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

Tire-Stiffness and Vehicle-State Estimation Based on Noise-Adaptive Particle Filtering

Berntorp, Karl; Di Cairano, Stefano

TR2018-025 February 05, 2018

Abstract

We present a novel approach to learning online the tire stiffness and vehicle state using only wheel-speed and inertial sensors. The deviations from nominal stiffness values are treated as a Gaussian disturbance acting on the vehicle. We formulate a Bayesian approach, in which we leverage particle filtering and the marginalization concept to estimate in a computationally efficient way the tire-stiffness parameters and the vehicle state. In the estimation model, the process and measurement noise are dependent on each other, and we present an efficient approach to account for the dependence. Our algorithm outperforms some previously reported approaches, both in terms of accuracy and robustness, and the results indicate significantly improved performance compared to a standard particle filter. Monte-Carlo trials on several experimental data sets verify that the estimator identifies the tire stiffness on both snow and dry asphalt within 1% on average, with a settling time of a few seconds. On snow, the largest steady-state error in any Monte-Carlo trial is less than 4%.

IEEE Transactions on Control Systems Technology

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

Tire-Stiffness and Vehicle-State Estimation Based on Noise-Adaptive Particle Filtering

Karl Berntorp[1] and Stefano Di Cairano[1]

***Abstract*—We present a novel approach to learning online the tire stiffness and vehicle state using only wheel-speed and inertial sensors. The deviations from nominal stiffness values are treated as a Gaussian disturbance acting on the vehicle. We formulate a Bayesian approach, in which we leverage particle filtering and the marginalization concept to estimate in a computationally efficient way the tire-stiffness parameters and the vehicle state. In the estimation model, the process and measurement noise are dependent on each other, and we present an efficient approach to account for the dependence. Our algorithm outperforms some previously reported approaches, both in terms of accuracy and robustness, and the results indicate significantly improved performance compared to a standard particle filter. Monte-Carlo trials on several experimental data sets verify that the estimator identifies the tire stiffness on both snow and dry asphalt within 1% on average, with a settling time of a few seconds. On snow, the largest steady-state error in any Monte-Carlo trial is less than 4%.**

I. Introduction

The tire–road contact is the main responsible for generating the forces that alter the motion of a ground vehicle, and the knowledge of the variables related to the tire–road interaction is essential for advanced driver-assistance systems (ADAS). The interaction between road and vehicle is highly nonlinear, and individual tires may have different characteristics. A common way to simplify the tire–road modeling is to assume a static relationship between force and slip, but to reliably model the interaction is complicated even with this simplification. For example, when considering the tire stiffness (i.e., the initial slope of the force-slip curve), the measurements can vary considerably between experiments [1]–[3].

The force-slip relation is approximately linear for small slip values, which are typical for driving in normal conditions. It is therefore reasonable to model the longitudinal and lateral tire force as proportional to the respective slip quantity [4]. The initial slopes in the longitudinal and lateral directions of the tire are known as the longitudinal and lateral (cornering) stiffness, respectively. There is a dependence between the tire stiffness and the peak road-friction coefficient [3], [5], but a complicating factor is that the tire stiffness is highly dependent on several other factors, for example, tire and air temperature, tire pressure, and material and smoothness of the surface, which implies that the tire stiffness changes with time. The cornering stiffness is a key parameter when using the linear single-track vehicle model for control [4] and estimation [6]. Therefore, knowledge of the tire stiffness can be used directly, and is often imperative, in ADAS, and even partial knowledge of the tire stiffness can be used to classify surface types for road-condition monitoring. A factor in complicating this problem further is that the vehicle states involved in the tire-stiffness estimation are not directly measured in production vehicles.

Due to the relevance of the problem in the automotive domain, and the fact that it is not fully and satisfactorily solved by cost-effective approaches, there is a rich literature on tire-stiffness estimation. Some regression-based methods are found in [5], [7]–[10]. Identification of the longitudinal stiffness using nonlinear total least-squares is presented in [3], and [11] reports an observer based on the longitudinal dynamics and the LuGre friction model. In [12], a sliding-mode observer and an extended Kalman filter (EKF) are implemented in series to estimate forces, sideslip, and cornering stiffness. Several developed approaches use sensors that are nonstandard in passenger vehicles. A Kalman-filtering approach based on global position information (GPS) is developed in [13], whereas [14] exploits independent wheel actuation to estimate friction coefficient and cornering stiffness. In [10] a high precision localization system based on multiple GPS antennas and a high-end inertial measurement unit is used for obtaining some of the measurements. In [15] the tire-aligning moment measured by torque sensors in the steering rack is used in an observer-based approach, and [16] uses wireless in-tire strain gage sensors to estimate the parameters of a tire Brush model.

The focus of this paper is the development of a method for jointly estimating the stiffness parameters and the vehicle dynamics state using only sensors available in production cars according to current regulations, namely wheel-speed sensors and inexpensive, for example, mems-based, accelerometers and gyroscopes. Hence, as opposed to some of the previous contributions, we aim at using the same hardware components, and developing only new software, which is cost-effective in large-volumes industries such as automotive, since it results in no increase in per-unit cost. While our primary focus has been the lateral dynamics, the method developed here can be applied to either lateral or longitudinal dynamics, or to the two combined.

In our approach, the deviations from nominal stiffness values are considered as a Gaussian variable affecting the vehicle dynamics, and the mean and covariance of the process noise are unknown and time varying. The parameters of interest affect the vehicle state, which is only partially (and indirectly) observed through the inertial sensors. The considered problem is in general hard to solve for multiple reasons. First, the estimation quality of the vehicle state affects the identification of

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

the noise statistics, and vice versa. Second, the measurements from the inertial sensors are biased and significantly noisy. Third, the estimation model for our target application shows a dependence between the process and measurement noise. In this paper, we formulate the problem in a Bayesian framework as a joint state and parameter estimation problem, and develop a computationally efficient particle filter-based approach that accounts for the aforementioned bias and noise dependence.

Particle filtering is a technique that solves nonlinear, non-Gaussian estimation problems by generating random state trajectories and assigning a weight to them according to how well they predict the observations [17]. Particle filtering estimators have been previously reported in numerous automotive applications (see, e.g., [18]–[20]). When considering joint state and parameter estimation, a common approach is to augment the state vector with the vector of parameters and let the parameters be driven by artificial dynamics [12]. However, increasing the state dimension is problematic, especially in automotive applications, since the number of propagated particles and hence the computational burden usually increases exponentially with the dimensions, and the computational capabilities of automotive micro-controllers that run the estimation algorithm are very limited. The state-augmentation approach also relies on introducing artificial dynamics, which is not justified by the physics of the process and hence there is no intuition on how to select the artificial dynamics and their noise models. In our approach, we avoid the introduction of artificial dynamics and rely instead on marginalization [21] and propagation of the sufficient statistics of the noise parameters, conditioned on the estimated vehicle states, by exploiting the concept of conjugate priors [22]. Dependent noise when the statistics of both noise sources are unknown, which might have implications on the observability and identifiability of the model, has been considered in a particle-filtering framework in [23]. Another work related to our developments is [24], which considers estimation of noise sources in a similar framework as in this paper and gives explicit expressions for the case of uncorrelated noise sources.

In this paper[1] for designing the estimator for the stiffness parameters and the states of the vehicle dynamics, we start from the methods developed in [23], [24]. Due to the models resulting from our application and the fact that in our application the measurement noise statistics can be determined a priori, we modify the methods in [23], [24] to handle the case of dependent noise sources and partially known measurement noise. Since usually automotive-grade inertial sensors deliver biased measurements, we must also include online bias estimation. Finally, to reduce the computational burden as required for implementation in micro-controllers with limited resources, such as the ones that run automotive applications today, we develop suitable approximations based on moment matching that preserve asymptotic properties. To validate the approach, we extensively evaluate our method in simulations and on experimental data, both on snow and asphalt, and we compare with other particle filters, with a recursive least-squares (RLS) approach, and with an EKF approach. The experimental results also show that the probability density functions for the estimated quantities are non-Gaussian. This validates our choice of a particle filter, which does not require a Gaussian distribution of the estimates.

[1]A preliminary version of this work was presented in [25], [26]. The current, elaborated, version contains a detailed explanation of the estimation algorithm development and the approximation to increase computation speed, comparisons with other methods, a significantly expanded experimental evaluation based on several datasets, and an algorithm to detect loss of observability in the estimator.

Outline: Section II presents the vehicle model, sensor setup, and the problem definition. Section III provides the background material on particle filtering necessary to understand our approach, which is explained in Section IV. Section V addresses implementation aspects. Simulation and experimental evaluation of the method follow in Sections VI and VII, respectively, and Section VIII concludes the paper.

Preliminaries and Notation: The sensor configuration is crucial for the algorithm design and the limits of performance. We focus on a sensor setup that is standard in modern cars equipped with electronic stability control, namely measurements of the lateral (optionally also longitudinal) acceleration, the yaw rate, the wheel-speed sensors, and the steering-wheel angle. Furthermore, automotive-grade inertial sensors have (time-varying) bias, which needs to be taken into account in any realistic implementation. We tackle the joint estimation problem in a Bayesian framework. With $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k})$, we mean the posterior density function of the state trajectory $\boldsymbol{x}_{0:k}$ from time index 0 to time index k given the measurement sequence $\boldsymbol{y}_{0:k} := \{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_k\}$. We define $\boldsymbol{f}_k := \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ for a function $\boldsymbol{f}$, where $\boldsymbol{u}$ is the deterministic input. Throughout, for a vector $\boldsymbol{x}$, $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicates that $\boldsymbol{x}$ is Gaussian distributed with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and $|\boldsymbol{\Sigma}|$ is the determinant of the matrix $\boldsymbol{\Sigma}$. The notation $\mathrm{St}(\boldsymbol{\mu}, \Upsilon, \nu)$ means the multivariate Student-t distribution with mean $\boldsymbol{\mu}$, scaling Υ, and ν degrees of freedom. Similarly, $\mathrm{NiW}(\gamma, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \nu)$ reads the Normal-inverse-Wishart distribution with statistics (hyperparameters) summarized in $S := (\gamma, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \nu)$. The notation $\hat{\boldsymbol{z}}_{k|m}$ denotes the estimate of $\boldsymbol{z}$ at time index k given measurements up to time index m.

II. Modeling and Problem Formulation

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, [4], [27]. The approach developed here can handle several cases where such assumptions do not hold, at the price of increased computational burden, for instance due to using a two-track model.

Assumption 1: The steering angles of the front right and left wheels are the same, denoted by δ.

Assumption 2: The steering and acceleration commands are small enough such that $\sin(\delta) \approx \delta$ and the vehicle operates in the linear region of the tire-force curve, with negligible inclination and bank angles.

Assumption 3: The left and right wheels on each axle have the same stiffness.

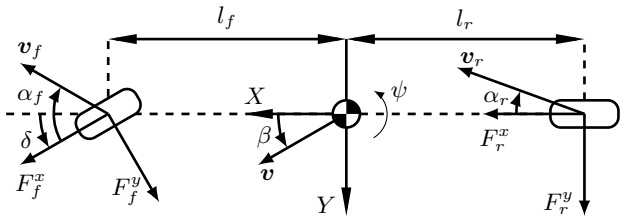

Fig. 1. A schematics of the single-track model and related notation.

