

Noise-Statistics Learning of Automotive-Grade Sensors Using Adaptive Marginalized Particle Filtering

Berntorp, K.; Di Cairano, S.

TR2018-181 December 29, 2018

Abstract

This paper presents a method for real-time identification of sensor statistics especially aimed for low-cost automotive-grade sensors. Based on recent developments in adaptive particle filtering and under the assumption of Gaussian distributed noise, our method identifies the slowly time-varying sensor offsets and variances jointly with the vehicle state, and it extends to banked roads. While the method is primarily focused on learning the noise characteristics of the sensors, it also produces an estimate of the vehicle state. This can then be used in driver-assistance systems, either as a direct input to the control system, or indirectly to aid other sensor-fusion methods. The paper contains verification against several simulation and experimental data sets. The results indicate that our method is capable of bias-free estimation of both the bias and variance of each sensor, that the estimation results are consistent over different data sets, and that the computational load is feasible for implementation on computationally limited embedded hardware typical of automotive applications.

Journal of Dynamic Systems, Measurement, and Control

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.

Noise-Statistics Learning of Automotive-Grade Sensors Using Adaptive Marginalized Particle Filtering

Karl Berntorp¹ and Stefano Di Cairano^{1*}

This paper presents a method for real-time identification of sensor statistics especially aimed for low-cost automotive-grade sensors. Based on recent developments in adaptive particle filtering and under the assumption of Gaussian distributed noise, our method identifies the slowly time-varying sensor offsets and variances jointly with the vehicle state, and it extends to banked roads. While the method is primarily focused on learning the noise characteristics of the sensors, it also produces an estimate of the vehicle state. This can then be used in driver-assistance systems, either as a direct input to the control system, or indirectly to aid other sensor-fusion methods. The paper contains verification against several simulation and experimental data sets. The results indicate that our method is capable of bias-free estimation of both the bias and variance of each sensor, that the estimation results are consistent over different data sets, and that the computational load is feasible for implementation on computationally limited embedded hardware typical of automotive applications.

1 Introduction

Production vehicles are typically equipped with low-cost sensors that are prone to time-varying offset and scale errors, and may furthermore have relatively low signal-to-noise ratio [1]. For instance, the lateral acceleration and heading-rate measurements are known to have significant drift and noise in the sensor measurements, leading to measurements that are only reliable for prediction over a very limited time interval. Similarly, the sensor measuring the steering-wheel angle has an offset error that, when used for dead reckoning in a vehicle model, leads to prediction errors that accumulate over time.

Some of the active safety systems developed in the past (e.g., electronic stability control [2] and anti-lock braking systems [3]) have focused on aiding the driver over relatively short time intervals. However, according to the survey [1], even for those safety systems, accurate estimation may be more important than advanced control algorithms. The recent surge for enabling new autonomous capabilities [4–7] implies a need for sensor information that can be used over longer time intervals to reliably predict the vehicle motion.

The underlying theme of how to achieve more reliable sensor information is to leverage sensor fusion, to utilize existing low-cost sensors as efficiently as possible for as many purposes and driver-assistance features as possible.

Offset estimation methods for the steering wheel and yaw rate found in production vehicles are typically based on averaging, to compensate for the yaw rate and steering wheel bias. However, this leads to performance that is sometimes more than one order of magnitude away from the requirements imposed by the next-generation advanced driver-assistance systems (ADAS) [8]. Various methods have been proposed to improve the offset compensation in the steering-wheel angle and/or inertial measurements. The method in [9] estimates the yaw-rate offset in a state-augmented Kalman filter based on a kinematic vehicle model, where the yaw-rate offset is modeled as a random walk. The approach in [8] extends this to also include estimation of the steering offset in a linear regression. Oftentimes, the bias of the inertial sensors are modeled as a random walk, integrated, and solved for in an estimation algorithm targeted for a specific application [1, 10–12]. However, this results in repeating the same tasks in different filters, with the consequence of unnecessarily repeated computations. Also, it implies that each estimator may have redundant components, which might have implications on observability and feasibility of the approach.

In this paper,¹ we develop a method for real-time learning of the sensor-noise statistics of the acceleration, gyro, steering-wheel, and (optionally) the roll-rate measurements. While our primary focus is the lateral vehicle dynamics, the established framework can be applied to either lateral or longitudinal dynamics, or to the two combined. We model the sensor measurements as Gaussian random variables with unknown mean and covariance, and the task is to learn these unknown quantities in real time. As opposed to Kalman-type filtering methods, our method naturally estimates not only the bias but also the variance of the sensor, that is, without the need to augment the state vector, which increases the problem dimensionality. The results from our method can then be used in other sensor-fusion methods to support the vehicle control functions.

*¹Karl Berntorp and Stefano Di Cairano are with Mitsubishi Electric Research Laboratories (MERL), 02139 Cambridge, MA, USA. Email:{karl.o.berntorp, dicairano}@ieee.org

¹A preliminary version of this work was presented in [13]. The current, elaborated, version contains a more detailed explanation of the estimation algorithm development and extension to road-bank estimation, and a significantly expanded experimental evaluation based on several datasets.

The vehicle dynamics and the measurements are described by a state-space model, parametrized by the unknown mean and covariance. The resulting estimation problem is non-Gaussian and includes both the vehicle state trajectory and the parameters, which introduces dynamic coupling between state and parameters and further necessitates approximate methods. We use particle filtering (PF) [14] for solving our non-Gaussian estimation problem. PFs have previously been used in several automotive applications (see, e.g., [15, 16]). PFs generally provide asymptotic performance guarantees. A common way to estimate slowly time-varying parameters such as sensor offsets, is to augment the state vector [10–12, 17]. However, this leads to an increased state dimension that may be problematic for particle filters, since the number of propagated particles, and hence the computational burden, increases exponentially with the state-space dimension, and the computational capabilities of automotive micro-controllers that run the estimation algorithm are limited with respect to other applications of PF, such as avionics and target tracking. Instead, we rely on marginalization [18] and propagation of the sufficient statistics of the noise parameters, conditioned on the estimated vehicle states, by exploiting the concept of conjugate priors [19]. We use similar concepts as in [20], but adapt our algorithm to a different type of noise dependency. Furthermore, we handle the road bank by executing a Rao-Blackwellized particle filter (RBPF), in which the road-bank is estimated by a set of Kalman filters (KFs), each KF conditioned on a state trajectory and a set of noise parameters.

Outline: Sec. 2 presents the vehicle model, sensor setup, and the problem definition. Our proposed approach is described in Sec. 3. Secs. 4 and 5 present the simulation and experimental evaluations, respectively, and the paper is concluded in Sec. 6.

Notation: With $p(\mathbf{x}_{0:k}|\mathbf{y}_{0:k})$, we mean the posterior density function of the state trajectory $\mathbf{x}_{0:k}$ from time index 0 to time index k given the measurement sequence $\mathbf{y}_{0:k} := \{\mathbf{y}_0, \dots, \mathbf{y}_k\}$. We write \mathbf{f}_k for a function $\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$, where \mathbf{u} is the deterministic input. Throughout, for a vector \mathbf{x} , $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicates that \mathbf{x} is Gaussian distributed with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and $|\boldsymbol{\Sigma}|$ is the determinant of the matrix $\boldsymbol{\Sigma}$. The notation $\text{St}(\boldsymbol{\mu}, \Upsilon, \nu)$ denotes the multivariate Student-t distribution with mean $\boldsymbol{\mu}$, scaling Υ , and ν degrees of freedom. Similarly, $\text{NiW}(\gamma, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \nu)$ denotes the Normal-inverse-Wishart distribution with statistics (hyperparameters) summarized in $(\gamma, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \nu) = \mathcal{S}$. The notation $\hat{\mathbf{z}}_{k|m}$ denotes the estimate of \mathbf{z} at time index k given measurements up to time index m .

2 Modeling and Problem Formulation

Our algorithm aims at estimating the offsets during normal driving. We therefore model the vehicle dynamics by a single-track (i.e., bicycle) model [21, 22], in which the two wheels on each axle are lumped together, where the vehicle operates in the linear region of the tire-force curve. We incorporate modeling of the influence of the bank angle into the single-track model to correct for rotational disturbances

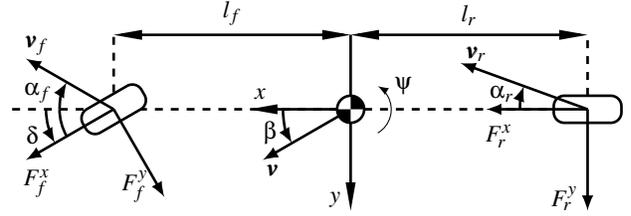


Fig. 1. A schematics of the single-track model and related notation.

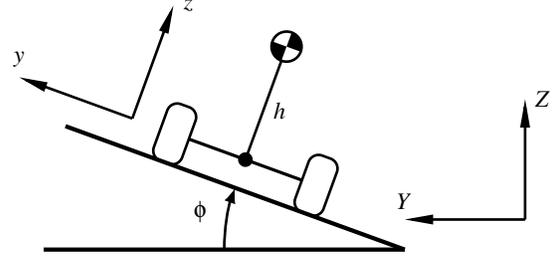


Fig. 2. Bank-angle model.

from the road. This paper focuses on the sensors mainly related to the lateral vehicle dynamics, but our approach can also handle the combined longitudinal and lateral setting by considering the corresponding sensors, and can be extended to incorporate estimation of the road grade.

