} = (\boldsymbol{z}; \boldsymbol{r})$. Similarly $\bar{\boldsymbol{f}}(\bar{\boldsymbol{x}}; \boldsymbol{u})$ denotes the functional relationships in* (6) *or* (8)*, often omitting $\boldsymbol{u}$ for brevity.*

C. Measurement Model

We consider the GNSS algorithm to output position measurements $\boldsymbol{y}^p$ generated by an estimator using code and carrier-phase measurements, for example, by the methods in [8], [10], [23]. We assume the position measurements to be Gaussian distributed and because the estimation quality will continuously change with environmental conditions and receiver movements, both the mean $\boldsymbol{\mu}^p$ and covariance $\boldsymbol{R}^p$ are considered to be time varying, resulting in $\boldsymbol{y}_k^p \sim \mathcal{N}(\boldsymbol{\mu}_k^p, \boldsymbol{R}_k^p)$. For simplicity but without loss of generality, we assume $\boldsymbol{y}^p \in \mathbb{R}^2$ in this work.

For use together with the dynamic single-track model, we utilize an IMU measuring the vehicle body frame longitudinal, $a^X = \dot{v}^X - v^Y\dot{\psi}$, and lateral, $a^Y = \dot{v}^Y + v^X\dot{\psi}$, acceleration and the yaw rate $\dot{\psi}$. The estimator uses the acceleration, a^X, a^Y, and yaw-rate $\dot{\psi}$ as measurements, forming the measurement vector $\boldsymbol{y}^a = (a^X; a^Y; \dot{\psi})$. Automotive-grade inertial sensors usually have a slowly time-varying bias, which must be modeled for any realistic implementation. For now, we assume that the bias has been predetermined offline, which is reasonable for experiments lasting only a few minutes. Future implementations will also address this.

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 the following measurement is available through the camera module,

$$\boldsymbol{y}^c = \begin{bmatrix} c_0 & \delta_r & w & l_R \end{bmatrix}^\top, \tag{21}$$

where l_R is the distance from the camera center position and the road, see Fig. 1. This results in the measurement equation

$$\boldsymbol{h}^c = \begin{bmatrix} c_0 & \delta_R + \beta & w & l_R \end{bmatrix}^\top. \tag{22}$$

The camera measurements are assumed Gaussian distributed according to $\boldsymbol{y}_k^c \sim \mathcal{N}(\boldsymbol{\mu}_k^c, \boldsymbol{R}_k^c)$, where, similarly to the GNSS

[1] Note that (16) differs slightly from the corresponding equation in [2], which included a typo.

measurements, both the mean and covariance are time varying. In summary, the complete measurement model is

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{e}_k \in \mathbb{R}^{n_{\boldsymbol{y}}}, \tag{23}$$

where $\boldsymbol{y} = (\boldsymbol{y}^p; \boldsymbol{y}^a; \boldsymbol{y}^c) \in \mathbb{R}^9$ for the dynamic single-track model (6) and $\boldsymbol{y} = (\boldsymbol{y}^p; \boldsymbol{y}^c) \in \mathbb{R}^6$ for the kinematic single-track model (8), and $\boldsymbol{e}$ is zero-mean Gaussian distributed with a block-diagonal covariance matrix.

Remark 2 *What the vision system measures and outputs varies according to the implementation. The notation we use here is the same as in [2]. However, in [3], the vision system outputs the coefficients from a third-order polynomial. Our method can be modified to support such changes.*

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, and an analytic solution to the estimation problem does not exist. While an extended Kalman filter (EKF) may work, the estimation model has significant nonlinearities for which a first-order linearization may be insufficient. However, owing 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 as