Assumption 4: The dominant forces are the tire–road contact forces. Influences due to wind and air resistance are ignored.

Assumption 5: The lateral dynamics is approximately decoupled from the longitudinal dynamics.

Under these assumptions, our approach for joint state and parameter estimation can be based on a planar single-track model, see Fig. 1. 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 $\boldsymbol{x} = \begin{bmatrix} v^X & v^Y & \dot{\psi} \end{bmatrix}^{\mathrm{T}}$, where v^X is the longitudinal velocity of the vehicle, v^Y is the lateral velocity of the vehicle, and $\dot{\psi}$ is the yaw rate, the equations of motion are

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

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

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

where m is the mass, I is the inertia, and $F^X = F_f^x \cos(\delta) + F_r^x$. In several ADAS applications, the lateral dynamics are of main interest [4], [27]. In this case, the longitudinal dynamics can be discarded, assuming that the longitudinal velocity is provided by some other estimation logic.[2] From Assumption 2, the longitudinal and lateral tire forces can be expressed as linear functions of the wheel slip λ and slip angle α, respectively,

$$F^x \approx C^x\lambda, \quad F^y \approx C^y\alpha, \tag{2}$$

where C^x and C^y are the longitudinal and lateral stiffness, respectively. We define the wheel slip as in [28],

$$\lambda = \frac{v^X - R_w\omega}{\max(v^X, R_w\omega)}, \tag{3}$$

where ω is the wheel rotation rate and R_w is the effective wheel radius. The slip angles are approximated as

$$\alpha_f \approx \delta - \frac{v^Y + l_f\dot{\psi}}{v^X}, \quad \alpha_r \approx \frac{l_r\dot{\psi} - v^Y}{v^X}. \tag{4}$$

In (3), (4), we use the velocity at the center of mass instead of the velocity at the center of the wheel. The complete vehicle model (1)–(4) is nonlinear in v^X. More importantly, there are bilinearities between both states, and states and parameters. Considering both longitudinal and lateral dynamics allows to properly account for the coupling in (1), but increases computational burden. The wheel rotation rates ω_f, ω_r, and the steer angle δ, form the input vector $\boldsymbol{u}$, which is assumed known in the following. This is consistent with many navigation systems, where dead reckoning is used to decrease state dimension. Note that the wheel rotation rates could be alternatively considered as noisy measurements. Once again, this would increase the dimensions of the estimation problem, which is undesired in automotive applications due to the increase in the computational load must be limited. After discretization, (1)–(4) can be written as

$$\boldsymbol{x}_{k+1} = \bar{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{5}$$

where $\bar{\boldsymbol{f}}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ is the discretized counterpart to (1)–(4). In this work, we use a standard Euler discretization scheme, but other methods can be used. For instance, considering lateral dynamics only, by plugging (2), (4) into (1b) and discretizing according to an Euler discretization scheme with sampling time T_s,

$$v_{k+1}^Y = v_k^Y + \frac{T_s}{m}\left(C_r^y\frac{l_r\dot{\psi} - v^Y}{v^X} + C_f^y\Big(\delta - \frac{l_f\dot{\psi} + v^Y}{v^X}\Big)\right). \tag{6}$$

A. Estimation Model

Eq. (5) can be rewritten by decomposing the stiffness parameters into one known nominal part and one unknown part,

$$C^x = C_n^x + \Delta C^x, \quad C^y = C_n^y + \Delta C^y, \tag{7}$$

where C_n is the nominal value of the stiffness, for example, a priori determined on a nominal surface, and ΔC is a time-varying, unknown part. We define

$$\boldsymbol{w}_k := \begin{bmatrix} \Delta C_f^x & \Delta C_f^y & \Delta C_r^y \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{n_w} \tag{8}$$

as random process noise acting on the otherwise deterministic system. Note that we only include the front longitudinal stiffness, that is, $n_w = 3$, because the system is poorly observable with the considered sensor setup if both front and rear longitudinal tire stiffness are considered. Here we consider the front longitudinal stiffness because we will validate the algorithm with a front-wheel drive vehicle, where the longitudinal force is expressed only at the front tires. For rear-wheel drive vehicles, one will substitute ΔC_f^x by ΔC_r^x, while for all-wheel drive vehicles the rear and front longitudinal stiffness should be estimated iteratively.

We model the noise term $\boldsymbol{w}_k$ as Gaussian distributed according to $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the unknown, usually time varying, mean and covariance. Inserting (7) into (5) allows us to write the dynamics as

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k)\boldsymbol{w}_k. \tag{9}$$

Thus, in (6) the decomposition (7) leads to

$$v_{k+1}^Y = \underbrace{v_k^Y + \frac{T_s}{m}\left(C_{r,n}^y\frac{l_r\dot{\psi} - v^Y}{v^X} + C_{f,n}^y\Big(\delta - \frac{l_f\dot{\psi} + v^Y}{v^X}\Big)\right)}_{\boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k)} + \underbrace{\frac{T_s}{m}\begin{bmatrix} \delta - \frac{l_f\dot{\psi} + v^Y}{v^X} & \frac{l_r\dot{\psi} - v^Y}{v^X} \end{bmatrix}}_{\boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k)} \underbrace{\begin{bmatrix} \Delta C_f^y & \Delta C_r^y \end{bmatrix}^{\mathrm{T}}}_{\boldsymbol{w}_k}. \tag{10}$$

[2]For example, estimated using the wheel-speed sensors, transmission-shaft speed sensors, accelerometers, or a combination of them.

Hence, the vehicle dynamics naturally leads to an interpretation of the unknown part of the tire stiffness as a disturbance on the process with statistics with unknown parameters, which motivates our noise-adaptive approach.

In the following, we are interested in estimating both the state $\boldsymbol{x}_k$ and the parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$, being the mean and variance of the process noise $\boldsymbol{w}_k$. One interpretation of the parameters is that the mean models the stiffness variations based on the surface type, such as asphalt or snow, and the variance models the uncertainty due to either variations on a surface, such as road unevenness, patches of loose snow, road in mixed conditions, or other unmodeled effects.

The available measurements are the longitudinal and lateral accelerations, a_m^X, a_m^Y, and the yaw rate $\dot{\psi}_m$, forming the measurement vector $\boldsymbol{y}_k = [a_m^X \; a_m^Y \; \dot{\psi}_m]^{\mathrm{T}}$. To relate $\boldsymbol{y}_k$ to the states, note that a^X and a^Y can be extracted from the right-hand sides of (1a) and (1b), respectively, after dividing the vehicle mass and shifting over the first terms on the right-hand sides. The yaw-rate measurement is directly related to the yaw rate. When turning, a car typically exhibits a slight roll angle about the vehicles forward axis. The roll angle causes the lateral acceleration component to pick up parts of the gravitational acceleration, namely $\sin\phi g \approx \phi g$, where g is the gravitational acceleration and ϕ is the roll angle. The roll angle is proportional to the lateral acceleration according to $a^Y \approx c_\phi \phi$, where c_ϕ is the roll-angle gradient. Hence, the influence of the roll angle on the measurements can be suppressed by dividing the measured lateral acceleration with the constant $1 + c_\phi g$, see [29].

Automotive-grade inertial sensors have bias $\boldsymbol{b}$, which must be modeled for any realistic implementation. The bias for each sensor can be modeled as a first-order Markov process $\dot{b} = -b/\tau + w_b$ with time constant τ, where $b/\tau \approx 0$. This approximation is reasonable for estimation processes lasting some minutes, since τ typically is in the order of several minutes [30]. Hence, we model the bias as a random walk,

$$\boldsymbol{b}_{k+1} = \boldsymbol{b}_k + \boldsymbol{w}_{b,k}, \tag{11}$$

where $\boldsymbol{b}_k = [b_{x,k} \; b_{y,k} \; b_{\psi,k}]^{\mathrm{T}} \in \mathbb{R}^3$ are the bias terms for the acceleration vector and yaw rate, and $\boldsymbol{w}_{b,k}$ is modeled as zero-mean Gaussian with known covariance matrix $\boldsymbol{Q}$, $\boldsymbol{w}_{b,k} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$. The characteristics of the noise source $\boldsymbol{w}_{b,k}$ depend only on the sensor, and hence can be determined a priori from an Allan-variance analysis [30], thus avoiding the need for real-time estimation of $\boldsymbol{Q}$. The measurement model can be written as

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{b}_k + \boldsymbol{d}(\boldsymbol{x}_k, \boldsymbol{u}_k)\boldsymbol{w}_k + \boldsymbol{e}_k, \tag{12}$$

where

$$\boldsymbol{d}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \begin{bmatrix} T_s \boldsymbol{g}_1(\boldsymbol{x}_k, \boldsymbol{u}_k)^{\mathrm{T}} & T_s \boldsymbol{g}_2(\boldsymbol{x}_k, \boldsymbol{u}_k)^{\mathrm{T}} & \boldsymbol{0}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}},$$

in which $\boldsymbol{g}_j$ is the jth row of $\boldsymbol{g}$ in (9) and $\boldsymbol{e}_k \in \mathbb{R}^{n_e}$, $n_e = 3$, is the Gaussian zero-mean noise from the inertial sensors, $\boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$, where $\boldsymbol{R}$ is determined a priori.

Now, the joint Gaussian distribution of the tire-stiffness parameters $\boldsymbol{w}_k$ and measurement noise $\bar{\boldsymbol{e}}_k$ can be written as

$$\bar{\boldsymbol{w}}_k = \begin{bmatrix} \boldsymbol{w}_k^{\mathrm{T}} & \bar{\boldsymbol{e}}_k^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \sim \mathcal{N}\left(\bar{\boldsymbol{\mu}}_k, \bar{\boldsymbol{\Sigma}}_k\right),$$

where we have introduced the short-hand notation $\bar{\boldsymbol{e}}_k = \boldsymbol{d}(\boldsymbol{x}_k, \boldsymbol{u}_k)\boldsymbol{w}_k + \boldsymbol{e}_k$, and where

$$\bar{\boldsymbol{\mu}}_k = \begin{bmatrix} \boldsymbol{\mu}_k \\ \boldsymbol{d}_k \boldsymbol{\mu}_k \end{bmatrix}, \tag{13a}$$

$$\bar{\boldsymbol{\Sigma}}_k = \begin{bmatrix} \boldsymbol{\Sigma}_k & \boldsymbol{\Sigma}_k \boldsymbol{d}_k^{\mathrm{T}} \\ \boldsymbol{d}_k \boldsymbol{\Sigma} & \boldsymbol{d}_k \boldsymbol{\Sigma}_k \boldsymbol{d}_k^{\mathrm{T}} + \boldsymbol{R} \end{bmatrix}. \tag{13b}$$

In (13), $\boldsymbol{d}_k := \boldsymbol{d}_k(\boldsymbol{x}_k, \boldsymbol{u}_k)$. Thus, the noise sources with the structure given by (13b) are dependent. In this work we need to estimate the process-noise statistics $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, which are embedded in (13), together with the state trajectory. As for the bias, the measurement noise stemming from the inertial sensors $\boldsymbol{R}$ is also determined a priori.