Fig. 1 provides a schematic of the single-track model. In what follows, F^y is the lateral tire force, α is the wheel-slip angle, ψ is the yaw, δ is the steering angle of the front wheel at the road and subscripts f, r denote front and rear, respectively. The state vector is $\mathbf{x} = [v^y \ \dot{\psi}]^T$, where v^y is the lateral velocity of the vehicle and $\dot{\psi}$ is the yaw rate. From the assumption of driving in the linear regime of the tire-force curve, the lateral tire force can be expressed as a linear function of the slip angle α , $F^y \approx C^y \alpha$, where C^y is the lateral stiffness. The slip angles are approximated as

$$\alpha_f \approx \delta - \frac{v^y + l_f \dot{\psi}}{v^x}, \quad \alpha_r \approx \frac{l_r \dot{\psi} - v^y}{v^x}, \quad (1)$$

where l_r and l_f are the distance from the center of mass to the front and rear wheel, respectively. In (1), we use the velocity at the center of mass instead of the velocity at the center of the wheel. Fig. 2 illustrates the influence of the road-bank angle ϕ . We assume that the effects on the vehicle dynamics from the road-bank angle ϕ are considerably larger than the effects on the vehicle dynamics due to the roll angle. This assumption usually holds under normal driving and/or for vehicles with low center of mass h . The equations of motion of the single-track model including effects of the bank angle are

$$m\dot{v}^y = -mv^x\dot{\psi} + C_f^y \left(\delta - \frac{v^y + l_f\dot{\psi}}{v^x} \right) + C_r^y \frac{l_r\dot{\psi} - v^y}{v^x} - \frac{g}{v^x} \phi, \quad (2a)$$

$$I\dot{\psi} = l_f C_f^y \left(\delta - \frac{v^y + l_f\dot{\psi}}{v^x} \right) - l_r C_r^y \frac{l_r\dot{\psi} - v^y}{v^x}, \quad (2b)$$

where m is the vehicle mass, I is the inertia, g is the gravitational acceleration, and we have used the small-angle approximation $\sin\phi \approx \phi$. Model (2) is nonlinear in v^x and there are bilinearities between states and parameters. In this paper we focus on the lateral dynamics and therefore the longitudinal velocity v^x is assumed known in the estimator. This is consistent with many navigation systems, where dead reckoning is used to decrease the state dimension. In this work, we determine v^x from the wheel rotation rates given by the wheel-speed sensors and assuming a known tire radius (see, e.g., [16] for tire-radius estimation).

2.1 Estimation Model

The estimation model consists of the vehicle dynamics model (2) and the road bank and sensor models, which will be introduced next. The road-bank angle ϕ depends on the road geometry. The road changes are independent on the vehicle dynamics and furthermore vary with the specific road the vehicle travels on. However, the road banking changes smoothly, implying that $\dot{\phi}$ is continuous. We therefore model the changes of the road-bank angle as white noise on $\dot{\phi}$, that is, a (nearly) constant acceleration model with Gaussian distributed noise $w_\phi \sim \mathcal{N}(0, \sigma_\phi)$ with known variance σ_ϕ ,

$$\mathbf{z}_{k+1} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}_\phi} \mathbf{z}_k + \underbrace{\begin{bmatrix} T_s^3 \\ 6 \\ T_s^2 \\ 2 \\ T_s \end{bmatrix}}_{\mathbf{B}_\phi} w_{\phi,k}, \quad (3)$$

where $\mathbf{z} = [\phi \ \dot{\phi} \ \ddot{\phi}]^T$. The ability to capture rapid changes in the road bank will depend on the choice of σ_ϕ . Hence, if there is prior knowledge about the environment, σ_ϕ can be adjusted to reflect this.

The sensor measuring the steering angle is usually placed on the steering wheel. Hence, the steering angle δ at the wheel is not directly measured. Furthermore, the Ackermann steering configuration causes a slight deviation between the left and the right wheel. We assume a single-track model (Fig. 1), where δ is modeled as the average between the left and right wheel angles. In general, δ can be calculated from a static map of the measured steering-wheel angle. However, the resulting measurement of δ is known to be subject to an offset, for instance due to the use of relative encoders that are cheaper and more robust to the mounting process than absolute encoders, which in some cases can even

be time varying. An objective with the present contribution is therefore to estimate such an offset, which is important for proper operation of certain ADAS functions, such as the Electronic Stability Control (ESC) [2]. To this end, we decompose the steering angle into one known nominal part and one unknown part,

$$\delta = \delta_m + \Delta\delta, \quad (4)$$

where δ_m is the measured value of the steering angle, and where $\Delta\delta$ is the, possibly time-varying, offset. We model

$$w_k := \Delta\delta \quad (5)$$

as random process noise acting on the otherwise deterministic vehicle dynamics. The noise term w_k is modeled as Gaussian distributed according to $w_k \sim \mathcal{N}(\mu_k, \sigma_k^2)$, where μ_k and σ_k are the unknown, slowly time varying, mean and standard deviation. Inserting (4) into (2), and performing a discretization using sampling period T_s leads to

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \bar{\mathbf{A}}_\phi \mathbf{z}_k + \mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) w_k, \quad (6)$$

where $\mathbf{u}_k = [v^x \ \delta_m]^T$ are the known inputs, and where v^x is estimated from the wheel rotation rates. For the purposes of this paper (6) is linear in state and inputs. However, the method we propose is generally applicable to nonlinear vehicle models.

Accounting for the road-bank angle dynamics, the state vector consists of \mathbf{x} and \mathbf{z} , which are propagated forward in time by (6) and (3), respectively.

2.1.1 Measurement Model

We are also interested in learning the slowly time-varying offsets in the acceleration and gyro measurements, as well as their corresponding variances. The measurement model therefore incorporates the measurements of the lateral acceleration, a_m^y , and the yaw rate $\dot{\psi}_m$. The yaw-rate measurement is directly related to $\dot{\psi}$, whereas the lateral acceleration a^y can be extracted from the right-hand side of (2a), after dividing by the vehicle mass. The roll-rate measurement $\dot{\phi}_m$ picks up effects from both the vehicle roll rate $\dot{\phi}_v$ and the bank rate $\dot{\phi}$, $\dot{\phi}_m = \dot{\phi}_v + \dot{\phi}$. However, we assume that $\dot{\phi}_v \approx 0$ and therefore $\dot{\phi}_m \approx \dot{\phi}$. The inertial measurements are summarized in the state vector

$$\mathbf{y}_k = [a_m^y \ \dot{\psi}_m \ \dot{\phi}_m]^T. \quad (7)$$

We also have measurements of ϕ , ϕ_m , where we similar to the roll rate assume that the roll angle is sufficiently small. Production vehicles are typically not equipped with direct roll angle measurements. However, this information can either be retrieved from the suspension system of the vehicle or from differential GPS [23].

Similar to the steering offset (5), we model the measurement noise \mathbf{e}_k of the inertial sensors as Gaussian with unknown mean \mathbf{b}_k (the IMU bias) and covariance \mathbf{R}_k according to $\mathbf{e}_k \sim \mathcal{N}(\mathbf{b}_k, \mathbf{R}_k)$. Since the primary focus is learning the noise statistics of the inertial sensors, we assume that ϕ_m is zero-mean Gaussian distributed with known standard deviation σ_ϕ , $e_{\phi,k} \sim \mathcal{N}(0, \sigma_\phi)$. Thus, the measurement model can be written as

$$\begin{bmatrix} \mathbf{y}_k \\ \phi_{m,k} \end{bmatrix} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_k, \mathbf{z}_k, \mathbf{u}_k) \\ \mathbf{C}\mathbf{z}_k \end{bmatrix} + \begin{bmatrix} \mathbf{d}(\mathbf{x}_k, \mathbf{z}_k, \mathbf{u}_k) \\ 0 \end{bmatrix} w_k + \begin{bmatrix} \mathbf{e}_k \\ e_{\phi,k} \end{bmatrix}, \quad (8)$$

where $\mathbf{d} = [\frac{C_f^y}{m} \ 0 \ 0]^T$ relates the steering angle to the measurements according to (2) and $\mathbf{C} = [1 \ 0 \ 0]$. Again, for our purposes (8) is linear, but the method we propose is generally applicable to nonlinear systems, which is the case if, for instance, the longitudinal dynamics are also considered, or nonstandard driving conditions are considered (e.g., performance-limit maneuvers).

The joint Gaussian distribution of the steering offset w_k and the inertial sensor measurement noise $\bar{\mathbf{e}}_k$ can be written as $\bar{\mathbf{w}}_k = [w_k^T \ \bar{\mathbf{e}}_k^T]^T \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where we have introduced the short-hand notation $\bar{\mathbf{e}}_k = \mathbf{d}_k w_k + \mathbf{e}_k$ and

$$\boldsymbol{\mu}_k = \begin{bmatrix} \mu_{w,k} \\ \mathbf{d}_k \mu_{w,k} + \mathbf{b}_k \end{bmatrix}, \quad \boldsymbol{\Sigma}_k = \begin{bmatrix} \sigma_k^2 & \sigma_k^2 \mathbf{d}_k^T \\ \mathbf{d}_k \sigma_k^2 & \mathbf{d}_k \sigma_k^2 \mathbf{d}_k^T + \mathbf{R} \end{bmatrix}. \quad (9)$$

Since w_k enters in both the vehicle model (6) and the measurement model (8), the process and measurement noise (9) are dependent. There are two main reasons for assuming Gaussian noise. First, inertial sensors typically have characteristics that fit well to the Gaussian distribution (see e.g., [24]). Furthermore, as we show in Sec. 3, assuming Gaussian noise leads to desired mathematical properties as it enables analytic computations and thus implementation of the method efficiently in real time.