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

by its first two moments. Given the assumed Gaussian filtering posterior (24) at time step k, the distribution of the state prediction at time step $k+1$ is approximated by a Gaussian,

$$p(\bar{\boldsymbol{x}}_{k+1} | \bar{\boldsymbol{x}}_k, \boldsymbol{y}_{0:k}) \approx \mathcal{N}\left(\bar{\boldsymbol{x}}_{k+1} | \hat{\bar{\boldsymbol{x}}}_{k+1|k}, \boldsymbol{P}_{k+1|k}\right), \tag{25}$$

by direct evaluation of the associated moment integrals

$$\hat{\bar{\boldsymbol{x}}}_{k+1|k} = \int \bar{\boldsymbol{f}}(\bar{\boldsymbol{x}}_k) p(\bar{\boldsymbol{x}}_k | \boldsymbol{y}_{0:k}) \, \mathrm{d}\bar{\boldsymbol{x}}_k, \tag{26a}$$

$$\boldsymbol{P}_{k+1|k} = \int \big(\bar{\boldsymbol{f}}(\bar{\boldsymbol{x}}_k) - \hat{\bar{\boldsymbol{x}}}_{k+1|k}\big)(\star)^\top p(\bar{\boldsymbol{x}}_k | \boldsymbol{y}_{0:k}) \, \mathrm{d}\bar{\boldsymbol{x}}_k + \boldsymbol{Q}_k, \tag{26b}$$

here simplified by the assumptions on $\boldsymbol{w}_k$. By insertion of the approximation in (24), this becomes equivalent to evaluating two Gaussian integrals. For a general $\bar{\boldsymbol{f}}$, no closed-form solutions exist, but numerical integration methods also known as *cubature rules*, can be employed [24]. To facilitate this, we transform the coordinates $\boldsymbol{\xi}_k = \boldsymbol{L}_{k|k}^{-1}(\bar{\boldsymbol{x}}_k - \hat{\bar{\boldsymbol{x}}}_{k|k})$, using the Cholesky factorization of the covariance matrix $\boldsymbol{P}_{k|k} = \boldsymbol{L}_{k|k}\boldsymbol{L}_{k|k}^\top$. The LRKFs approximate the transformed integrals by evaluating the nonlinearity $\bar{\boldsymbol{f}}$ in a set of integration points $\mathcal{P} = \{\omega^i, \boldsymbol{\xi}^i\}_{i=1}^{|\mathcal{P}|}$, where $|\mathcal{P}|$ denotes the total number of points used. Hence, for each such point $\boldsymbol{\xi}^i$ and the filtered mean estimate $\hat{\bar{\boldsymbol{x}}}_{k|k}$, we compute

$$\hat{\bar{\boldsymbol{x}}}_{k+1|k}^i = \bar{\boldsymbol{f}}\left(\hat{\bar{\boldsymbol{x}}}_{k|k} + \boldsymbol{L}_{k|k}\,\boldsymbol{\xi}^i\right), \tag{27}$$

subsequently approximating the moment integrals in (26a) as

$$\hat{\bar{\boldsymbol{x}}}_{k+1|k} \approx \sum_{i=1}^{|\mathcal{P}|} \omega^i \hat{\bar{\boldsymbol{x}}}_{k+1|k}^i, \tag{28a}$$

$$\boldsymbol{P}_{k+1|k} \approx \sum_{i=1}^{|\mathcal{P}|} \omega^i (\hat{\bar{\boldsymbol{x}}}_{k+1|k}^i - \hat{\bar{\boldsymbol{x}}}_{k+1|k})(\star)^\top. \tag{28b}$$

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

$$\begin{aligned} p\big((\bar{\boldsymbol{x}}_{k+1}; \boldsymbol{y}_{k+1}) | \bar{\boldsymbol{x}}_k, \boldsymbol{y}_{0:k}\big) \\ \approx \mathcal{N}\left(\begin{bmatrix} \hat{\bar{\boldsymbol{x}}}_{k+1|k} \\ \hat{\boldsymbol{y}}_{k+1|k} \end{bmatrix} \begin{bmatrix} \boldsymbol{P}_{k+1|k} & \boldsymbol{P}_{k+1|k}^{\bar{\boldsymbol{x}}\boldsymbol{y}} \\ \boldsymbol{P}_{k+1|k}^{\boldsymbol{y}\bar{\boldsymbol{x}}} & \boldsymbol{P}_{k+1|k}^{\boldsymbol{y}\boldsymbol{y}} \end{bmatrix}\right). \end{aligned} \tag{29}$$

Conditioning of the joint density in (29) on the new measurement $\boldsymbol{y}_{k+1}$ amounts to the usual Kalman filter update

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

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

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

For the numerical examples, we use an LRKF with point set in Definition 1. Other integration rules can be used without modifications to the underlying method.