Remark 1: It is possible to treat the measurement-noise parameters as unknown and include any bias terms in the mean of the measurement noise. Then, the resulting measurement-noise parameters to be estimated would include a combination of stiffness parameters, measurement noise, and bias terms. We believe that, for a computationally efficient implementation, which is needed in resource-limited automotive microcontrollers, it is beneficial to treat the noise parameters from the inertial measurements as known, since these depend on the sensor only, and hence can be determined a priori. Furthermore, the characteristics of the bias can also, as pointed out, be determined offline for the same reason. This information can then be utilized for more reliable estimation of the noise parameters.

Remark 2: Our proposed method can be extended to estimate a piecewise-affine tire-stiffness map, such as described in [4], by combining detection of large slip angles with anomaly detection on estimated stiffness values. This covers the case of maneuvers at the limit of the stability region of the vehicle. This extension will be the subject of future studies.

B. Observability

Observability can be analyzed by augmenting the dynamic model (5) with the stiffness parameters, model them as a random walk, and derive the observability Gramian by linearization. In our case, assuming nonzero steering angle and wheel slip, it can be shown that the Gramian is nonsingular, and hence the system is weakly observable.

C. Problem Formulation

We want to estimate the tire-stiffness parameters and vehicle state online. In a Bayesian setting, this can be expressed as jointly learning the parameters $\boldsymbol{\theta}_k := \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ of the Gaussian process noise $\boldsymbol{w}_k$ and estimating the state vector $\boldsymbol{x}_k$. We approach this problem in the following way. Given the system model (9)–(12), and dependent Gaussian noise between $\boldsymbol{w}_k$ and $\bar{\boldsymbol{e}}_k$ characterized by (13), where the unknown parameters $\boldsymbol{\theta}_k$ may be time varying, we recursively estimate

$$p(\boldsymbol{x}_k | \boldsymbol{y}_{0:k}), \tag{14a}$$

$$p(\boldsymbol{\theta}_k | \boldsymbol{y}_{0:k}). \tag{14b}$$

Because the measurements are affected by the unknown bias, to compute (14), we also need to estimate the bias density as an intermediate step in the method.

III. Background on Particle Filtering and Conjugate Priors

First, we provide background material on particle filtering and conjugate priors that are necessary for the understanding of the development of the joint state and stiffness estimator in the subsequent section.

A. Particle Filtering with Dependent Noise

Particle filters [17] approximate the posterior of the state trajectory using a weighted sum of particles

$$p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k}) \approx \sum_{i=1}^{N} q_k^i \delta(\boldsymbol{x}_{0:k} - \boldsymbol{x}_{0:k}^i), \tag{15}$$

where $\delta(\cdot)$ is the Dirac delta mass, N is the number of particles, and q_k^i is the importance weight for the ith state trajectory $\boldsymbol{x}_{0:k}^i$. In this work we use a relatively standard sequential importance resampling (SIR) based particle filter to propagate the samples and update the weights [17]. In general, the particles are sampled from a proposal distribution $\pi(\boldsymbol{x}_{k+1}|\boldsymbol{x}_{0:k}^i, \boldsymbol{y}_{0:k+1})$. For dependent noise [23], the weight update is performed as

$$q_k^i \propto q_{k-1}^i \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k}^i, \boldsymbol{y}_{0:k-1}) p(\boldsymbol{x}_k^i|\boldsymbol{x}_{0:k-1}^i, \boldsymbol{y}_{0:k-1})}{\pi(\boldsymbol{x}_k^i|\boldsymbol{x}_{0:k-1}^i, \boldsymbol{y}_{0:k})} \tag{16}$$

where $p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k}^i, \boldsymbol{y}_{0:k-1})$ is the likelihood. If the proposal is chosen equal to $p(\boldsymbol{x}_k^i|\boldsymbol{x}_{0:k-1}^i, \boldsymbol{y}_{0:k-1})$, (16) simplifies to

$$q_k^i \propto q_{k-1}^i p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k}^i, \boldsymbol{y}_{0:k-1}). \tag{17}$$

In case of known parameters and Gaussian noise, (17) can be computed analytically [23]. However, in our case, since the unknown process-noise parameters affect both the measurement and prediction step (see (12), (13)),

$$p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k}^i, \boldsymbol{y}_{0:k-1}), \tag{18a}$$
$$p(\boldsymbol{x}_{k+1}^i|\boldsymbol{x}_{0:k}^i, \boldsymbol{y}_{0:k}), \tag{18b}$$

respectively, the weight update will depend upon the parameter estimates.

B. Learning with Conjugate Priors

Conjugate priors [22], [31] are useful for parameter learning.

Definition 1: Given a likelihood, the conjugate prior is the prior distribution such that the prior and posterior are in the same family of distributions. ■

Thus, for a conjugate prior, the prior and posteriors are of the same type, and the estimation problem simplifies to updating the hyperparameters, that is, the parameters of the distribution. Lemma 1 provides an explicit expression of the conjugate prior for Gaussian likelihoods [31].

Lemma 1 ([31]): For multivariate Normal data $\bar{\boldsymbol{w}} \in \mathbb{R}^d$ with unknown mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, a Normal-inverse-Wishart distribution defines the conjugate prior $p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) := \mathrm{NiW}(\gamma_{k|k}, \hat{\boldsymbol{\mu}}_{k|k}, \boldsymbol{\Lambda}_{k|k}, \nu_{k|k})$, through the hierarchical model

$$\begin{aligned} \boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k &\sim \mathcal{N}(\hat{\boldsymbol{\mu}}_{k|k}, \gamma_{k|k}\boldsymbol{\Sigma}_k), \\ \boldsymbol{\Sigma}_k &\sim \mathrm{iW}(\nu_{k|k}, \boldsymbol{\Lambda}_{k|k}) \\ &\propto |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}(\nu_{k|k}+d+1)} e^{\left(-\frac{1}{2}\mathrm{tr}(\boldsymbol{\Lambda}_{k|k}\boldsymbol{\Sigma}_k^{-1})\right.} \end{aligned}$$

where $\mathrm{tr}(\cdot)$ is the trace operator. ■

The statistics $S_{k|k} := (\gamma_{k|k}, \hat{\boldsymbol{\mu}}_{k|k}, \boldsymbol{\Lambda}_{k|k}, \nu_{k|k})$ in Lemma 1 can be updated as suggested in [24], [32],

$$\gamma_{k|k} = \frac{\gamma_{k|k-1}}{1+\gamma_{k|k-1}}, \tag{19a}$$
$$\hat{\boldsymbol{\mu}}_{k|k} = \hat{\boldsymbol{\mu}}_{k|k-1} + \gamma_{k|k}\boldsymbol{z}_k, \tag{19b}$$
$$\nu_{k|k} = \nu_{k|k-1} + 1, \tag{19c}$$
$$\boldsymbol{\Lambda}_{k|k} = \boldsymbol{\Lambda}_{k|k-1} + \frac{1}{1+\gamma_{k|k-1}}\boldsymbol{z}_k\boldsymbol{z}_k^{\mathrm{T}}, \tag{19d}$$
$$\boldsymbol{z}_k = \bar{\boldsymbol{w}}_k - \hat{\boldsymbol{\mu}}_{k|k-1}. \tag{19e}$$

For time-varying parameters, the prediction step consists of

$$\begin{aligned} \gamma_{k|k-1} &= \frac{1}{\lambda}\gamma_{k-1|k-1}, \\ \hat{\boldsymbol{\mu}}_{k|k-1} &= \hat{\boldsymbol{\mu}}_{k-1|k-1}, \\ \nu_{k|k-1} &= \lambda\nu_{k-1|k-1}, \\ \boldsymbol{\Lambda}_{k|k-1} &= \lambda\boldsymbol{\Lambda}_{k-1|k-1}, \end{aligned} \tag{20}$$

where $\lambda \in [0, 1]$ provides exponential forgetting. Further, for a Normal-inverse-Wishart prior, the predictive distribution of the data $\bar{\boldsymbol{w}}$ is a Student-t,

$$\mathrm{St}(\hat{\boldsymbol{\mu}}_{k|k-1}, \tilde{\boldsymbol{\Lambda}}_{k|k-1}, \nu_{k|k-1} - d + 1), \tag{21}$$

with

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

The key question is how to generate $\bar{\boldsymbol{w}}_k$ in (19e) so that the dependence structure is taken into account. Our approach for solving this is discussed in Sec. IV-B. Also, note that in a particle implementation, each particle keeps its own estimate of the statistics.

The distribution of the parameters $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ is in general computed based on the state trajectory and/or the measurement history, depending on the assumptions on the noise processes. From (18), knowing both the state and measurement trajectory leads to full knowledge about $\bar{\boldsymbol{w}}_{0:k}$. Hence, for a specific realization of the state trajectory $\boldsymbol{x}_{0:k}^i$ and of the measurement history, the posterior for the noise parameters can be written as a function of the noise realization corresponding to the state trajectory,

$$p(\boldsymbol{\theta}_k|\bar{\boldsymbol{w}}_{0:k}^i). \tag{22}$$

Using Bayes' rule on (22) leads to

$$p(\boldsymbol{\theta}_k|\bar{\boldsymbol{w}}_{0:k}) \propto p(\bar{\boldsymbol{w}}_k|\boldsymbol{\theta}_k) p(\boldsymbol{\theta}_k|\bar{\boldsymbol{w}}_{0:k-1}). \tag{23}$$

The posterior of the parameters in (23) is composed of a likelihood $p(\bar{\boldsymbol{w}}_k|\boldsymbol{\theta}_k)$ and a prior $p(\boldsymbol{\theta}_k|\bar{\boldsymbol{w}}_{0:k-1})$.

Suppose now that the predictive distribution $p(\boldsymbol{\theta}_k|\bar{\boldsymbol{w}}_{0:k-1})$ in (23) is Normal-inverse-Wishart, that is,

$$p(\boldsymbol{\theta}_k|\bar{\boldsymbol{w}}_{0:k-1}) = \mathrm{NiW}(\gamma_{k|k-1}, \hat{\boldsymbol{\mu}}_{k|k-1}, \boldsymbol{\Lambda}_{k|k-1}, \nu_{k|k-1}). \tag{24}$$

By (23), (24), and Lemma 1, the posterior is also Normal-inverse-Wishart,

$$p(\boldsymbol{\theta}_k|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{0:k}) = \mathrm{NiW}(\gamma_{k|k}, \hat{\boldsymbol{\mu}}_{k|k}, \boldsymbol{\Lambda}_{k|k}, \nu_{k|k}). \tag{25}$$

Thus, to compute (14b), we integrate over the state trajectory,

$$p(\boldsymbol{\theta}_k|\boldsymbol{y}_{0:k}) = \int p(\boldsymbol{\theta}_k|\boldsymbol{x}_{0:k},\boldsymbol{y}_{0:k})p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k})\mathrm{d}\boldsymbol{x}_{0:k} \approx \sum_{i=1}^{N} q_k^i p(\boldsymbol{\theta}_k|\boldsymbol{x}_{0:k}^i,\boldsymbol{y}_{0:k})\delta(\boldsymbol{x}_{0:k}-\boldsymbol{x}_{0:k}^i), \quad (26)$$

which has complexity $\mathcal{O}(N)$, where $p(\boldsymbol{\theta}_k|\boldsymbol{x}_{0:k}^i,\boldsymbol{y}_{0:k})$ is given by (25). The unknown parameters can be extracted from (26); for instance, the minimum mean-square estimate of $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ can be found as

$$\hat{\boldsymbol{\mu}}_k = \sum_{i=1}^{N} q_k^i \hat{\boldsymbol{\mu}}_{k|k}^i, \quad (27a)$$

$$\hat{\boldsymbol{\Sigma}}_k = \sum_{i=1}^{N} q_k^i \left(\frac{1}{\nu_{k|k}-4}\boldsymbol{\Lambda}_{k|k}^i + (\hat{\boldsymbol{\mu}}_{k|k}^i - \hat{\boldsymbol{\mu}}_k)(\hat{\boldsymbol{\mu}}_{k|k}^i - \hat{\boldsymbol{\mu}}_k)^{\mathrm{T}} \right). \quad (27b)$$

In the next section we describe how to connect the measurement update (18a) and prediction step (18b) in the particle filter with the update of the noise statistics (19) and the bias components.