2.1.2 Observability

Observability can be analyzed by augmenting (3), (6) with a random walk model of the steering offset and bias terms, and by computing the observability Gramian. With the current sensor setup, the system is detectable but not fully observable. To remedy this, we utilize that the angular velocities of the rear wheels can be converted to virtual measurements of the yaw rate according to

$$\dot{\Psi}_{\text{virt}} = \frac{\omega_r^{(r)} r - \omega_r^{(l)} r}{l_T}, \quad (10)$$

where l_T is the distance between the rear left and rear right wheel and $\omega_r^{(l)}$, $\omega_r^{(r)}$, are the rotation rates for the rear left and rear right wheel, respectively. With the additional measurement (10), the system composed of (3), (6), (8), and (10) is observable. In this paper we assume that the virtual measurement (10) is Gaussian distributed with zero mean and

a priori determined standard deviation σ_{virt} , and we denote the full measurement vector $\bar{\mathbf{y}}_k = [\mathbf{y}_k^T \ \phi_m \ \dot{\Psi}_{\text{virt}}]^T$. In practice, measurements using the wheel rotation speed have scale errors due to differences between the true and estimated wheel radius r . This is not considered here but we refer to [16] for one way to estimate the tire radii.

Remark 1. We stress that it is common to model the bias vector \mathbf{b}_k as a random walk and extend the state vector, and the covariance \mathbf{R}_k of the measurement noise is typically also determined a priori. However, determining the covariance a priori can be significantly time consuming, and error prone. The same applied to the process noise of the bias random walk. In fact, unmodeled effects can lead to differences between the effective measurement noise and the sensor specifications. Hence, including the bias and variances in the estimation problem formulation provides additional robustness throughout the entire vehicle operating life.

Remark 2. The incorporation of bank-angle estimation (3) and the measurements ϕ_m , $\dot{\phi}_m$ is optional. For instance, if it is expected that the road is mostly flat it may be preferable to avoid incorporating the road-bank effects since it leads to a more complex estimation problem and requires using more sensors.

2.2 Problem Formulation

We want to recursively estimate the steering offset and the noise statistics of the inertial measurements. In a Bayesian setting, this can be expressed as learning the parameters $\boldsymbol{\theta}_k := \{\mu_{w,k}, \mathbf{b}_k, \sigma_k, \mathbf{R}_k\}$ of the Gaussian noise sources w_k , \mathbf{e}_k . We approach this problem in the following way. Given the system model (6)–(10) and dependent Gaussian noise between w_k and $\bar{\mathbf{e}}_k$ characterized by (9), where the unknown parameters $\boldsymbol{\theta}_k$ may be time varying, we recursively estimate

$$p(\boldsymbol{\theta}_k | \bar{\mathbf{y}}_{0:k}), \quad (11a)$$

$$p(\mathbf{x}_k | \bar{\mathbf{y}}_{0:k}). \quad (11b)$$

Eqs. (11a) and (11b) are coupled, which will be apparent in the derivation of the proposed solution, because (11a) depends on the state trajectory and the density (11b) depends on the parameter estimates. Because the lateral velocity and lateral acceleration depend on the bank angle, we also need to estimate the state vector \mathbf{z}_k .

3 Marginalized Particle Filter for Sensor Estimation

This section focuses on determining the densities in (11). We formulate the joint estimation in a Bayesian framework as approximating the joint filtering density $p(\mathbf{z}_k, \boldsymbol{\theta}_k, \mathbf{x}_{0:k} | \bar{\mathbf{y}}_{0:k})$, that is, the joint posterior conditioned on all measurements from time index 0 to k . We decompose

$$p(\mathbf{z}_k, \boldsymbol{\theta}_k, \mathbf{x}_{0:k} | \bar{\mathbf{y}}_{0:k}) = p(\mathbf{z}_k | \boldsymbol{\theta}_k, \mathbf{x}_{0:k} | \bar{\mathbf{y}}_{0:k}) p(\boldsymbol{\theta}_k | \mathbf{x}_{0:k} | \bar{\mathbf{y}}_{0:k}) p(\mathbf{x}_{0:k} | \bar{\mathbf{y}}_{0:k}), \quad (12)$$

and recursively estimate the densities in (12). Note that for the case of planar motion, all terms involving \mathbf{z} can be ignored in the following derivations.

3.1 State Estimation

We approximate the posterior of the state trajectory with a particle filter as

$$p(\mathbf{x}_{0:k}|\bar{\mathbf{y}}_{0:k}) \approx \sum_{i=1}^N q_k^i \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i), \quad (13)$$

where $\delta(\cdot)$ is the Dirac delta mass and q_k^i is the importance weight for the i th state trajectory sample $\mathbf{x}_{0:k}^i$. The approximate distribution (13) is propagated with a sequential importance resampling (SIR) based particle filter [14]. In general, the particles are sampled using a proposal distribution $\pi(\mathbf{x}_{k+1}|\mathbf{x}_{0:k}^i, \bar{\mathbf{y}}_{0:k+1})$, which starts from the particles at the previous time step. For dependent noise, the weight update is performed as [20]

$$q_k^i \propto q_{k-1}^i \frac{p(\bar{\mathbf{y}}_k|\mathbf{x}_{0:k}^i, \bar{\mathbf{y}}_{0:k-1})p(\mathbf{x}_k^i|\mathbf{x}_{0:k-1}^i, \bar{\mathbf{y}}_{0:k-1})}{\pi(\mathbf{x}_k^i|\mathbf{x}_{0:k-1}^i, \bar{\mathbf{y}}_{0:k})}, \quad (14)$$

where $p(\bar{\mathbf{y}}_k|\mathbf{x}_{0:k}^i, \bar{\mathbf{y}}_{0:k-1})$ is the likelihood. If the proposal is chosen equal to $p(\mathbf{x}_k^i|\mathbf{x}_{0:k-1}^i, \bar{\mathbf{y}}_{0:k-1})$, (14) simplifies to

$$q_k^i \propto q_{k-1}^i p(\bar{\mathbf{y}}_k|\mathbf{x}_{0:k}^i, \bar{\mathbf{y}}_{0:k-1}). \quad (15)$$

Hence, to obtain new weights, we need to evaluate

$$p(\bar{\mathbf{y}}_k|\mathbf{x}_{0:k}^i, \bar{\mathbf{y}}_{0:k-1}), \quad (16a)$$

$$p(\mathbf{x}_{k+1}^i|\mathbf{x}_{0:k}^i, \bar{\mathbf{y}}_{0:k}). \quad (16b)$$

3.2 Parameter Estimation

According to (6) and (8), knowing both the state and measurement trajectory leads to full knowledge about $\bar{\mathbf{w}}_{0:k} = [w_{0:k} \ \bar{\mathbf{e}}_{0:k}]^T$. The posterior for the noise parameters can therefore be rewritten using Bayes' rule as

$$p(\boldsymbol{\theta}_k|\mathbf{x}_{0:k}, \mathbf{y}_{0:k}) = p(\boldsymbol{\theta}_k|\bar{\mathbf{w}}_{0:k}) \propto p(\bar{\mathbf{w}}_k|\boldsymbol{\theta}_k)p(\boldsymbol{\theta}_k|\bar{\mathbf{w}}_{0:k-1}). \quad (17)$$

Based on the assumption the conditional probability distribution of the noise with respect to the noise parameters, $p(\bar{\mathbf{w}}_k|\boldsymbol{\theta}_k)$ in (17), is Gaussian, we can utilize the concept of conjugate priors. If a prior distribution belongs to the same family as the posterior distribution, the prior is said to be conjugate to the particular likelihood. For multivariate Normal data $\bar{\mathbf{w}} \in \mathbb{R}^d$ with unknown mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, a Normal-inverse-Wishart distribution defines the conjugate prior [25], $p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) := \text{NiW}(\gamma_{k|k}, \hat{\boldsymbol{\mu}}_{k|k}, \boldsymbol{\Lambda}_{k|k}, \mathbf{v}_{k|k})$,

through the model

$$\begin{aligned} \boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k &\sim \mathcal{N}(\hat{\boldsymbol{\mu}}_{k|k}, \boldsymbol{\Sigma}_k), \\ \boldsymbol{\Sigma}_k &\sim \text{iW}(\mathbf{v}_{k|k}, \boldsymbol{\Lambda}_{k|k}) \\ &\propto |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}(\mathbf{v}_{k|k}+d+1)} e^{(-\frac{1}{2}\text{tr}(\boldsymbol{\Lambda}_{k|k}\boldsymbol{\Sigma}_k^{-1}))}, \end{aligned}$$

where $\text{tr}(\cdot)$ is the trace operator. We compute the statistics $S_{k|k} := (\gamma_{k|k}, \hat{\boldsymbol{\mu}}_{k|k}, \boldsymbol{\Lambda}_{k|k}, \mathbf{v}_{k|k})$ for each particle as (see [26])

$$\gamma_{k|k} = \frac{\gamma_{k|k-1}}{1 + \gamma_{k|k-1}}, \quad (18a)$$

$$\hat{\boldsymbol{\mu}}_{k|k} = \hat{\boldsymbol{\mu}}_{k|k-1} + \gamma_{k|k} \boldsymbol{\eta}_k, \quad (18b)$$

$$\mathbf{v}_{k|k} = \mathbf{v}_{k|k-1} + 1, \quad (18c)$$

$$\boldsymbol{\Lambda}_{k|k} = \boldsymbol{\Lambda}_{k|k-1} + \frac{1}{1 + \gamma_{k|k-1}} \boldsymbol{\eta}_k \boldsymbol{\eta}_k^T, \quad (18d)$$

$$\boldsymbol{\eta}_k = \bar{\mathbf{w}}_k - \hat{\boldsymbol{\mu}}_{k|k-1}, \quad (18e)$$