Definition 1 *The spherical cubature (SC) rule, as used in the cubature Kalman filter (CKF) [25], consists of $|\mathcal{P}| = 2n$ with*

$$\boldsymbol{\Xi} = \sqrt{n}\begin{bmatrix} \boldsymbol{1}_{n\times n} & -\boldsymbol{1}_{n\times n} \end{bmatrix}, \quad \boldsymbol{\Omega} = \frac{1}{2n}\boldsymbol{1}_{2n}^\top, \tag{31}$$

with $n = n_{\bar{\boldsymbol{x}}}$. The integration point $\boldsymbol{\xi}^i$ and weight ω^i correspond to the ith column of $\boldsymbol{\Xi}$ and element of $\boldsymbol{\Omega}$, respectively.

B. Interacting Multiple-Model LRKF

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 LRKF in an IMM framework [26], [27], in which we have a set of m models that differ only 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,

$$\bar{\boldsymbol{x}}_{k+1} = \bar{\boldsymbol{f}}(\bar{\boldsymbol{x}}_k; \boldsymbol{u}_k) + \boldsymbol{w}_k, \qquad \boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k), \tag{32a}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\bar{\boldsymbol{x}}_k, \boldsymbol{u}_k) + \boldsymbol{e}_k(\theta_k), \;\; \boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k^{\theta_k}), \tag{32b}$$

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 $\{\boldsymbol{R}^{\theta_k} \in \mathbb{R}^{n_e \times n_e} | \boldsymbol{R}^{\theta_k} = (\boldsymbol{R}^{\theta_k})^\top, \boldsymbol{R}^{\theta_k} \succ \boldsymbol{0}\}_{\theta_k = 1}^m$.

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

$$\bar{q}^i_k = \sum_{j=1}^{m} \Pi_{ij} q^j_{k-1}, \tag{33a}$$

$$\hat{\bar{\boldsymbol{x}}}^i_{k-1|k-1} = \sum_{j=1}^{m} \Pi_{ij} \frac{q^j_{k-1}}{\bar{q}^i_k} \hat{\bar{\boldsymbol{x}}}^j_{k-1|k-1}, \tag{33b}$$

$$\begin{aligned} \boldsymbol{P}^i_{k-1|k-1} = \sum_{j=1}^{m} \Pi_{ij} \frac{q^j_{k-1}}{\bar{q}^i_k} \Big(& \boldsymbol{P}^j_{k-1|k-1} + \\ & (\hat{\bar{\boldsymbol{x}}}^j_{k-1|k-1} - \hat{\bar{\boldsymbol{x}}}^i_{k-1|k-1})(\star)^\top \Big). \end{aligned} \tag{33c}$$

Next, we execute a filter bank of m LRKFs to find the estimate of $\bar{\boldsymbol{x}}_k$, 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 m components,

$$\begin{aligned} p(\bar{\boldsymbol{x}}_k|\boldsymbol{y}_{0:k}) &= \sum_{j=1}^{m} p(\bar{\boldsymbol{x}}^j_k|\boldsymbol{y}_{0:k}) = \sum_{j=1}^{m} \frac{p(\boldsymbol{y}_k|\bar{\boldsymbol{x}}^j_k) p(\bar{\boldsymbol{x}}_k|\boldsymbol{y}_{0:k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{0:k-1})} \\ &\approx \sum_{j=1}^{m} q^j_k \mathcal{N}(\bar{\boldsymbol{x}}^j_k|\hat{\bar{\boldsymbol{x}}}^j_{k|k-1}, \boldsymbol{P}^j_{k|k-1}), \end{aligned} \tag{34}$$

where

$$q^j_k \propto p(\boldsymbol{y}_k|\bar{\boldsymbol{x}}^j_k) = \mathcal{N}\Big(\boldsymbol{y}_k|\hat{\boldsymbol{y}}^j_{k|k-1}, \boldsymbol{P}^{\boldsymbol{yy},j}_{k|k-1}\Big)\bar{q}^j_k, \quad \forall j \in [1, m], \tag{35}$$