IV. Joint State and Tire-Stiffness Estimation

Considering also the bias, we approach our estimation problem by estimating the density $p(\boldsymbol{b}_k, \boldsymbol{\theta}_k, \boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k})$, that is, the joint posterior conditioned on all measurements from time index 0 to k. We decompose

$$p(\boldsymbol{b}_k, \boldsymbol{\theta}_k, \boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k}) = p(\boldsymbol{b}_k|\boldsymbol{\theta}_k, \boldsymbol{x}_{0:k}, \boldsymbol{y}_{0:k})p(\boldsymbol{\theta}_k|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{0:k}) \cdot p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k}). \quad (28)$$

The three densities on the right-hand side of (28) are estimated recursively. The third term on the right-hand side is given by (15). The key idea is that given the state trajectory, we can update the distribution of the noise parameters, that is, the second distribution on the right-hand side of (28) as given by (25). Similarly, given the parameters and the state trajectory, the estimation of the posterior density of the bias simplifies considerably. However, because of the structure of the estimation model, the three densities are tightly connected.

In solving our estimation problem, we approximate the posterior of the state trajectory using particle filtering, see Section III-A, and the update of the noise statistics conditioned on the state trajectory is done according to Section III-B. However, the likelihood (18a) and predictive density (18b) are not straightforward to compute because of the partially known noise. Furthermore, the measurement update of the noise statistics given by (19) depends on the realization of the noise term through the error equation (19e), which is connected to the predictive density (18b) because of the dependent noise.

Remark 3: Note that only the parameters of $\boldsymbol{w}_k$ are unknown, so (19) and (20) are only applied to the process noise (i.e., $\bar{\boldsymbol{w}} \in \mathbb{R}^{n_w}$ in Lemma 1), which decreases the dimensionality from $d = 6$ to $n_w = 3$. However, to include information from the likelihood into the update of the parameters, thereby taking advantage of that the noise sources are dependent, $\bar{\boldsymbol{w}}_k$ in (19e) must be generated through the measurement model. We will describe this in more detail in Sec. IV-B.

A. Bias Estimation

The bias estimation update relies on having computed both the posterior for the state trajectory and the one for the noise parameters, (26). Thus, the bias estimation is concerned with computing the posterior $p(\boldsymbol{b}_k|\boldsymbol{\theta}_k, \boldsymbol{x}_{0:k}, \boldsymbol{y}_{0:k})$, based on the following observation. First, the prediction model (11) of the bias states is a random walk, which is linear and Gaussian. Second, the dynamics of the bias states are independent on both the unknown process noise and the vehicle states. Hence, the Kalman predictor is the optimal predictor, resulting in

$$\hat{\boldsymbol{b}}_{k+1|k}^i = \hat{\boldsymbol{b}}_{k|k}^i, \quad \boldsymbol{P}_{k+1}^i = \boldsymbol{P}_k^i + \boldsymbol{Q}. \quad (29)$$

For the measurement update, (12) conditioned on the state trajectory and the noise parameters is affine in $\boldsymbol{b}_k$ with known, Gaussian measurement noise $\bar{\boldsymbol{e}}_k$. Hence, the measurement update per particle consists of a Kalman update,

$$\begin{aligned} \hat{\boldsymbol{b}}_{k|k}^i &= \hat{\boldsymbol{b}}_{k|k-1}^i + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{h}_k^i - \boldsymbol{d}_k^i\hat{\boldsymbol{\mu}}_{k|k}^i - \hat{\boldsymbol{b}}_{k|k-1}^i), \\ \boldsymbol{K}_k^i &= \boldsymbol{P}_{k|k-1}^i(\boldsymbol{S}_k^i)^{-1}, \\ \boldsymbol{P}_{k|k}^i &= \boldsymbol{P}_{k|k-1}^i - \boldsymbol{K}_k^i(\boldsymbol{S}_k^i)^{-1}(\boldsymbol{K}_k^i)^{\mathrm{T}}, \\ \boldsymbol{S}_k^i &= \boldsymbol{P}_{k|k-1}^i + \boldsymbol{R} + \boldsymbol{d}_k^i\hat{\boldsymbol{\Sigma}}_{k|k}^i(\boldsymbol{d}_k^i)^{\mathrm{T}}, \end{aligned} \quad (30)$$

where $\hat{\boldsymbol{\Sigma}}_{k|k}^i = \boldsymbol{\Lambda}_{k|k}^i/(\nu_{k|k}-4)$. Note that (29) and (30) are prediction and update equations applied to each particle in the particle filter, and not a stand-alone, cascade Kalman filter. The first term on the right-hand side of (28) can now be written as a Gaussian for each particle, where the mean and covariance of each Gaussian is obtained from (30). The second and third terms are given by (15) and (25), respectively, and (28) can therefore be written as a weighted sum of the three distributions, where the weights are given by the particle filter. The computation of the weights is done by marginalization of the bias states and parameters, as discussed next.

B. State Update and Prediction through Marginalization

To estimate the state, we need expressions for (18a), (18b), which are then used to compute (15), from which the filtering distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k})$ is obtained. We first state the following lemma on transformation of variables in probabilities.

Lemma 2 (See [33]): Let X be a random variable with probability density function $p(\boldsymbol{x})$. Let $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x})$ and assume that $\boldsymbol{g}^{-1}(\boldsymbol{y})$ exists with Jacobian $J(\boldsymbol{y}) = \frac{\partial \boldsymbol{g}^{-1}}{\partial \boldsymbol{y}}$. Then, the random variable $Y = \boldsymbol{g}(X)$ has the probability density function

$$p(\boldsymbol{y}) = |J(\boldsymbol{y})|p(\boldsymbol{g}^{-1}(\boldsymbol{y})).$$

■

Consider first the weight update and corresponding measurement likelihood. To compute the likelihood (18a), we apply Lemma 2 to our scenario, which results in

$$p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{0:k-1}) = p(\bar{\boldsymbol{e}}_k(\boldsymbol{y}_k, \boldsymbol{x}_k)|\boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{0:k-1}). \quad (31)$$

Furthermore, the knowledge of both the state trajectory $\boldsymbol{x}_{0:k-1}$ and measurement sequence $\boldsymbol{y}_{0:k-1}$ gives full information about $\bar{\boldsymbol{e}}_{0:k-1}$. Eq. (31) therefore becomes

$$p(\bar{\boldsymbol{e}}_k(\boldsymbol{y}_k, \boldsymbol{x}_k)|\boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{0:k-1}) = p(\bar{\boldsymbol{e}}_k(\boldsymbol{y}_k, \boldsymbol{x}_k)|\bar{\boldsymbol{e}}_{0:k-1}). \quad (32)$$

Using marginalization, (31) can be rewritten as

$$p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k},\boldsymbol{y}_{0:k-1}) = \int p(\boldsymbol{y}_k|\boldsymbol{\theta}_k,\boldsymbol{x}_k) \cdot p(\boldsymbol{\theta}_k|\boldsymbol{x}_{0:k-1},\boldsymbol{y}_{0:k-1})\,\mathrm{d}\boldsymbol{\theta}_k. \quad (33)$$

The total measurement noise $\bar{\boldsymbol{e}}_k = \boldsymbol{d}_k\boldsymbol{w}_k + \boldsymbol{e}_k$ is the sum of a Gaussian $\boldsymbol{e}_k$ with known statistics and a Gaussian $\boldsymbol{w}_k$ with unknown statistics, given by a NiW prior. Integrating over the product of a NiW prior and a Gaussian distribution leads to a Student-t distribution [31]. This implies that (33) is a mixture of Gaussian and Student-t distributions, which can be expressed as an infinite weighted sum of exponential functions $\phi_k(\boldsymbol{z})$, $\boldsymbol{z} \in \mathbb{R}^d$, on the form [34]

$$p(\boldsymbol{z}) = \sum_{k=0}^{\infty} \zeta_k \phi_k(\boldsymbol{z}), \quad (34)$$

where

$$\phi_k(\boldsymbol{z}) \propto (\|\boldsymbol{z}\|^2)^k \exp\left(-\frac{1}{2}\boldsymbol{z}^{\mathrm{T}}\boldsymbol{z}/\sigma\right),$$

and where it is assumed that $\phi_k(\boldsymbol{z})$ has a diagonal covariance matrix with σ on the diagonal. Computing the coefficients ζ_k of the sum in (34) involves integrations that are computationally demanding. Since the tire-stiffness estimator needs to operate in real-time at several tens of Hz in automotive platforms, which have limited computing capabilities, we need to seek for a computationally efficient implementation. Therefore, we seek an alternative expression based on moment matching. Specifically, we relax the assumption on Gaussian distribution of the inertial measurements by matching the first two moments (mean and covariance) of the Gaussian with a corresponding Student-t using the following approximation.

Approximation 1: We approximate (33) with the Student-t distribution

$$p(\bar{\boldsymbol{e}}_k(\boldsymbol{y}_k,\boldsymbol{x}_k)|\bar{\boldsymbol{e}}_{0:k-1}) \approx \mathrm{St}(\hat{\boldsymbol{\mu}}_{\bar{e},k|k-1}, \tilde{\boldsymbol{\Lambda}}_{\bar{e},k|k-1}, \tilde{\nu}_{k|k-1}), \quad (35)$$

with mean $\hat{\boldsymbol{\mu}}_{\bar{e},k|k-1}$ and scale $\tilde{\boldsymbol{\Lambda}}_{\bar{e},k|k-1}$ as

$$\begin{aligned}\hat{\boldsymbol{\mu}}_{\bar{e},k|k-1} &= \boldsymbol{d}_k\hat{\boldsymbol{\mu}}_{k|k-1},\\ \tilde{\boldsymbol{\Lambda}}_{\bar{e},k|k-1} &= \frac{1+\gamma_{k|k-1}}{\tilde{\nu}_{k|k-1}}\boldsymbol{d}_k\boldsymbol{\Lambda}_{k|k-1}\boldsymbol{d}_k^{\mathrm{T}}\\ &\quad + \frac{\tilde{\nu}_{k|k-1}-2}{\tilde{\nu}_{k|k-1}}(\boldsymbol{R}+\boldsymbol{P}_{k|k-1}).\end{aligned}$$

Approximation 1 is based on moment matching of a known Gaussian with a Student-t, by choosing the smallest common degree of freedom, and also implicitly involves marginalization of the bias state. The Student-t is a heavy-tailed version of the Gaussian and can be interpreted as a robustification of the measurement noise, see Fig. 2. Moment matching is rather common in nonlinear estimation for enabling analytic expressions, see [35], [36] for recent examples. We justify Approximation 1 in the following proposition.