where the data $\bar{\mathbf{w}}_k$ for each particle is generated by

$$\bar{\mathbf{w}}_k^i = \begin{bmatrix} w_k^i \\ \mathbf{e}_k^i \end{bmatrix} = \begin{bmatrix} \mathbf{g}_k^{i,-\dagger} (\mathbf{x}_{k+1}^i - \mathbf{f}_k^i - \bar{\mathbf{A}}_\phi \mathbf{z}_{k|k}^i) \\ \mathbf{y}_k - \mathbf{h}_k^i - \mathbf{d}_k^i \boldsymbol{\mu}_{w,k}^i \end{bmatrix}, \quad (19)$$

where $\mathbf{g}_k^{i,-\dagger}$ is the pseudo-inverse of \mathbf{g}_k^i . Hence, a key task in this paper is how to generate the particles in (19) to update the parameters. For slowly time-varying parameters, the prediction step consists of

$$\gamma_{k|k-1} = \frac{1}{\lambda} \gamma_{k-1|k-1}, \quad (20a)$$

$$\hat{\boldsymbol{\mu}}_{k|k-1} = \hat{\boldsymbol{\mu}}_{k-1|k-1}, \quad (20b)$$

$$\mathbf{v}_{k|k-1} = \lambda \mathbf{v}_{k-1|k-1}, \quad (20c)$$

$$\boldsymbol{\Lambda}_{k|k-1} = \lambda \boldsymbol{\Lambda}_{k-1|k-1}, \quad (20d)$$

where $\lambda \in (0, 1]$ introduces exponential forgetting. Since we know the dependence structure (9), the scale matrix $\boldsymbol{\Lambda}_k$ can be decomposed as

$$\boldsymbol{\Lambda}_k = \begin{bmatrix} \Lambda_{w,k} & \Lambda_{w,k} \mathbf{d}_k^T \\ \mathbf{d}_k \Lambda_{w,k} & \mathbf{d}_k \Lambda_{w,k} \mathbf{d}_k^T + \boldsymbol{\Lambda}_{\mathbf{e},k} \end{bmatrix}, \quad (21)$$

implying that it suffices to propagate $\Lambda_{w,k}$ and $\boldsymbol{\Lambda}_{\mathbf{e}}$ in (18d) and (20d). Further, for a Normal-inverse-Wishart prior, the predictive distribution of the data $\bar{\mathbf{w}}$ is a Student-t, $\text{St}(\hat{\boldsymbol{\mu}}_{k|k-1}, \tilde{\boldsymbol{\Lambda}}_{k|k-1}, \mathbf{v}_{k|k-1} - d + 1)$, with

$$\tilde{\boldsymbol{\Lambda}}_{k|k-1} = \frac{1 + \gamma_{k|k-1}}{\mathbf{v}_{k|k-1} - d + 1} \boldsymbol{\Lambda}_{k|k-1}.$$

If the predictive distribution $p(\boldsymbol{\theta}_k|\bar{\mathbf{w}}_{0:k-1})$ in (17) is a Normal-inverse-Wishart distribution, from (17), (18), also

the posterior is Normal-inverse Wishart, $p(\boldsymbol{\theta}_k | \mathbf{x}_{0:k}, \mathbf{y}_{0:k}) = \text{NiW}(\hat{\boldsymbol{\mu}}_{k|k}, \hat{\boldsymbol{\Lambda}}_{k|k}, \mathbf{v}_{k|k})$. To obtain estimates of the mean and covariance of the noise processes, we rewrite the marginal (11a) as

$$p(\boldsymbol{\theta}_k | \bar{\mathbf{y}}_{0:k}) = \int p(\boldsymbol{\theta}_k | \mathbf{x}_{0:k}, \mathbf{y}_{0:k}) p(\mathbf{x}_{0:k} | \bar{\mathbf{y}}_{0:k}) d\mathbf{x}_{0:k} \\ \approx \sum_{i=1}^N q_k^i p(\boldsymbol{\theta}_k | \mathbf{x}_{0:k}^i, \bar{\mathbf{y}}_{0:k}), \quad (22)$$

which has complexity $O(N)$. Based on (22), the unknown parameters can be extracted; for example, the estimate of \mathbf{b}_k and \mathbf{R}_k can be found as

$$\hat{\mathbf{b}}_k = \sum_{i=1}^N q_k^i \hat{\mathbf{b}}_{k|k}^i, \quad (23a)$$

$$\hat{\mathbf{R}}_k = \sum_{i=1}^N q_k^i \left(\frac{1}{\tilde{\mathbf{v}}_{k|k}} \boldsymbol{\Lambda}_{\mathbf{e},k|k}^i + (\hat{\mathbf{b}}_{k|k}^i - \hat{\mathbf{b}}_k)(\hat{\mathbf{b}}_{k|k}^i - \hat{\mathbf{b}}_k)^T \right), \quad (23b)$$

and similarly for $\hat{\boldsymbol{\mu}}_{w,k}$, $\hat{\boldsymbol{\sigma}}_k$, where $\tilde{\mathbf{v}}_{k|k} = \mathbf{v}_{k|k} - d - 1$.

3.3 Noise Marginalization

Consider first the likelihood (16a) resulting in the weight update (15), and note that the noise processes of the inertial sensors and the steering-wheel angle are independent of ϕ_m and ψ_{virt} . Hence, from the state-space model (3), (6), and (8), the knowledge of $\mathbf{x}_{0:k}$, $\phi_{0:k}$, and $\mathbf{y}_{0:k}$ gives full knowledge of the unknown noise sequence $\bar{\mathbf{e}}_{0:k}$. The property of transformations of variables in densities [27] gives that

$$p(\mathbf{y}_k | \mathbf{x}_{0:k}, \mathbf{y}_{0:k-1}) \propto p(\bar{\mathbf{e}}_k(\mathbf{y}_k, \mathbf{x}_k) | \bar{\mathbf{e}}_{0:k-1}). \quad (24)$$

We marginalize out the noise parameters and \mathbf{z} as

$$p(\mathbf{y}_k | \mathbf{x}_{0:k}, \mathbf{y}_{0:k-1}) = \int p(\mathbf{y}_k | \boldsymbol{\theta}_k, \mathbf{z}_k, \mathbf{x}_k) \\ \cdot p(\mathbf{z}_k | \mathbf{y}_{0:k-1}, \boldsymbol{\theta}_k, \mathbf{x}_{0:k}) p(\boldsymbol{\theta}_k | \mathbf{x}_{0:k}, \mathbf{y}_{0:k-1}) d\mathbf{z}_k d\boldsymbol{\theta}_k. \quad (25)$$

The first two integrands in (25) are Gaussian and the third integrand is Normal-inverse-Wishart. Hence, (25) is Student-t distributed [25], implying that

$$p(\bar{\mathbf{e}}_k(\mathbf{y}_k, \mathbf{x}_k) | \bar{\mathbf{e}}_{0:k-1}) = \text{St}(\hat{\boldsymbol{\mu}}_{\bar{\mathbf{e}},k|k-1}, \tilde{\boldsymbol{\Lambda}}_{\bar{\mathbf{e}},k|k-1}, \tilde{\mathbf{v}}_{k|k-1}),$$

with $\tilde{\mathbf{v}}_{k|k-1} = \mathbf{v}_{k|k-1} - d + 1$, and mean and scaling

$$\hat{\boldsymbol{\mu}}_{\bar{\mathbf{e}},k|k-1} = \mathbf{d}_k \hat{\boldsymbol{\mu}}_{w,k|k-1} + \hat{\mathbf{b}}_{k|k-1}, \\ \tilde{\boldsymbol{\Lambda}}_{\bar{\mathbf{e}},k|k-1} = \frac{1 + \gamma_{k|k-1}}{\tilde{\mathbf{v}}_{k|k-1}} \left(\mathbf{d}_k \boldsymbol{\Lambda}_{w,k|k-1} \mathbf{d}_k^T + \boldsymbol{\Lambda}_{\mathbf{e},k|k-1} \right).$$

The full measurement noise also contains the scalar components e_{ψ} and e_{ϕ} due to the virtual measurement (10) and road bank measurement ϕ_m , which are both zero-mean Gaussian with known variance. However, this gives a joint density which has no closed form expression. An approach to resolve this is to resort to moment matching; that is, we model the full measurement noise as a Student-t distribution with a common degree of freedom,

$$\begin{bmatrix} \bar{\mathbf{e}}_k \\ e_{\phi} \\ e_{\psi} \end{bmatrix} \sim \text{St} \left(\begin{bmatrix} \hat{\boldsymbol{\mu}}_{\bar{\mathbf{e}},k|k-1} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \tilde{\boldsymbol{\Lambda}}_{\bar{\mathbf{e}},k|k-1} & \mathbf{0} \\ \mathbf{0}^T & \Lambda_{\text{aug}} \end{bmatrix}, \tilde{\mathbf{v}}_{k|k-1} \right), \quad (26)$$

where

$$\Lambda_{\text{aug}} = \frac{\tilde{\mathbf{v}}_{k|k-1} - 2}{\tilde{\mathbf{v}}_{k|k-1}} \begin{bmatrix} \sigma_{\phi}^2 & 0 \\ 0 & \sigma_{\text{virt}}^2 \end{bmatrix}.$$