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

$$\hat{\bar{\boldsymbol{x}}}_{k|k} = \sum_{j=1}^{m} q^j_k \hat{\bar{\boldsymbol{x}}}^j_{k|k}, \tag{36a}$$

$$\boldsymbol{P}_{k|k} = \sum_{j=1}^{m} q^j_k \left(\boldsymbol{P}^j_{k|k} + (\hat{\bar{\boldsymbol{x}}}^j_{k|k} - \hat{\bar{\boldsymbol{x}}}_{k|k})(\star)^\top \right). \tag{36b}$$

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

IV. Simulation Study

In this section we validate the proposed IMM method in a Monte-Carlo simulation study. To generate the synthetic data, we consider a car modeled by the dynamic single-track model (6) in closed loop with a reference tracking controller driving in the vicinity of Boston, USA. The route is extracted using the open-source routing machine (OSRM) tool [28]. This map is represented by a sequence of points. To generate a smooth map and determine the ground-truth camera measurements, we low-pass filterered the points in the map, and determined the measurements using standard methods of optimization.[2]

Algorithm 1 Pseudo-code of proposed IMM algorithm

```
Initialize: $\{\boldsymbol{\xi}^i, \omega^i\}_{i=1}^{|\mathcal{P}|}, \{\hat{\bar{\boldsymbol{x}}}^j_{-1|-1}, \boldsymbol{P}^j_{-1|-1}, \boldsymbol{R}^j, q^j_{-1}\}_{j=1}^m, \boldsymbol{\Pi}$
for $k = 0, 1, \dots$ do
    // IMM mixing
    for $i \in \{1, \dots, m\}$ do
        Mix estimates according to (33).
    end for
    for $j \in \{1, \dots, m\}$ do
        // LRKF time update
        for $i \in \{1, \dots, |\mathcal{P}|\}$ do
            Determine $\hat{\bar{\boldsymbol{x}}}^i_{k|k-1}$ according to (27).
        end for
        Determine $\hat{\bar{\boldsymbol{x}}}^j_{k|k-1}$, $\boldsymbol{P}^j_{k|k-1}$ according to (28a).
        // LRKF measurement update
        for $i \in \{1, \dots, |\mathcal{P}|\}$ do
            Determine $\hat{\boldsymbol{y}}^i_{k|k-1}$ akin to (27)
        end for
        // IMM weight update
        Determine $\hat{\boldsymbol{y}}^j_{k|k-1}$, $\boldsymbol{P}^{\boldsymbol{yy},j}_{k|k-1}$, $\boldsymbol{P}^{\bar{\boldsymbol{x}}\boldsymbol{y},j}_{k|k-1}$ akin to (28a).
        Determine $\hat{\bar{\boldsymbol{x}}}^j_{k|k}$, $\boldsymbol{P}^j_{k|k}$ using (30).
        Determine $q^j_k$ according to (35).
    end for
    // Form weighted mean
    Determine $\hat{\bar{\boldsymbol{x}}}_{k|k}$, $\boldsymbol{P}_{k|k}$ according to (36).
end for
```

We have set the IMU measurement noise comparable to the noise for a low-cost IMU, and the GNSS position measurements nominally provide Gaussian zero-mean measurements with standard deviation 0.5m in both X and Y direction. Furthermore, the camera measurements provide lane measurements that nominally are Gaussian distributed according to $\boldsymbol{y}^c \sim \mathcal{N}(\boldsymbol{h}^c(\bar{\boldsymbol{x}}), \boldsymbol{R}_c)$, $\boldsymbol{R}_c = \text{diag}(0.1\pi/180, 10^{-5}, .05, .05)^2$. When executing Algorithm 1 using the kinematic single-track model (8), only GNSS and camera measurements are used. For each of the Monte-Carlo runs, we generate the initial state by sampling it from some initial distribution with 5m initial uncertainty on the position, in both the X and Y direction. 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 three different models:

1) $\boldsymbol{R}^p = \boldsymbol{R}^p_{nom}, \boldsymbol{R}^c = \boldsymbol{R}^c_{nom}$;
2) $\boldsymbol{R}^p = \boldsymbol{R}^p_{nom}, \boldsymbol{R}^c = 50^2\boldsymbol{R}^c_{nom}$;
3) $\boldsymbol{R}^p = 10^2\boldsymbol{R}^p_{nom}, \boldsymbol{R}^c = \boldsymbol{R}^c_{nom}$;

where $\boldsymbol{R}^c_{nom}$ denotes the nominal camera covariance matrix and $\boldsymbol{R}^p_{nom}$ is the nominal GNSS covariance matrix. Hence, Model 1 is correct when no outliers occur, Model 2 is correct when camera outliers occur, and Model 3 is correct when GNSS outliers occur. Note that the chosen values in a real experiment should be determined based on actual data.