Proposition 1: As $k \to \infty$, (35) converges to a Gaussian with precision determined by the forgetting factor λ.

Proof 1: Inserting the expression for $\nu_{k|k-1}$ from (20) into $\nu_{k|k}$ in (19), $\nu_{k|k} = \lambda\nu_{k-1|k-1} + 1$. At equilibrium, $\nu_{k|k} = \nu_{k-1|k-1}$ and it follows that $\lim_{k\to\infty} \nu_{k|k} = 1/(1-\lambda)$. The result follows from $\lim_{\nu\to\infty} \mathrm{St}(\boldsymbol{\mu},\boldsymbol{\Lambda},\nu) = \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Lambda})$ and from the sum of Gaussian variables being a Gaussian variable. ■

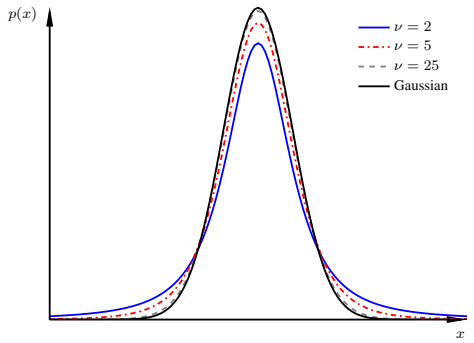

Fig. 2. Illustration of how the Student-t approaches a Gaussian as the degrees of freedom increase.

Note that while Proposition 1 states an asymptotic property of our approximation, the moment matching of the first and second moments holds also during transients. We compute the likelihood (18a) by using Approximation 1 and (31)–(35), and the subsequent measurement update (17) becomes

$$q_k^i \propto q_{k-1}^i \mathrm{St}(\boldsymbol{\mu}^*, \tilde{\boldsymbol{\Lambda}}^*, \tilde{\nu}^*), \quad (36)$$

where

$$\begin{aligned}\boldsymbol{\mu}^* &= \boldsymbol{h}_k + \boldsymbol{b}_{k|k-1} + \boldsymbol{d}_k\hat{\boldsymbol{\mu}}_{k|k-1},\\ \tilde{\boldsymbol{\Lambda}}^* &= \frac{1+\gamma_{k|k-1}}{\tilde{\nu}_{k|k-1}}\boldsymbol{d}_k\boldsymbol{\Lambda}_{k|k-1}\boldsymbol{d}_k^{\mathrm{T}} + \frac{\tilde{\nu}_{k|k-1}-2}{\tilde{\nu}_{k|k-1}}(\boldsymbol{R}+\boldsymbol{P}_{k|k-1}).\end{aligned}$$

Eq. (36) provides the importance weights that are used in the particle filter (15), and for combining the estimates of the bias obtained by (30) for each particle and the parameter density (25), resulting in (28).

Remark 4: According to Proposition 1, Approximation 1 is arbitrarily precise as $k \to \infty$ for $\lambda = 1$. However, introducing exponential forgetting is important for avoiding accumulation of errors. Thus, the particular choice of λ is a tradeoff between the presence of path degeneracy and the accuracy of Approximation 1.[3]

We now turn to the predictive density (18b). Similarly to (31), using Lemma 2 leads to

$$\begin{aligned}p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_{0:k},\boldsymbol{y}_{0:k}) &\propto p(\boldsymbol{g}_k^{-\dagger}(\boldsymbol{x}_{k+1}-\boldsymbol{f}_k)|\boldsymbol{x}_{0:k},\boldsymbol{y}_{0:k})\\ &= p(\boldsymbol{g}_k^{-\dagger}(\boldsymbol{x}_{k+1}-\boldsymbol{f}_k)|\bar{\boldsymbol{e}}_{0:k})\\ &= p(\boldsymbol{w}_k(\boldsymbol{x}_{k+1})|\bar{\boldsymbol{e}}_{0:k}), \quad (37)\end{aligned}$$

where $\boldsymbol{g}_k^{-\dagger}$ is the pseudo-inverse of $\boldsymbol{g}_k$. By integrating over the noise parameters in (18b),

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_{0:k},\boldsymbol{y}_{0:k}) = \int p(\boldsymbol{x}_{k+1}|\boldsymbol{\theta}_k,\boldsymbol{x}_k,\boldsymbol{y}_k) \cdot p(\boldsymbol{\theta}_k|\boldsymbol{x}_{0:k-1},\boldsymbol{y}_{0:k-1})\,\mathrm{d}\boldsymbol{\theta}_k. \quad (38)$$

The integrand in (38) is the product of a Gaussian distribution and a Normal-inverse-Wishart distribution, which is a Student-

[3]Path degeneracy means that the number of unique particle trajectories will decrease as time proceeds.

t distribution. Combining with (37), the predictive distribution of $\boldsymbol{w}_k$ is also a Student-t,

$$p(\boldsymbol{w}_k(\boldsymbol{x}_{k+1})|\bar{\boldsymbol{e}}_{0:k}) = \mathrm{St}(\hat{\boldsymbol{\mu}}_k^*, \tilde{\boldsymbol{\Lambda}}_k^*, \nu_k^*). \tag{39}$$

Using (19), the samples from (39) are used to update (25) (i.e., the second term on the right-hand side of (28)) and (26).

Theorem 1: The hyperparameters in (39) are given by

$$\begin{aligned}
\nu_k^* &= \nu_{k|k-1} - n_e + 1, \\
\boldsymbol{\mu}_k^* &= \hat{\boldsymbol{\mu}}_{k|k-1} + \boldsymbol{d}_k \boldsymbol{\Lambda}_{k|k-1} \tilde{\boldsymbol{\Lambda}}_{\bar{e},k|k-1}^{-1} \boldsymbol{z}_k, \\
\tilde{\boldsymbol{\Lambda}}_k^* &= \frac{\nu_{k|k-1} - d + 1 + \boldsymbol{z}_k \tilde{\boldsymbol{\Lambda}}_{\bar{e},k|k-1}^{-1} \boldsymbol{z}_k^{\mathrm{T}}}{\nu_{k|k-1} - n_e + 1} \Big(\boldsymbol{\Lambda}_{k|k-1} \\
&\quad - \boldsymbol{d}_k \boldsymbol{\Lambda}_{k|k-1} \tilde{\boldsymbol{\Lambda}}_{\bar{e},k|k-1}^{-1} \boldsymbol{\Lambda}_{k|k-1}^{\mathrm{T}} \boldsymbol{d}_k^{\mathrm{T}} \Big), \\
\boldsymbol{z}_k &= \bar{\boldsymbol{e}}_k - \hat{\boldsymbol{\mu}}_{\bar{e},k|k-1}.
\end{aligned} \tag{40}$$

Proof 2: First, given Approximation 1, the joint predictive distribution of $\boldsymbol{w}_k$ and $\bar{\boldsymbol{e}}_k$ is a Student-t. For $\boldsymbol{x}_1 \in \mathbb{R}^{d_1}$ and $\boldsymbol{x}_2 \in \mathbb{R}^{d_2}$ jointly distributed according to a Student-t,

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2) = \mathrm{St}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{12}^{\mathrm{T}} & \boldsymbol{\Lambda}_{22} \end{bmatrix}, \nu \right),$$

and by using the factorization $p(\boldsymbol{x}_1|\boldsymbol{x}_2) = p(\boldsymbol{x}_1, \boldsymbol{x}_2)/p(\boldsymbol{x}_2)$, it can be shown [36] that the conditional density is given by

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) = \mathrm{St}(\boldsymbol{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Lambda}_{1|2}, \nu_{1|2}), \tag{41}$$

where

$$\begin{aligned}
\nu_{1|2} &= \nu + d_2, \\
\boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Lambda}_{12} \boldsymbol{\Lambda}_{22}^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2), \\
\boldsymbol{\Lambda}_{12} &= \frac{\nu + (\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \boldsymbol{\Lambda}_{22}^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2)^{\mathrm{T}}}{\nu + d_2} \\
&\quad \cdot \Big(\boldsymbol{\Lambda}_{11} - \boldsymbol{\Lambda}_{12} \boldsymbol{\Lambda}_{22}^{-1} \boldsymbol{\Lambda}_{12}^{\mathrm{T}} \Big).
\end{aligned}$$

Hence, the hyperparameters (40) follow from (41) by utilizing (35), (36), yielding

$$\begin{aligned}
\nu &= \nu_{k|k-1} - d + 1, \quad d_2 = n_e, \\
\boldsymbol{\mu}_1 &= \boldsymbol{\mu}_{w,k|k-1}, \\
\boldsymbol{\Lambda}_{11} &= \boldsymbol{\Lambda}_{w,k|k-1}, \\
\boldsymbol{\Lambda}_{12} &= \bar{\boldsymbol{g}}_k \boldsymbol{\Lambda}_{w,k|k-1}, \\
\boldsymbol{\Lambda}_{22} &= \tilde{\boldsymbol{\Lambda}}_{\bar{e},k|k-1}, \\
\boldsymbol{x}_2 - \boldsymbol{\mu}_2 &= \bar{\boldsymbol{e}}_k - \hat{\boldsymbol{\mu}}_{\bar{e},k|k-1}.
\end{aligned}$$

■

Using the prediction model as proposal density in the particle filter greatly simplifies the algorithm, since we can draw samples from (39), and use these samples both to create particles $\boldsymbol{x}_{k+1}^i$ according to (9), thereby finding an expression for (18b), and to update the sufficient statistics in (19). Note also that the correlation between process and measurement noise is accounted for by inserting (40) into (39), and that the bias term affects the update of the noise statistics through the error term $\boldsymbol{z}_k$ in (40).

Particle filters can often benefit from an improved proposal distribution $\pi(\cdot)$, but the simplification by using the sampled noise values in both the measurement update of the sufficient statistics and the weight update is particularly appealing in applications with reduced computational capabilities. Furthermore, for non-Gaussian process noise, which is the case here when the mean and variance are unknown, it is nontrivial to find good proposal densities.

Finally, to obtain an approximation of the filtering distribution (14a), we marginalize out $\boldsymbol{x}_{0:k-1}$ from the third term on the right-hand side of (28) (i.e., (15)), which amounts to extracting the last state,

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k}) \approx \sum_{i=1}^{N} q_k^i \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^i). \tag{42}$$

Algorithm 1 summarizes the method for the convenient choice of the prediction model as proposal, and Fig. 3 contains a conceptual flowchart of the algorithm.