The Student-t converges to the Gaussian as the degrees of freedom tend to infinity. Hence, from $\lim_{\nu \rightarrow \infty} \text{St}(\boldsymbol{\mu}, \boldsymbol{\Lambda}, \nu) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ and the update formulas (18c) and (20c), it follows that we recover the Gaussian measurement noise of the virtual measurement with precision determined by the forgetting factor [12]. Hence, the measurement update (15) is

$$q_k^i \propto q_{k-1}^i \text{St}(\boldsymbol{\mu}^*, \tilde{\boldsymbol{\Lambda}}^*, \tilde{\mathbf{v}}), \quad (27)$$

which can be evaluated analytically, where

$$\boldsymbol{\mu}^* = \mathbf{h}_k + \mathbf{d}_k \hat{\boldsymbol{\mu}}_{w,k|k-1} + \mathbf{b}_k, \\ \tilde{\boldsymbol{\Lambda}}^* = \frac{1 + \gamma_{k|k-1}}{\tilde{\mathbf{v}}_{k|k-1}} \left(\mathbf{d}_k \boldsymbol{\Lambda}_{w,k|k-1} \mathbf{d}_k^T + \boldsymbol{\Lambda}_{\mathbf{e},k|k-1} \right) + \Lambda_{\text{aug}}.$$

The prediction step (16b) is resolved in a similar way,

$$p(\mathbf{x}_{k+1} | \mathbf{x}_{0:k}, \bar{\mathbf{y}}_{0:k}) \propto p(\mathbf{g}_k^{-\dagger}(\mathbf{x}_{k+1} - \mathbf{f}_k - \mathbf{A}_{\phi} \mathbf{z}_k) | \mathbf{x}_{0:k}, \bar{\mathbf{y}}_{0:k}) \\ = p(\mathbf{g}_k^{-\dagger}(\mathbf{x}_{k+1} - \mathbf{f}_k - \mathbf{A}_{\phi} \mathbf{z}_k) | \bar{\mathbf{e}}_{0:k}) \\ = p(w_k(\mathbf{x}_{k+1}) | \bar{\mathbf{e}}_{0:k}). \quad (28)$$

We integrate out $\boldsymbol{\theta}_k$, \mathbf{z}_k in (16b),

$$p(\mathbf{x}_{k+1} | \mathbf{x}_{0:k}, \bar{\mathbf{y}}_{0:k}) = \int p(\mathbf{x}_{k+1} | \boldsymbol{\theta}_k, \mathbf{z}_k, \mathbf{x}_{0:k}, \bar{\mathbf{y}}_{0:k}) \\ \cdot p(\mathbf{z}_k | \boldsymbol{\theta}_k, \mathbf{x}_{0:k}, \bar{\mathbf{y}}_{0:k}) p(\boldsymbol{\theta}_k | \mathbf{x}_{0:k}, \bar{\mathbf{y}}_{0:k}) d\mathbf{z}_k d\boldsymbol{\theta}_k. \quad (29)$$

The integrand in (29) is a product of Gaussian and Normal-inverse-Wishart distributions, resulting in a Student-t distribution after integration [25]. Combining with (28), we obtain a Student-t distribution for w_k as

$$p(w_k(\mathbf{x}_{k+1}) | \bar{\mathbf{e}}_{0:k}) = \text{St}(\hat{\boldsymbol{\mu}}_k^*, \tilde{\boldsymbol{\Lambda}}_k^*, \mathbf{v}_k^*), \quad (30)$$

where

$$\begin{aligned} \mathbf{v}_k^* &= \mathbf{v}_{k|k-1} - d + 1 + d_y, \\ \boldsymbol{\mu}_k^* &= \hat{\boldsymbol{\mu}}_{w,k|k-1} + \mathbf{d}_k \boldsymbol{\Lambda}_{w,k|k-1} \tilde{\boldsymbol{\Lambda}}_{\bar{\mathbf{e}},k|k-1}^{-1} \boldsymbol{\eta}_k, \\ \tilde{\boldsymbol{\Lambda}}_k^* &= \frac{\mathbf{v}_{k|k-1} - d + 1 + \boldsymbol{\eta}_k \tilde{\boldsymbol{\Lambda}}_{\bar{\mathbf{e}},k|k-1}^{-1} \boldsymbol{\eta}_k^T}{\mathbf{v}_{k|k-1} - d_y + 1} \left(\boldsymbol{\Lambda}_{w,k|k-1} \right. \\ &\quad \left. - \mathbf{d}_k \boldsymbol{\Lambda}_{w,k|k-1} \tilde{\boldsymbol{\Lambda}}_{\bar{\mathbf{e}},k|k-1}^{-1} \boldsymbol{\Lambda}_{w,k|k-1}^T \mathbf{d}_k^T \right), \\ \boldsymbol{\eta}_k &= \bar{\mathbf{e}}_k - \hat{\boldsymbol{\mu}}_{\bar{\mathbf{e}},k|k-1}. \end{aligned}$$

3.4 Road-Bank Estimation

The main purpose of the road-bank estimation is to remove errors in the noise-statistics estimation due to effects of the road bank on the vehicle dynamics. Consequently, we use a process noise w_ϕ in (3) and measurement noise e_ϕ of the angle measurement ϕ_m with constant parameters, and we propose an add-on scheme to the existing estimator for estimating the road bank.

The motion model (3) is linear and the measurement model (8) is linear in \mathbf{z}_k given \mathbf{x}_k . Furthermore, given \mathbf{x}_{k+1} and \mathbf{x}_k and by moving over \mathbf{f}_k to the left-hand side, the vehicle model (6) is linear and Gaussian. Hence, we can approximate $p(\mathbf{z}_k | \boldsymbol{\theta}_k, \mathbf{x}_{0:k}, \bar{\mathbf{y}}_k)$ in (12) by an RBPF [18], where the vehicle model (6) acts as an extra measurement of \mathbf{z}_k , and where the variance of the measurement noise is given by $\boldsymbol{\Sigma}_k^i = \boldsymbol{\Lambda}_k^i / (\mathbf{v}_k - d_y - 1)$.

3.5 Algorithm Summary

In the implementation, the process noise is generated from (30) and used in (6) to generate samples $\{\mathbf{x}_{k+1}^i\}_{i=1}^N$. This removes the need for the matrix inversion in (19), since w_k is directly generated from (30) instead of generated from the samples \mathbf{x}_{k+1} . Algorithm 1 summarizes the method.

4 Simulation Study

This section evaluates the proposed method in simulation using measured input data (wheel rotation rates and steer angle) from an experimental test drive. We then generate synthetic ground-truth data by feeding the single-track model (2) with the measured input data. The true variances of the sensors are fixed and the sensor bias is generated from a random-walk model, which is motivated by the low-frequency dynamics of a typical inertial sensor unit [24] after averaging out the initial offset. Note that the generation of the synthetic bias does not use the same model as the estimation algorithm, where the bias vector is modeled as the mean of a Gaussian disturbance, hence allowing some assessment of the robustness. The steering-wheel offset is fixed to a constant in the simulations, which cannot be averaged out, and the road bank rate is modeled as piecewise linear.

4.1 Simulation Results

The statistics of the Normal-inverse-Wishart is initialized as zero-mean with the estimated standard deviations set

Algorithm 1 Pseudo-code of the estimation algorithm

Initialize: Set $\{\mathbf{x}_0^i\}_{i=1}^N \sim p_0(\mathbf{x}_0)$, $\{q_0^i\}_{i=1}^N = 1/N$, $\{S_0^i\}_{i=1}^N = \{\gamma_0, \bar{\boldsymbol{\mu}}_0, \boldsymbol{\Lambda}_{w,0}, \mathbf{v}_0\}$, $\{\hat{\mathbf{z}}_{0|-1}^i, \mathbf{P}_{0|-1}^i\}_{i=1}^N = \{\mathbf{z}_0, \mathbf{P}_0\}$

- 1: **for** $k \leftarrow 0$ to T **do**
- 2: **for** $i \in \{1, \dots, N\}$ **do**
- 3: Update weight \bar{q}_k^i using (27).
- 4: Update noise statistics $S_{k|k}^i$ using (18).
- 5: **end for**
- 6: Normalize weights as $q_k^i = \bar{q}_k^i / (\sum_{i=1}^N \bar{q}_k^i)$.
- 7: Compute $N_{\text{eff}} = 1 / (\sum_{i=1}^N (q_k^i)^2)$
- 8: **if** $N_{\text{eff}} \leq N_{\text{thr}}$ **then**
- 9: Resample particles and copy the corresponding statistics. Set $\{q_k^i\}_{i=1}^N = 1/N$.
- 10: **end if**
- 11: **for** $i \in \{1, \dots, N\}$ **do**
- 12: Determine $\{\hat{\mathbf{z}}_{k|k}^i, \mathbf{P}_{k|k}^i\}_{i=1}^N$ (see (10) in [18]).
- 13: **end for**
- 14: Compute state estimates $\mathbf{x}_k = \sum_{i=1}^N q_k^i \mathbf{x}_k^i$.
- 15: Compute estimates of noise parameters using (23).
- 16: **for** $i \in \{1, \dots, N\}$ **do**
- 17: Predict noise statistics $S_{k+1|k}^i$ using (20).
- 18: Sample \mathbf{w}_k^i from (30).
- 19: Predict state \mathbf{x}_{k+1}^i using (6).
- 20: Predict $\{\hat{\mathbf{z}}_{k+1|k}^i, \mathbf{P}_{k+1|k}^i\}$ (see (17) in [18]).
- 21: **end for**
- 22: **end for**

to be roughly twice of the true standard deviation, and the forgetting factor is set to $\lambda = 0.995$.

Fig. 3 shows the estimated standard deviations for the lateral acceleration, yaw rate, and roll rate in red and the true values in black. After the initial transients, the gyro variances converge to their true values. The standard deviation estimate for the lateral acceleration sensor exhibits a longer transient, partially due to the more complicated measurement function, which also involves estimation of mean and variance of the steering offset. Setting the forgetting factor to a lower value or the initial variances to larger values leads to faster convergence (at the cost of larger fluctuations in steady state). The estimated offsets are shown in Fig. 4. There is a cross-dependence between the steering offset and the measurement offsets, especially the acceleration measurement. However, the estimator is able to track the low-frequency components of the offsets closely. Note that although Fig. 3 shows results for constant noise variances, the method handles slowly time-varying variances similar to the variations in the bias components in Fig. 4.