We inject the GNSS outlier measurements after 5s, lasting for 3s, and repeating every tenth second. Similarly, we inject camera outlier measurements after 10s, lasting for 3s, and repeating every tenth second. We use the three modes in the

[2] The filtered map position is C^2 and known: whereby the curvature $c(x^R)$ is known analytically; $\{c_0, c_1\}$ can be computed by a linear regression at any point in the map; and the distance l_R is found by a simple Newton method.

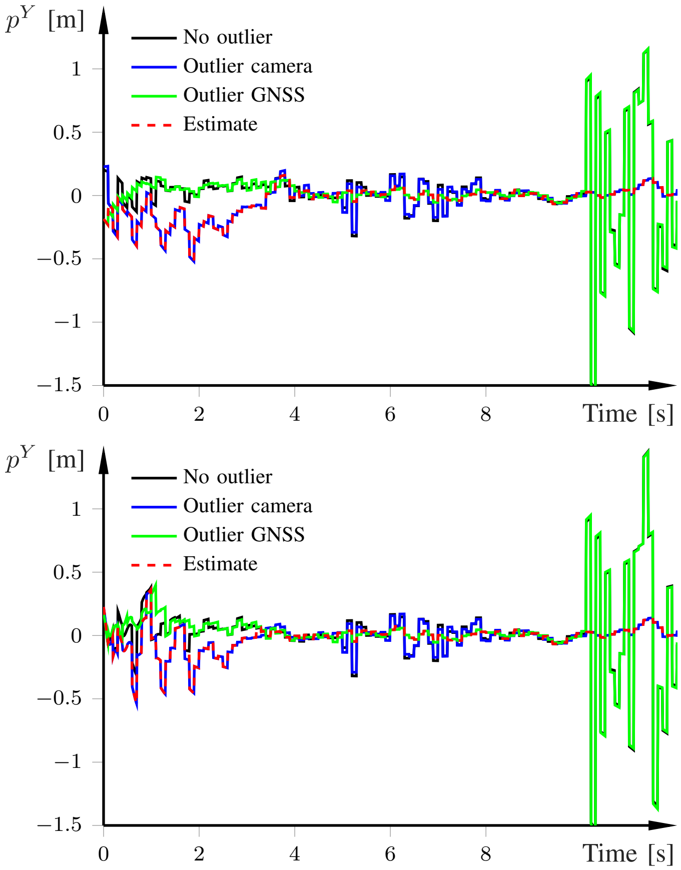

Fig. 3. Excerpt of the position error of the four (three models plus weighted average) compared combinations with occasional outliers for one realization. Results using the dynamic single-track model (upper) and kinematic single-track model (lower) in the estimation.

IMM, and the transition probability matrix is set to

$$\mathbf{\Pi} = \begin{bmatrix} 0.95 & 0.025 & 0.025 \\ 0.225 & 0.75 & 0.025 \\ 0.225 & 0.025 & 0.75 \end{bmatrix}, \tag{37}$$

to reflect that it is more likely for inliear measurements to stay inliers than it is for outliers to stay outliers. The vehicle drives on Interstate-90 in the suburbs of Boston, MA, with the reference velocity 25m/s (56mph).

A. Result with Outliers: One Realization

Fig. 3 shows the position error in the global Cartesian Y direction for the first ten seconds of one realization for the three 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. The position estimate of the proposed method shows the smallest position error for most of the time, that is, it correctly weighs together the different estimators in the IMM to produce the best possible estimate. This is clearly seen between 5–8s, when the model using camera outlier noise covariance gives an estimation error of more than 3m, which the IMM discards when producing the estimates. The first ten seconds of the realization are shown, but the results are similar across the full realization.