Algorithm 1 Pseudo-code of the estimation algorithm

Initialize: Set $\{\boldsymbol{x}_0^i\}_{i=1}^N \sim p_0(\boldsymbol{x}_0)$, $\{q_0^i\}_{i=1}^N = 1/N$, $\{S_0^i\}_{i=1}^N = \{\gamma_0^i, \boldsymbol{\mu}_0^i, \boldsymbol{\Lambda}_0^i, \nu_0^i\}$, $\{\boldsymbol{b}_0^i\}_{i=1}^N \sim p_0(\boldsymbol{b}_0)$
for $k \leftarrow 0$ to T **do**
  for each particle $i \in \{1, \ldots, N\}$ **do**
    Update weight $\bar{q}_k^i$ using (36).
    Update noise statistics $S_{k|k}^i$ using (19).
  end for
  Normalize weights as $q_k^i = \bar{q}_k^i / (\sum_{i=1}^N \bar{q}_k^i)$.
  Compute $N_{\mathrm{eff}} = 1/(\sum_{i=1}^N (q_k^i)^2)$
  if $N_{\mathrm{eff}} \leq N_{\mathrm{thr}}$ **then**
    Resample particles and copy the corresponding statistics. Set $\{q_k^i\}_{i=1}^N = 1/N$.
  end if
  Approximate state filtering posterior with (42).
  Approximate parameter posterior with (26).
  Compute estimates of noise parameters using (27).
  for each particle $i \in \{1, \ldots, N\}$ **do**
    Measurement update of bias using (30).
    Predict noise statistics $S_{k+1|k}^i$ using (20).
    Sample $\boldsymbol{w}_k^i$ from (39).
    Predict state $\boldsymbol{x}_{k+1}^i$ using (9).
    Predict bias using (29).
  end for
end for

Remark 5: Both (42), (26), and similarly (30) overlook a potential path-degeneracy problem, but taking into account different paths leads to an algorithm that is $\mathcal{O}(N^2)$ and therefore often intractable in real-time implementations when computational resources are limited. Furthermore, for sufficient mixing in the dynamic model (9), errors in the state are forgotten as time progresses, which ensures convergence of (42) as $N \to \infty$ [17]. For (26), the use of exponential forgetting suppresses the path-degeneracy problem, which causes issues in the estimation of static parameters [24]. Exponential forgetting includes mixing in the parameter estimation.

V. Aspects of the In-Vehicle Implementation

In actual in-vehicle implementations, the steer angle δ (i.e., the road-wheel angle) is not usually measured. Furthermore,

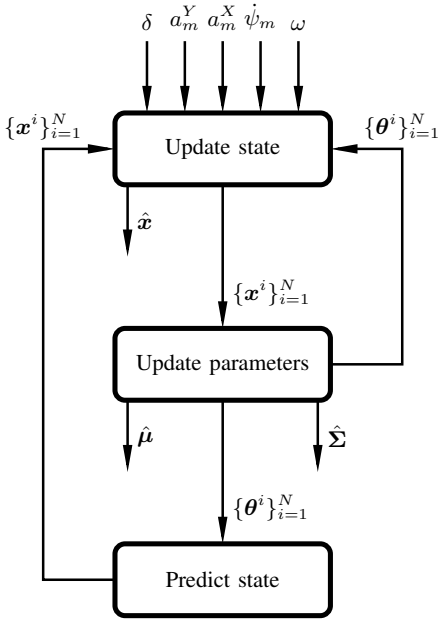

Fig. 3. A simplified flowchart of the proposed method. The bias blocks are not shown because they are byproducts from the estimation procedure. The measurements are compared to the predicted particles and a subsequent updated set of states are generated, together with an actual point estimate of the state. The predicted states are used to update the parameters, from which point estimates of the tire stiffness and associated covariances are computed.

in practice the Ackermann steering configuration causes a slight deviation between the left and right wheel. We assume a single-track model (see Fig. 1), where δ is modeled as the average between the left and right wheel angles. In general, δ can be calculated from a nonlinear map of the steering-wheel angle, which is measured.

The stiffness parameters can be bounded in an interval, which includes the range of stiffness values that can be obtained on different surfaces. This amounts to a parameter projection, which is applied whenever the parameter values are outside the bound. Note that in the experiments and simulations we have undertaken, this projection has never been applied for our approach, but proved valuable when implementing the linear estimators used in this paper for comparison.

In an actual implementation, the algorithm should only be executed when the system is observable. One option is to verify this relying on the discussion in Section II-B, which amounts to thresholding the activation of the algorithm based on the values of the inputs. This is the approach used for obtaining the results in this paper.

In the two sections that follow, we evaluate the algorithm on synthetic data and on a number of real data sets. The real data sets are collected with a mid-size SUV on both snow and dry asphalt, in normal driving. For the simulation results, we compare the method against a state-augmented particle filter (AUGPF) and a method based on linear regression (RLS), similar to that presented in [8]. The state-augmented particle filter models the deviation of the stiffness components as three extra states, in addition to the associated variance of each stiffness component, and is included to show the benefits with our adaptive particle filter where we marginalize out the stiffness parameters, thereby reducing the dimension of the particle filter.

In the experimental evaluation, we also compare with an EKF (denoted by EKF). This comparison is included to show the potential benefits with a particle filter compared to methods based on (local) linear and Gaussian assumptions. We have made extensive tuning of the EKF, but we acknowledge that it is difficult to make fair comparisons with experimental data.

VI. Simulation Study

This section evaluates the proposed method in simulation and compares it with two other methods.

A. Problem Setup

For generating synthetic data, we use a single-track model with steering angle and wheel torque as inputs. The inputs are square waves with period time 4 s and 5 s, respectively, and they are chosen such that the longitudinal wheel slip is below 5% and the small-slip approximations (4) for the wheel-slip angles hold. The tire-stiffness parameters are individually independent and Gaussian distributed with standard deviation approximately 5% of the true values. At the 30 s mark a sudden change in road surface occurs, which results in a 50% decrease in tire stiffness. In the simulations, the mean of the initial vehicle state and the initial covariance are $x_0 = [22\ 0\ 0]^{\mathrm{T}}, P_0 = \mathrm{diag}([1\ 1\ \pi/180]^2)$, which are also used to generate the ground truth. The initial standard deviation of the stiffness estimates is 30% of the true values and the stiffness values are 70% of the true values. The noise values of the measurements are typical for automotive grade inertial sensors. The bias is set to zero for simplicity, as its impact is better tested later, in experiments.

We initialize all the methods with the same values for the variables in common. In AUGPF, the state vector also includes the unknown parameters of the process noise, resulting in a state vector $\boldsymbol{x} \in \mathbb{R}^9$ to be estimated with a particle filter, as opposed to $\boldsymbol{x} \in \mathbb{R}^3$ for our proposed method. Both filters use the weighted mean of the particles as state estimates. In AUGPF, the mean is modeled as a Gaussian random walk, and the inverse-Wishart distribution is used to propagate the unknown variance for each component, $p(\Sigma_{w,k}|\Sigma_{w,k-1}) = \mathrm{iW}(\alpha, \beta)$.[4] The deviation of the random walk is set to 2% of the value of the true parameter, and α, β are set such that the mean value is held constant and the deviation of the distribution is 1% of the previous value. These are the values that result in the smallest root-mean-square error (RMSE) on 100 Monte-Carlo executions.

RLS is based on linear regression, but the combined estimation of longitudinal and lateral dynamics, and of the tire stiffness results in a model with sharp nonlinearities due to the longitudinal velocity. Therefore, we found that a better performance is obtained by first estimating the lateral dynamics by a Kalman filter, and then feeding the obtained estimate to the linear regression, where we also use the true

[4]The inverse-Wishart is the conjugate prior for unknown variance [22].

longitudinal velocity to obtain a more challenging comparison for our proposed approach.

In what follows, all results for the stiffness are presented in normalized scale, with normalization with respect to the true value on asphalt as obtained in bench testing with high-accuracy calibration sensors, to better highlight relative, i.e., percentage, estimation errors.[5] Note that the deviation from the true stiffness values are modeled as a Gaussian disturbance with unknown mean and covariance as in (8). Hence, the mean of the stiffness deviations are computed from (27a) and the estimated stiffness values shown in the following plots are the total stiffness values as defined in (7), and the corresponding confidence intervals are computed from (27b).

B. Results

Fig. 4 shows the results for the front wheel for one Monte-Carlo trial. The results are similar for the rear lateral stiffness. Algorithm 1 and AUGPF use 500 particles. Algorithm 1 shows improvements in the estimation performance. The accuracy of the estimated standard deviation in stationarity for Algorithm 1 in this realization is within 5%. Note that the estimates for AUGPF have larger variability. Indeed, changing the tuning of the noise sources to the artificial dynamics in AUGPF can result in smoother estimates, but also leads to slower convergence when the friction changes. RLS works reasonably well, however, it converges slower and the estimates are biased. Note also that RLS assumes perfect knowledge of the longitudinal dynamics that amounts to an idealized implementation, and this is why the RLS estimate for C_f^x is not shown in Fig. 4. Even if in Fig. 4 AUGPF gives worse performance for the longitudinal stiffness than the lateral stiffness, this is not the case for all the simulations, as for other realizations the longitudinal stiffness is more accurately estimated than the lateral stiffness. However, we did not find a case where both estimates show a performance superior to those generated from Algorithm 1.

It is instructive to see how the performance varies with the number of particles. The variation in the time-averaged RMSE of the lateral velocity is shown in Fig. 5. This plot clearly shows the benefit from modeling the tire stiffness as an unknown disturbance in a marginalized approach. Although the state-augmented particle filter in the limit $N \to \infty$ will achieve a performance similar to Algorithm 1, the number of particles in automotive application is limited by the computational resources. Note that Algorithm 1 and AUGPF have comparable per-particle computation burden and both have complexity $\mathcal{O}(N)$. Due to avoiding introducing artificial dynamics, Algorithm 1 solves an estimation problem of smaller dimension, and hence needs fewer particles to achieve a certain performance metrics, thus allowing for implementation in applications with limited computational resources.

Fig. 6 displays histograms over the estimated mean and standard deviation associated with the front lateral stiffness.

[5]Note to the Reviewers and AE: this is also due to confidentiality, since the tires are product of a third party manufacturer whose data cannot be disclosed publicly.

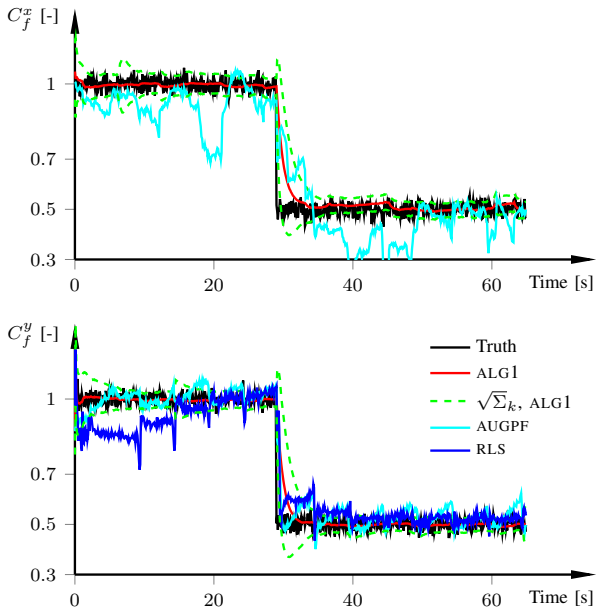

Fig. 4. Estimated normalized front-wheel tire stiffness (red) and associated standard deviation (green) for 500 particles with a forgetting factor of $\lambda = 0.99$, for one realization. Normalized due to confidentiality. The estimated tire stiffness are the sum of the nominal stiffness value and $\boldsymbol{\mu}_k$ (i.e., ΔC_f^y, ΔC_r^y) as computed from (27a).