Fig. 5 displays the estimated steering-wheel offset over 100 Monte-Carlo executions. The maximum and minimum deviations from the true value in steady state are roughly ± 0.04 deg.

The road-bank angle and rate, respectively, are shown in Fig. 6, and Fig. 7 displays the lateral velocity and yaw rate. Also these quantities are tracked closely, which shows that the proposed approach is capable of estimating all quantities

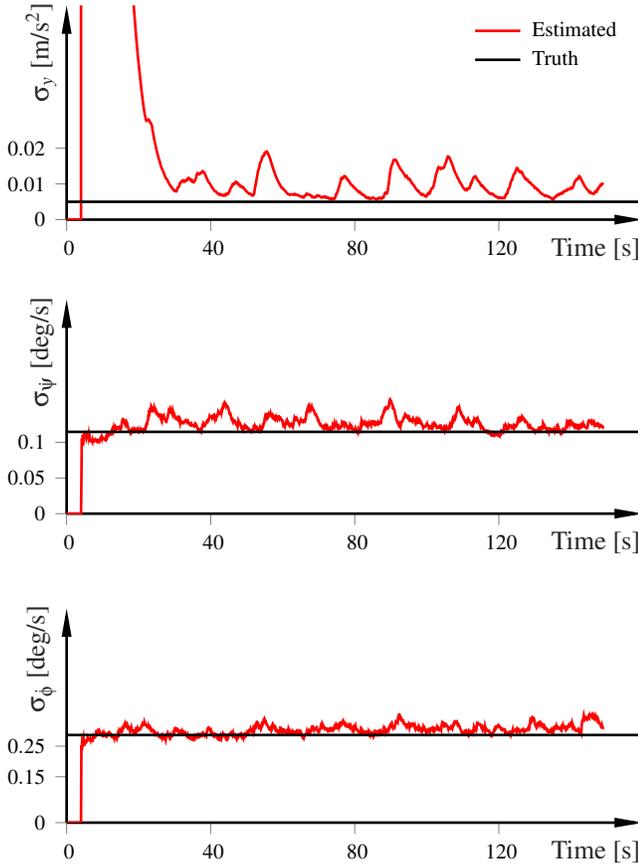


Fig. 3. Estimated standard deviations (red) and true values (black) of the lateral acceleration, yaw rate, and roll rate in simulation for $N = 100$ particles.

of interest and also correct for the bias in the sensors, in simulation.

5 Experimental study

For experimental validation, we have used a mid-size SUV, equipped with industry-grade validation equipment to gather data, and collected several different data sets using the same vehicle setup. The parameters of the vehicle model have been extracted from data sheets and bench testing. The true steering offset has been determined from an offline optimization procedure using a particle Monte-Carlo-Markov-Chain (PMCMC) method described in [28], where it was used for calibration of tire-friction parameters. After a certain number of MCMC iterations (the burn-in period), the method converges to produce consistent estimates [29]. Note that we use the standard internal IMU, steering angle, and wheel rotation rate measurements, obtained from the CAN bus, and only compare with the nonproduction sensors for validation purposes. However, the roll angle measurements have been extracted from an RT3100 mounted on the SUV, since we do not have access to a differential GPS or to the suspension deflection sensors. The tuning parameters are the same for all results in this section. We use a forgetting factor $\lambda = 0.999$ and activate the algorithm whenever all wheel rotation rates are nonzero.

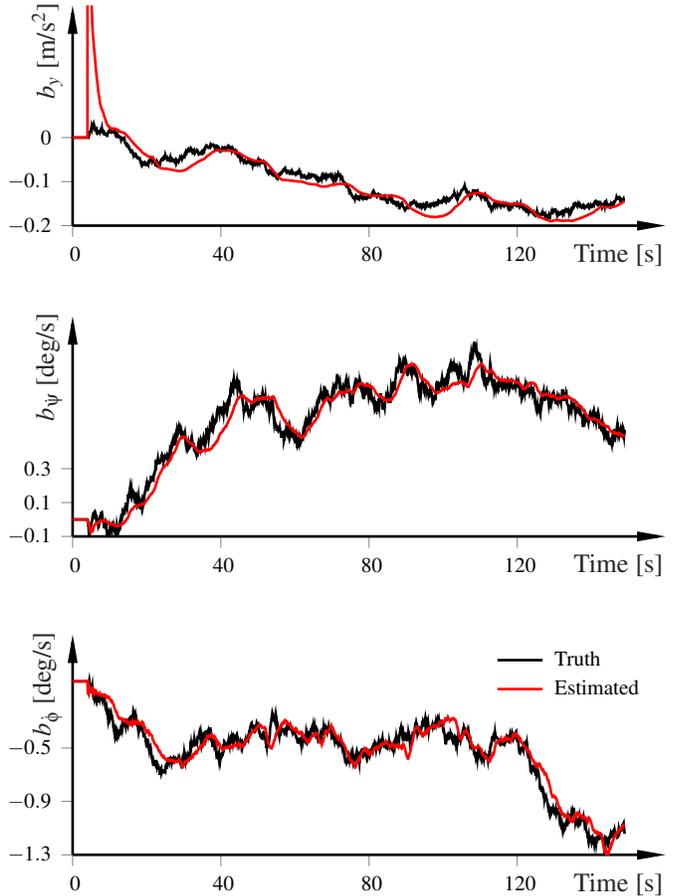


Fig. 4. The estimated bias (red) and true bias (black) in simulation for $N = 100$ particles.

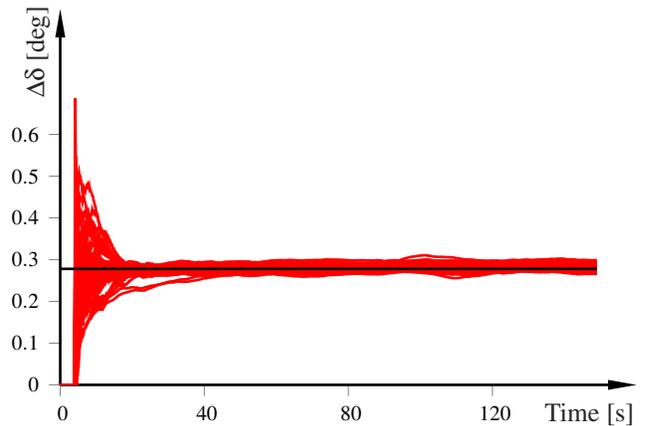


Fig. 5. Estimated steering offset (red) and true offset (black) in simulation for $N = 100$ particles. The results from 100 Monte-Carlo trials are displayed, and the offset is 0.28 deg at the road side in all simulations.

In the validation, each of the three data sets are about three minutes long. The data has been recorded from test drives on multiple loops of a track, with portions of significant (≈ 6 deg) road-bank angle on parts of the track. Due to the proximity in time between the test drives and the (very) slowly time-varying nature of the steering offset, the offset values are similar in all data sets.

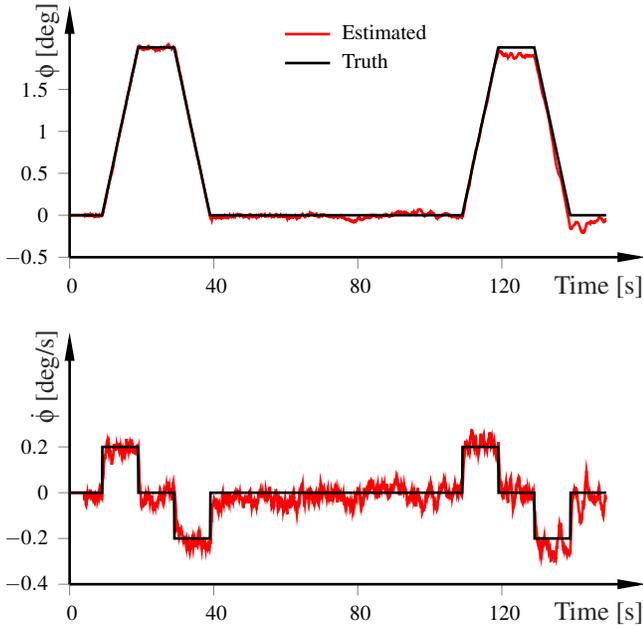


Fig. 6. The estimated road bank and rate (red) and true values (black) in simulation for $N = 100$ particles.

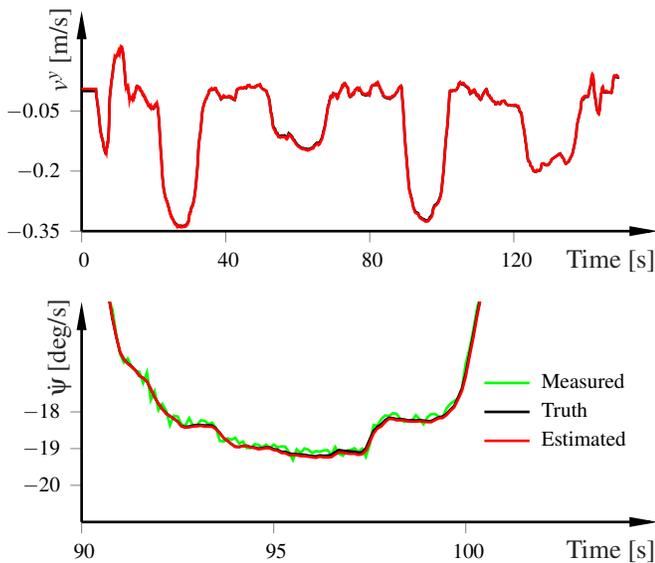


Fig. 7. The estimated lateral velocity and the yaw rate (red) and true values (black) in simulation for $N = 100$ particles. Both states are estimated accurately.