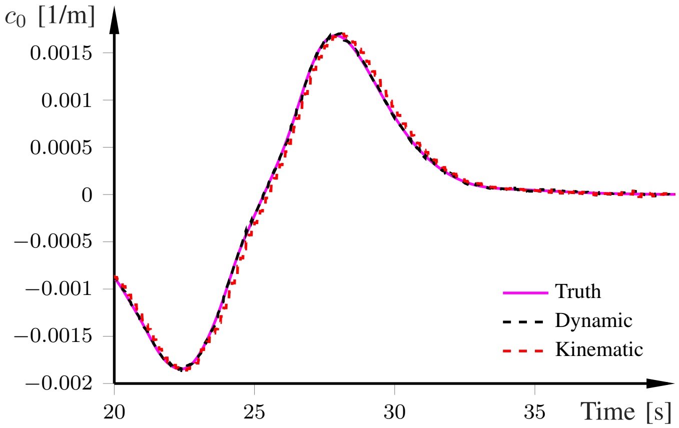

Fig. 4. Curvature estimation results for one on the MC realizations.

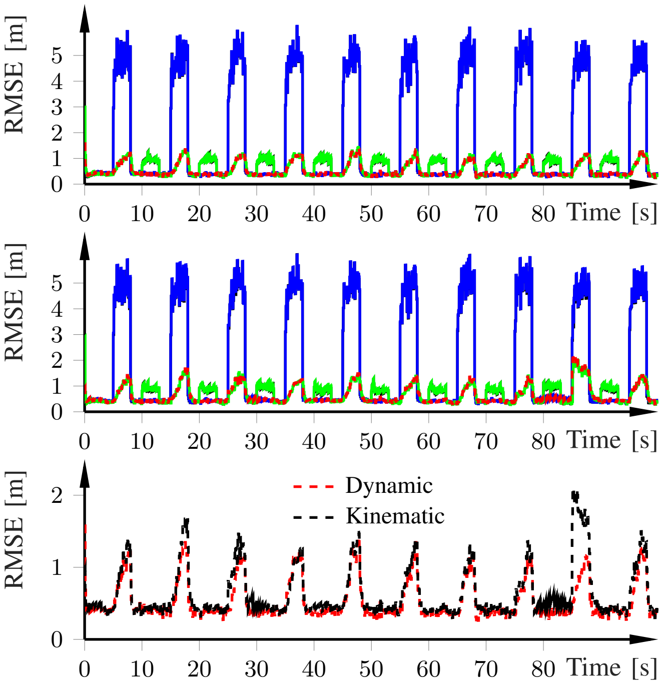

Fig. 5. Position RMSE with the dynamic (upper) and kinematic (middle) single-track model. The lower plot shows a comparison of the RMSE between the two vehicle models. The two upper plots use the same coloring as in Fig. 3.

Fig. 4 displays the curvature estimation results using Algorithm 1 for a period of time where the curvature change is significant. The estimation results are slightly improved when using the dynamic single-track model. The kinematic single-track model has a substantially simpler prediction model, which results in slightly inferior prediction capabilities.

B. Result with Outliers: Monte-Carlo Study

Fig. 5 shows the position RMSE for the three models plus the weighted average when using the dynamic single-track model (upper plot), the same when using the kinematic model (middle plot), and the comparison between the dynamic and kinematic single-track model (lower plot). Clearly, irrespective of which model is used, the IMM correctly weighs together the models. For this road type (highway) and reference velocity