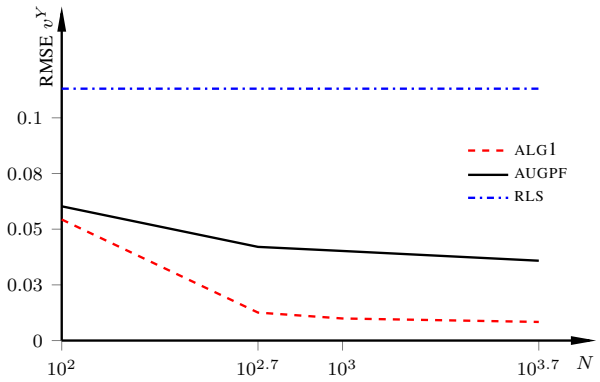

Fig. 5. Time-averaged RMSE of the lateral velocity for different number of particles, averaged over 100 Monte-Carlo executions.

The plots show the probability over 100 Monte-Carlo executions on the y-axis and the error percentage of the true values on the x-axis. The error of the mean value should ideally be Gaussian distributed, since all available information is utilized if the error is Gaussian. For Algorithm 1, the error of the stiffness is centered around zero with the error distribution resembling a Gaussian, as it should be when all available information is utilized correctly. This is clearly not the case for AUGPF. For the standard deviation error, Algorithm 1 gives close to correct estimates, while the estimates from AUGPF are biased. In accordance with the previous discussion, from Figs. 5 and 6 one can infer that AUGPF will need a significantly larger number of particles to achieve performance similar to that of Algorithm 1, which is due to the estimation problem being of higher dimension, and goes against the limitations in computational resources typical of automotive applications.

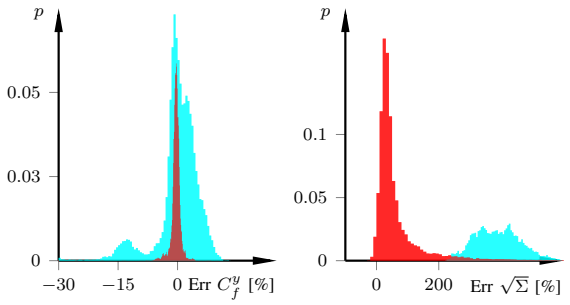

Fig. 6. Histograms of the mean and standard deviation error for the front lateral stiffness, using 500 particles over 100 Monte-Carlo executions. Algorithm 1 in red, AUGPF in cyan, and RLS in blue. Results are similar for the other stiffness quantities. The estimated tire stiffness are the sum of the nominal stiffness value and $\boldsymbol{\mu}_k$ (i.e., ΔC_f^y, ΔC_r^y) as computed from (27a), and the standard deviation is the square root of the diagonal elements of (27).

VII. Experimental Results

We have used a mid-size SUV, equipped with industry-grade validation equipment to gather data, and collected several different data sets, on both snow and asphalt. The parameters of the vehicle model and the tire-stiffness parameters are extracted from data sheets and extensive experimental validation. The true tire-stiffness parameters are determined with high precision in bench testing, albeit, because of sensor noise, calibration uncertainty, and variations due to, for instance, temperature, loading conditions, and wear, there is some remaining uncertainty. Nevertheless, in the evaluation, we treat the bench-determined stiffness as ground truth. Also, note that we use the standard internal sensors, obtained from the CAN bus, of the car for estimation, and only consult the state-of-the-art sensors for validation purposes. In the experimental setup used for collecting data, we have calibrated ground truth of the longitudinal velocity, yaw rate, and the lateral stiffness values. As mentioned in the introduction, and since we did not have access to the ground truth of the longitudinal tire stiffness, here we focus on experimentally validating the estimator performance for the lateral dynamics and stiffness. Preliminary experimental results for the longitudinal stiffness can be found in [26].

First we present the results on snow, and then the results on dry asphalt. We stress that the tuning parameters in the algorithm are the same for all data sets. Thus, we did not tune the algorithm to optimize performance for each individual data set. We use a forgetting factor of $\lambda = 0.997$ and set the measurement noise of the inertial sensors according to the sensor specifications. We have implemented a method based on thresholding (Section V) for activating Algorithm 1 when the conditions that make it observable hold.

A. Validation on Snow

The experimental data on snow have been collected from a number of test drives, each about 40 s long. The data sets consist of a straight-line segment followed by a sinusoidal steering maneuver. The sinusoidal motion takes about 10 s to complete. Due to the random nature of the algorithm, we have executed the algorithm 50 times per data set to get statistically

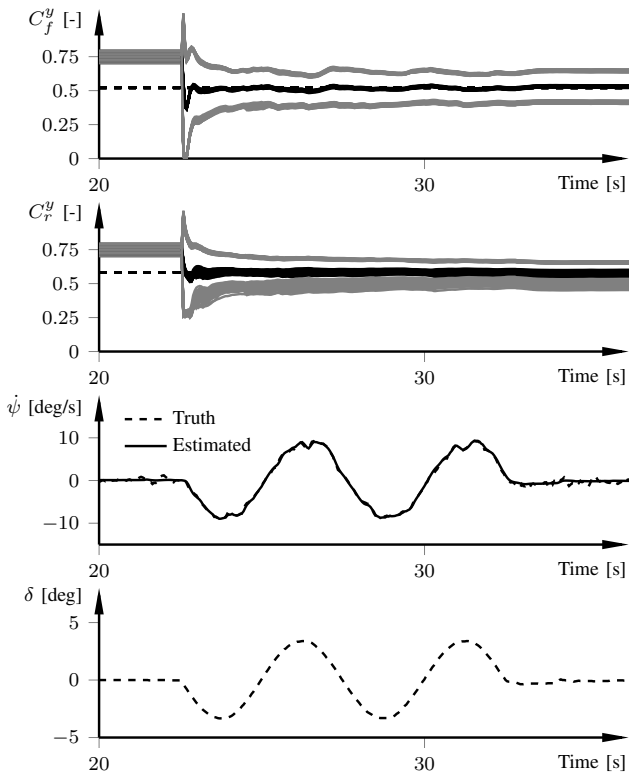

Fig. 7. Estimated normalized, tire stiffness (black) and associated standard deviation (gray) using 500 particles for 50 Monte-Carlo executions on snow. The normalization is with respect to ground truth on asphalt. The estimated tire stiffness and covariance are computed from (27), and the standard deviation is the square root of the diagonal elements of (27b).

meaningful results. The initial estimates of the stiffness are uniformly distributed within the range $150 \pm 10\%$ of the true values.

Fig. 7 summarizes the results for one typical data set. The third plot shows the measured (dashed) and estimated (solid) yaw rate, and the lower-most plot shows the measured average steering angle of the front wheels. Estimation of the lateral components is inactivated during approximately the first 22 s. As soon as turning is initiated, the thresholding activates Algorithm 1, and the correct stiffness values are found within a few seconds. The estimates are similar irrespective of the initial conditions, indicating robustness to the initialization value. The plots show that even with a 50% error in the initial estimate of the stiffness, steady state is reached within a few seconds in all executions. This is another indication of the robustness to large initial condition errors, and that the algorithm, although probabilistic in its nature, delivers similar results across different realizations. The maximum error in steady state is less than 5% across the 50 realizations.

To illustrate the repeatability of the algorithm for different data sets, Fig. 8 displays the average over all 50 Monte-Carlo trials, for each data set. Since the data sets differ slightly in when steering is initiated, the estimator is activated and deactivated at slightly different time steps. However, during the time when the estimator is active, similar steady-state stiffness values are achieved, which is an indication of the robustness to different sensor realizations and steering behavior. In terms of

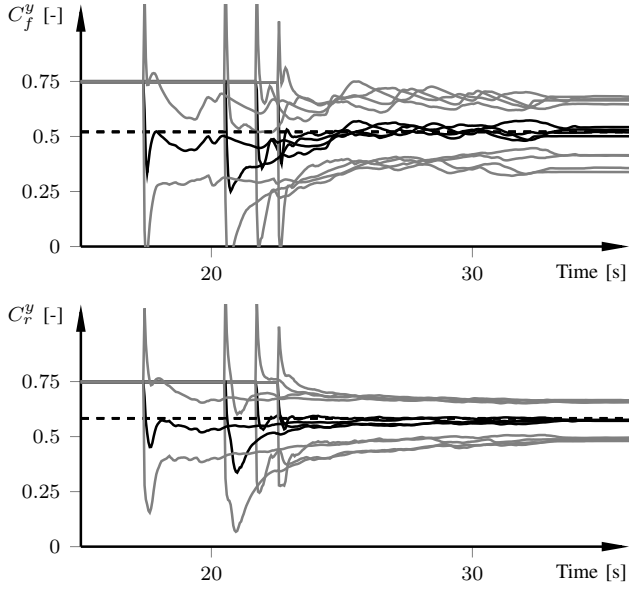

Fig. 8. Estimated normalized tire stiffness (black) and associated standard deviation (gray) using 500 particles, averaged over 50 Monte-Carlo simulations for the data sets collected on snow. The normalization is with respect to ground truth on asphalt. True values in dashed. The estimated tire stiffness are the sum of the nominal stiffness value and $\boldsymbol{\mu}_k$ (i.e., ΔC_f^y, ΔC_r^y) as computed from (27a), and the standard deviation is the square root of the diagonal elements of (27b).

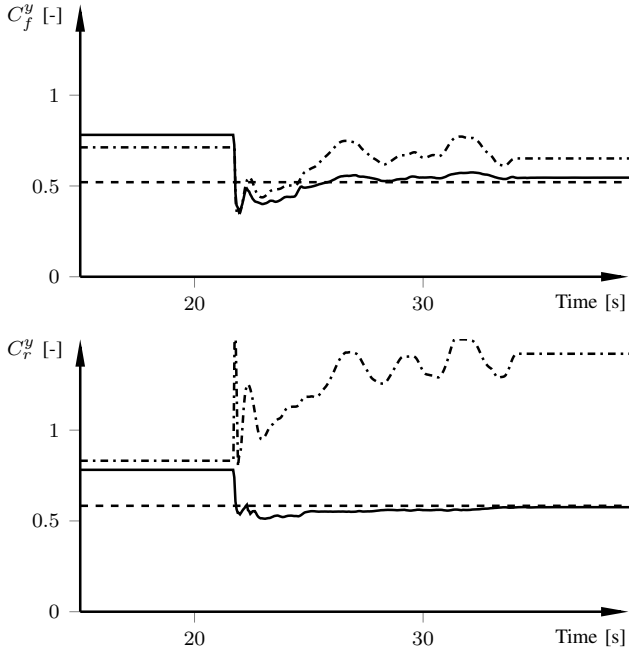

Fig. 9. Estimated normalized tire stiffness using EKF (dash-dotted) for two of the data sets collected on snow. The normalization is with respect to ground truth on asphalt. True values in dashed and our proposed approach in solid black. The estimated tire stiffness are the sum of the nominal stiffness value and $\boldsymbol{\mu}_k$ (i.e., ΔC_f^y, ΔC_r^y) as computed from (27a).

quantitative measures, the maximum error is less than 4% for the front lateral stiffness. The error is less than 1% in steady state for the rear counterpart. The steady-state deviations between the datasets that can be seen are suppressed as the number of Monte-Carlo simulations increase.

1) Comparison with EKF: Fig. 9 shows the resulting estimates for EKF for two of the data sets on snow compared with our proposed approach. We have undertaken extensive tuning efforts for EKF.[6] Even with the best calibration that we have found, EKF is not able to reliably estimate the stiffness when the state trajectory is estimated simultaneously, although it is occasionally successful. To get some insights into why EKF fails to accurately estimate the stiffness, Fig. 10 shows snapshots of the estimated posterior densities using our approach at different time steps. The estimated posterior is at times significantly different from a Gaussian. The underlying assumption of an EKF is that the sufficient statistics of the posterior density are the mean and covariance, while Fig. 10 shows that higher-order moments are in general needed to accurately describe the posterior.