In the simulation study, we assumed that the bias offsets in the IMU were zero initially due to averaging of the initial errors. However, in the experimental evaluation we activate the method as soon as the longitudinal velocity $v^x > 0$, which occurs immediately after start recording the data, and the biases of the IMU are nonzero initially.

5.1 Results

The first part of the experimental evaluation focuses on Monte-Carlo evaluation on a single data set, which is followed by an evaluation for different data sets, to ensure repeatability of the method over different driving behaviors.

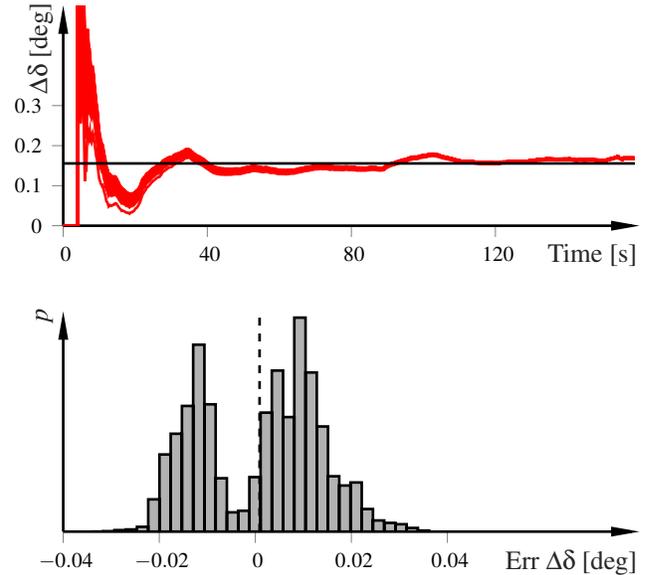


Fig. 8. The upper plot shows the estimated steering-wheel offset (red) for 100 Monte-Carlo executions and the ground truth (black), as obtained by an offline optimization-based procedure, in experiments for $N = 500$ particles. The lower plot displays a corresponding histogram of the error distribution accumulated over all time steps after the initial transients (25 s), for all 100 Monte-Carlo runs. The average error is displayed in dashed vertical line.

5.1.1 Monte-Carlo Evaluation

Fig. 8 displays the estimated steering-wheel angle offset (red) for 100 Monte-Carlo executions on one of the data sets. The ground truth as estimated by the PMCMC method is shown in black. After the initial transients (≈ 40 s), the estimate converge very close to the true mean. The average of the estimates is correct, with maximum deviations around 0.35 deg. Measured at the road side, this corresponds to an accuracy of roughly 0.02 deg road-wheel angle.

In Fig. 9 we show the estimated bias and standard deviation over 100 Monte-Carlo executions (two upper plots) and an excerpt of the measured and true yaw rate, respectively, together with the estimated yaw rate, for one realization. The initial values of the bias samples are set to zero. We do not have ground truth for the bias, as is a time-varying process. However, by comparing the measured yaw rate with the yaw rate from the validation equipment (a measurement system composed of a high-cost, high-precision fiber-optic gyro, and a GPS with real-time kinematic correction), we can get some information about the instantaneous errors between them. The lower plot shows the results after the initial transients of the bias parameters. The estimated yaw rate follows the validation sensor closely. Furthermore, the estimator finds similar bias values (upper plot) and standard deviations (middle plot) across the executions, indicating a high degree of repeatability in the method.

The experimental results for the quantities associated with the bank angle are shown in Fig. 10. The bank angle estimates follow the measurements closely, and the bias estimates are consistent over all 100 Monte-Carlo executions. The error distribution (lower plot) over the entire data set for

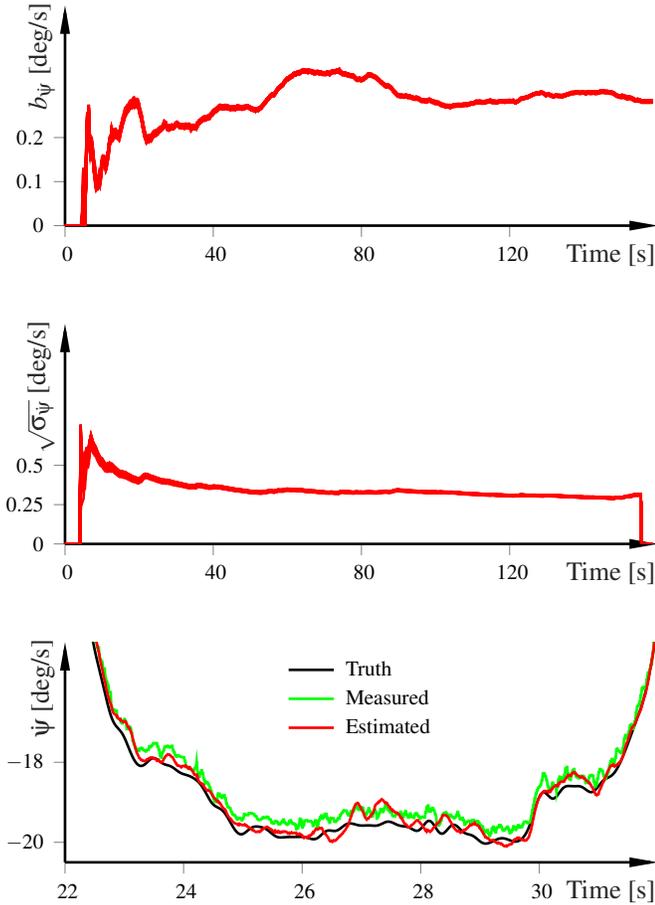


Fig. 9. The estimated bias (upper) standard deviation (mid) for 100 Monte-Carlo runs, and estimated yaw rate (red) for $N = 500$ particles in experiments. The measured yaw rate is in green and the true yaw rate as measured by the validation equipment is in black.

100 Monte-Carlo executions shows that the bank angle estimates are unbiased and that the error rarely exceeds 0.3 deg.

5.1.2 Results for Different Data Sets

Fig. 11 displays the estimated steering-wheel angle offset for four different data sets. After the initial transients, the estimated bias converges to values very close to the true offset for all data sets. This indicates that the method is reliable for different types of driver behaviors.

In Fig. 12 we show the measured yaw rate for the four different data sets (upper plot) and the deviation between the measured yaw rate and the estimated yaw rate with the estimated bias added. The error is for most parts smaller than 0.5 deg/s, which is in the range of the standard deviation of the yaw-rate sensor (see Fig. 9). The spikes that occur last for short periods of time and coincide with rapid changes in the yaw rate.

We conclude the evaluation of our proposed method with an assessment of the computational load shown in Fig. 13, where we plot the average computation time for one estimation iteration of Algorithm 1 (i.e., Lines 2–21) for a varying number of particles. The computer is a standard lap-

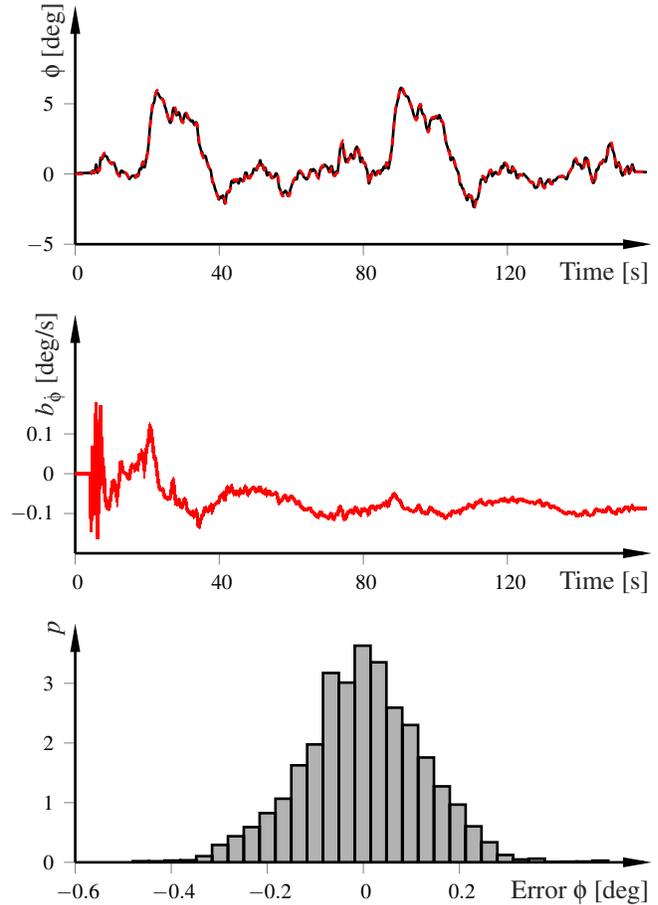


Fig. 10. Estimated bank angle for one realization (upper), estimated bias for 100 Monte-Carlo runs (mid), and the error distribution of the bank angle estimation over the 100 Monte-Carlo runs (c.f. Fig. 8) in experiments.