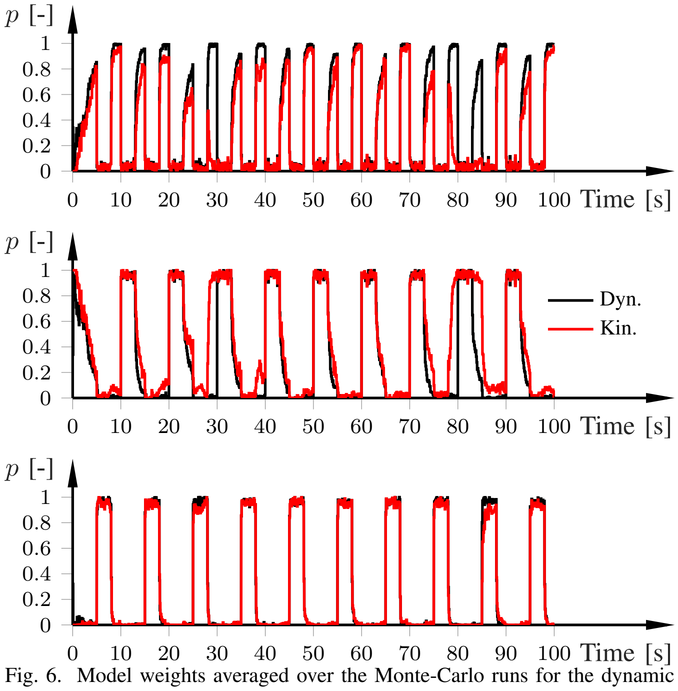

Fig. 6. Model weights averaged over the Monte-Carlo runs for the dynamic single-track (black) and kinematic single-track (red).

(25m/s), there is some benefit of using the dynamic single-track model. However, this performance increase should be considered in combination with the increased complexity of using the dynamic single-track model in terms of the number of parameters needed to be estimated a priori.

Fig. 6 displays the probabilities of each model weighted over the Monte-Carlo runs for both the dynamic and kinematic single-track model. On average, the correct model is found at each time step in the sense that it gets the highest weight. Relating to the estimation results in the lowet plot in Fig. 5, the slight performance increase when using the dynamic single-track model in the estimation is reflected by the slightly larger assigned probabilites to the correct model.

V. Conclusion

This paper presented an LRKF-based IMM estimator, where each model has a different noise covariance structure, that adaptively fuses GNSS and camera information to find the best combination of noise covariance that explains the measurements. The method can output estimates of the vehicle state, as well as the road state. Using a Monte-Carlo study, the results indicate that the method correctly determines the best model to explain the measurements.

We provided estimation results using two different vehicle models. The dynamic single-track model seemingly produces slightly more accurate estimates for the scenario used in this paper. However, the difference may be small, and should be considered in combination with the increased complexity compared to the kinematic single-track model.

References

[1] 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.

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

[3] 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.

[4] 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.

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

[6] A. Leick, L. Rapoport, and D. Tatarnikov, *GPS satellite surveying*. John Wiley & Sons, 2015.

[7] P. Teunissen, P. De Jonge, and C. Tiberius, “The LAMBDA method for fast GPS surveying,” in *Int. Symp. GPS Technology Applications*, Bucharest, Romania, 1995.

[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] P. J. Teunissen, “GPS carrier phase ambiguity fixing concepts,” in *GPS for Geodesy*. Springer, 1998, pp. 319–388.

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

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

[15] 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.

[16] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles,” *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, 2016.

[17] K. Berntorp, T. Hoang, R. Quirynen, and S. Di Cairano, “Control architecture design of autonomous vehicles,” in *Conf. Control Technol. and Applications*, Copenhagen, Denmark, Aug. 2018, invited paper.

[18] A. Carvalho, S. Lefévre, 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.

[19] 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.

[20] R. Rajamani, *Vehicle Dynamics and Control*. Springer-Verlag, 2006.

[21] E. Schindler, *Fahrdynamik: Grundlagen Des Lenkverhaltens Und Ihre Anwendung Für Fahrzeugregelsysteme*. Renningen, Germany: Expert-Verlag, 2007.

[22] “Vägutformning 94 version s-2,” Swedish National Road Administration, Tech. Rep., 1994.

[23] M. Greiff and K. Berntorp, “Optimal measurement projections with adaptive mixture Kalman filtering for GNSS positioning,” in *American Control Conference*, 2020.

[24] J. Steinbring and U. D. Hanebeck, “S 2 KF: The smart sampling Kalman filter,” in *16th Int. Conf. Information Fusion*, Istanbul, Turkey, Jul. 2013.

[25] I. Arasaratnam, “Cubature Kalman filtering theory & applications,” Ph.D. dissertation, McMaster University, 2009.

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

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

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