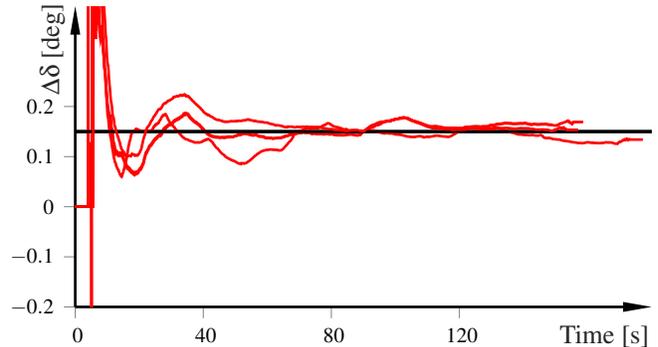


Fig. 11. The estimated steering-wheel offset (red) for four different data sets and the ground truth (black), as obtained by an offline optimization-based procedure, in experiments.

top equipped with a 2014 i5 2.8 GHz processor. We have implemented the algorithm as C-coded mex functions in MATLAB and measured the computation time with the built-in tic-toc functionality. Thus, what we report is an overestimate of the algorithm execution time, due to the overhead introduced by the context switch and the transfer of the variables from MATLAB to C, and due to the overhead introduced by tic-toc. The $O(N)$ line is also shown to verify

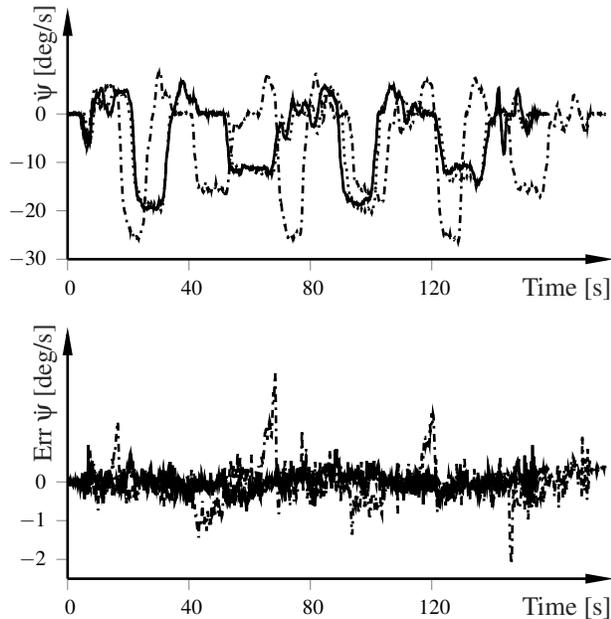


Fig. 12. The measured yaw rate (upper plot) and the error between the measured yaw rate and estimated yaw rate with bias compensation, in experiments for $N = 500$ particles.

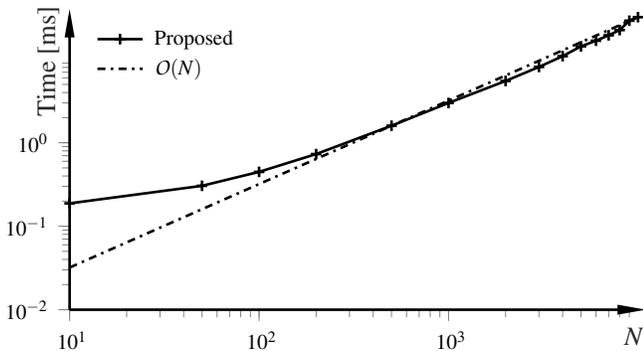


Fig. 13. Average computation time for one iteration of Algorithm 1 for varying number N of particles. The computation time is measured in MATLAB on a 2014 i5 2.8 GHz processor.

that the algorithm is linear in the number of particles. The C-implementation is not optimized for speed nor for the specific processor, and further performance improvements can be obtained. As shown in Fig. 13, the filter with $N = 500$ particles can be executed in the laptop in about 1.4 ms. Thus, a target implementation with sampling period 10 ms can operate in an ECU that is up to 7 times slower than our laptop CPU, and using a sampling period 25 ms leads to that the filter can operate in an ECU that is up to 18 times slower than our laptop CPU. The proposed estimator gives reliable estimates as long as $N \gtrsim 100$, for which the ECU can be up to 25 and 55 times slower, respectively. Note that the fast implementation is feasible due to the marginalization of the noise parameters and the bank dynamics, which reduces the state dimension in the particle filter and hence reduced computations.

6 Conclusion

We developed a method for learning of the offset and variance of low-cost automotive-grade sensors. The offset and noise of the different sensors are related through the vehicle state trajectory and the associated estimation problem is non-Gaussian. We provided a method based on marginalized particle filtering to solve the problem. Tests on simulation and experimental data sets verified that the method can accurately estimate both the offsets and sensor variances, and that the results are repeatable over different data sets with different driving behavior.

References

- [1] Gustafsson, F., 2009. “Automotive safety systems”. *IEEE Signal Process. Mag.*, **26**(4), July, pp. 32–47.
- [2] Reif, K., and Dietsche, K.-H., 2011. *Bosch Automotive Handbook*. Bosch Invented for life. Robert Bosch GmbH, Plochingen, Germany.
- [3] Johansen, T. A., Petersen, I., Kalkkuhl, J., and Ludemann, J., 2003. “Gain-scheduled wheel slip control in automotive brake systems”. *IEEE Trans. Control Syst. Technol.*, **11**(6), pp. 799–811.
- [4] Paden, B., Cap, M., Yong, S. Z., Yershov, D., and Frazzoli, E., 2016. “A survey of motion planning and control techniques for self-driving urban vehicles”. *IEEE Trans. Intell. Veh.*, **1**(1), pp. 33–55.
- [5] Berntorp, K., 2017. “Path planning and integrated collision avoidance for autonomous vehicles”. In *Amer. Control Conf.*
- [6] Thrun, S., Montemerlo, M., et al., 2006. “Stanley: The robot that won the DARPA grand challenge”. *J. Field R.*, **23**(9), pp. 661–692.
- [7] Nilsson, J., Falcone, P., Ali, M., and Sjöberg, J., 2015. “Receding horizon maneuver generation for automated highway driving”. *Control Eng. Pract.*, **41**, pp. 124–133.
- [8] Solyom, S., and Hultén, J., 2013. *Physical Model-Based Yaw Rate and Steering Wheel Angle Offset Compensation*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 51–58.
- [9] Gustafsson, F., Ahlqvist, S., Forssell, U., and Persson, N., 2001. Sensor fusion for accurate computation of yaw rate and absolute velocity. Tech. rep., SAE Technical Paper.
- [10] Berntorp, K., 2016. “Joint wheel-slip and vehicle-motion estimation based on inertial, GPS, and wheel-speed sensors”. *IEEE Trans. Control Syst. Technol.*, **24**(3), pp. 1020–1027.
- [11] Baffet, G., Charara, A., and Lechner, D., 2009. “Estimation of vehicle sideslip, tire force and wheel cornering stiffness”. *Control Eng. Pract.*, **17**(11), pp. 1255–1264.
- [12] Berntorp, K., and Di Cairano, S., 2018. “Tire-stiffness and vehicle-state estimation based on noise-adaptive particle filtering”. *IEEE Trans. Control Syst. Technol.* In press.
- [13] Berntorp, K., and Di Cairano, S., 2018. “Offset

- and noise estimation of automotive-grade sensors using adaptive particle filtering”. In Amer. Control Conf.
- [14] Doucet, A., and Johansen, A. M., 2009. “A tutorial on particle filtering and smoothing: Fifteen years later”. In *Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovsky, eds. Oxford University Press.
- [15] Eidehall, A., Schön, T. B., and Gustafsson, F., 2005. “The marginalized particle filter for automotive tracking applications”. In *Intell. Veh. Symp.*
- [16] Lundquist, C., Karlsson, R., Özkan, E., and Gustafsson, F., 2014. “Tire radii estimation using a marginalized particle filter”. *IEEE Trans. Intell. Transp. Syst.*, **15**(2), pp. 663–672.
- [17] Anderson, B. D. O., and Moore, J. B., 1979. *Optimal Filtering*. Prentice Hall, Englewood Cliffs, New Jersey.
- [18] Schön, T. B., Gustafsson, F., and Nordlund, P.-J., 2005. “Marginalized particle filters for mixed linear nonlinear state-space models”. *IEEE Trans. Signal Process.*, **53**, pp. 2279–2289.
- [19] Bishop, C. M., 2006. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, NJ, USA.
- [20] Saha, S., and Gustafsson, F., 2012. “Particle filtering with dependent noise processes”. *IEEE Trans. Signal Process.*, **60**(9), pp. 4497–4508.
- [21] Gillespie, T., 1992. *Fundamentals of vehicle dynamics*. Society of Automotive Engineers, Inc.
- [22] Rajamani, R., 2006. *Vehicle Dynamics and Control*. Springer-Verlag.
- [23] Ryu, J., and Gerdes, J. C., 2004. “Estimation of vehicle roll and road bank angle”. In Amer. Control Conf.
- [24] Gustafsson, F., 2010. *Statistical Sensor Fusion*. Utbildningshuset/Studentlitteratur, Lund, Sweden.
- [25] Murphy, K. P., 2007. Conjugate Bayesian analysis of the Gaussian distribution. Tech. rep., UBC.
- [26] Özkan, E., Šmídl, V., Saha, S., Lundquist, C., and Gustafsson, F., 2013. “Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters”. *Automatica*, **49**(6), pp. 1566–1575.
- [27] Rao, C. R., 2001. *Linear Statistical Inference and its Applications*. Wiley.
- [28] Berntorp, K., and Di Cairano, S., 2017. “Particle Gibbs with ancestor sampling for identification of tire-friction parameters”. In IFAC World Congress.
- [29] Lindsten, F., Jordan, M. I., and Schön, T. B., 2014. “Particle Gibbs with ancestor sampling.”. *J. Machine Learning Res.*, **15**(1), pp. 2145–2184.