B. Validation on Dry Asphalt

On dry asphalt, the data set consists of normal driving on a regular road and is about 400 s long. We stress that this data set is collected from a period of regular driving on a standard two-lane road and the test was not specifically designed for this experiment. The road requires only light steering, which reduces observability, and it has nonzero inclination and bank angles, which are not explicitly accounted for in the current implementation. Thus, the dataset also tests how robust the algorithm is to these unmodeled effects.

For the first set of results, we set the initial stiffness estimate to be uniformly distributed around $75\pm10\%$ of the true values. The initial distribution is chosen to reflect that the estimator initially does not know much about the surface, so a reasonable guess is to choose something in between snow and asphalt, which is consistent with the chosen uniform distribution. Fig. 11 displays the results for one realization. The first 67 s of the experiment consists of constant-speed driving on a straight road, and because of that the thresholding strategy keeps the estimator inactive. At activation, the uncertainty of the estimate (gray) is initially large but decreases steadily until it approximately reaches steady state. The different excitation levels in the input signals and the unevenness of the surface causes the covariance to fluctuate slightly around the steady state value. Excluding the transients, the largest error at a single time instant is 9% (around 270 s), but the average error is less than 1%. When the largest error occurs there is little excitation in the system, implying that tuning of the activation method can partly remove some of the errors. When comparing with Figs. 7 and 8, the results on asphalt are slightly less smooth. However, this is expected, since the stiffness values, and therefore the generated forces, are considerably larger for the same slip value.

Fig. 12 displays a histogram over the estimated mean associated with the front lateral stiffness. The figure shows the probability (summing to one) over 50 Monte-Carlo trials on the vertical axis and the error percentage on the horizontal axis. The algorithm is activated approximately 67 s into the

[6]Two work days, i.e., 16 hours.

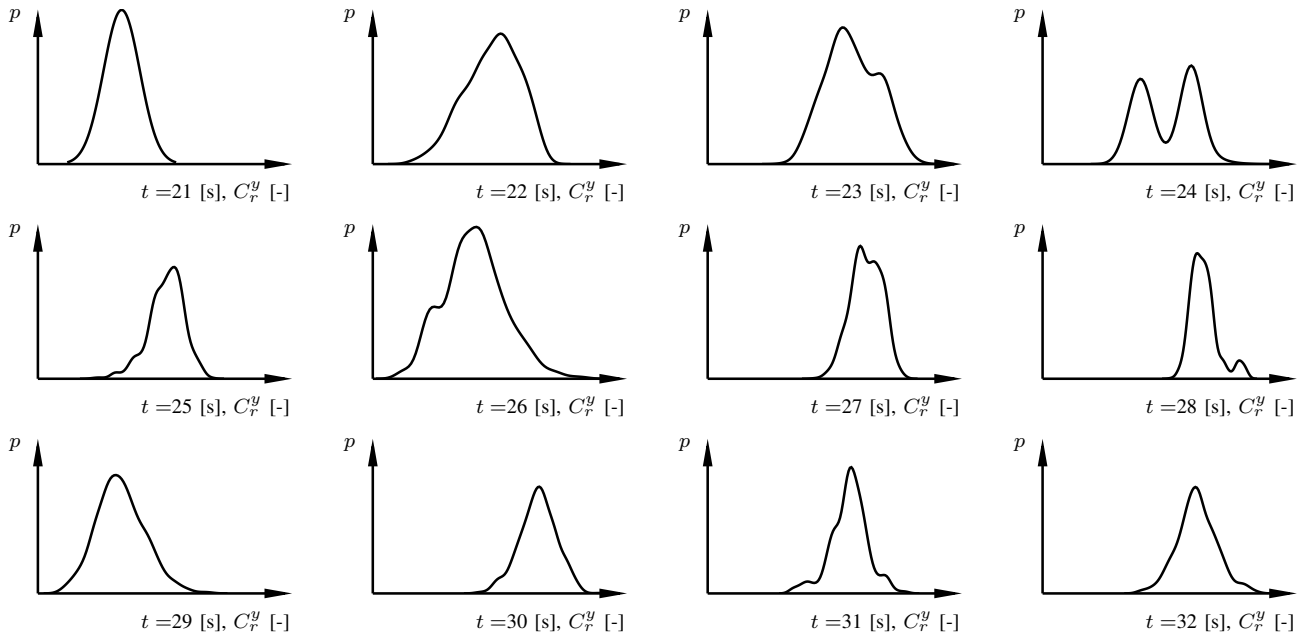

Fig. 10. Snapshots of the posterior density of the rear lateral tire stiffness as estimated using Algorithm 1, between $t = 21$–32 s of the data set. The continuous representation has been obtained from the particles and corresponding weights using a kernel density smoother.

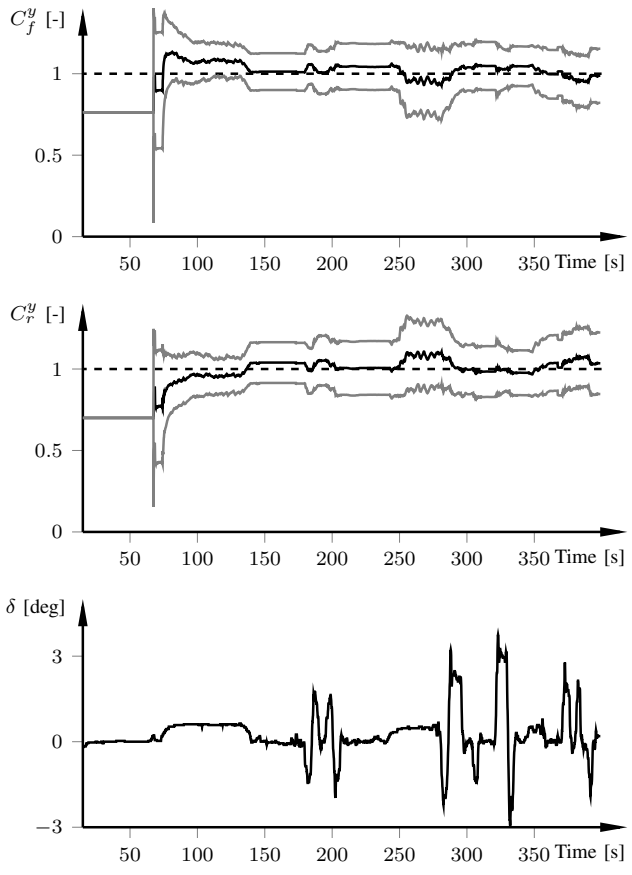

Fig. 11. Upper two plots show the estimated normalized tire stiffness (black) and associated standard deviation (gray) using 500 particles on one realization for the data set collected on dry asphalt. The true values are in dashed. Lowest plot shows the steer angle δ. The estimated tire stiffness are the sum of the nominal stiffness value and $\boldsymbol{\mu}_k$ (i.e., ΔC_f^y, ΔC_r^y) as computed from (27a), and the standard deviation is the square root of the diagonal elements of (27b).

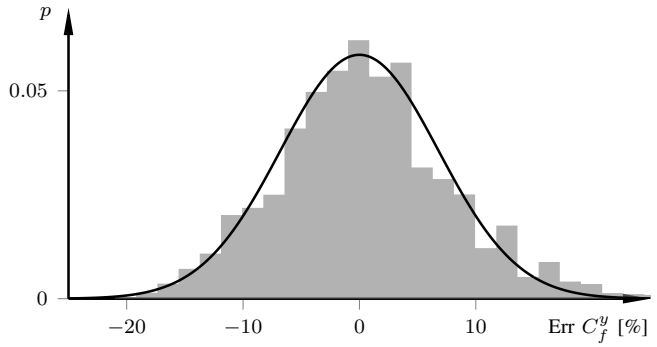

Fig. 12. Histogram (gray) of the mean error for the front lateral stiffness, using 500 particles in 50 Monte-Carlo trials. The overlaid Gaussian (black) has a standard deviation of 6.5%.

experiment (see Fig. 11). To remove most of the transients, only the data points between 75 s and 400 s, for all 50 Monte-Carlo trials, have been included. The mean average error over the trials is less than 1%. As previously mentioned, the error should ideally be Gaussian distributed, since all available information is utilized if the error is Gaussian. Indeed, the error distribution fits well with the overlaid Gaussian (black), which is estimated using a kernel density estimator, verifying estimator performance.

Our method accurately estimates the bias in the inertial sensors. Fig. 13 provides a 15 s excerpt of the experiment (Fig. 11), which shows the true (gray dashed), measured (gray solid), and estimated (black dashed) yaw rate, and the estimated yaw rate compensated with the bias estimates (black solid). The results indicate that the yaw-rate estimates correctly follow those of the high-precision sensor. Furthermore, the bias estimator manages to capture the bias inherent in the yaw-

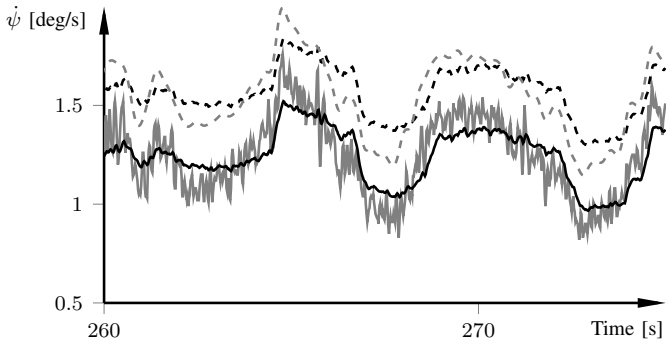

Fig. 13. Measured (gray solid), true (gray dashed), and estimated (black dashed) yaw rate. The sum of the estimated yaw rate and bias is shown in black solid. The bias estimator accurately captures the difference between measured and true yaw rate.

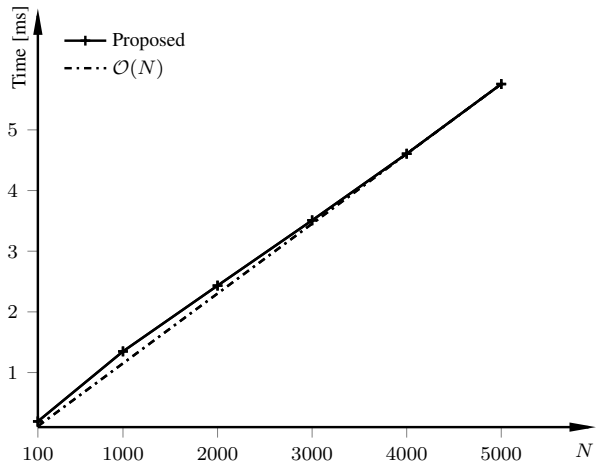

Fig. 14. Average computation time for one iteration of Algorithm 1 for varying number N of particles. The computation time is measured in MATLAB on a 2014 i5 2.8 GHz processor.

rate sensor, despite the relatively moderate lateral motion and the significant noise in the yaw-rate sensor (gray solid).

We conclude the evaluation of our proposed method with an assessment of the computational load shown in Fig. 14, where we plot the average computation time for one estimation iteration of Algorithm 1 for a varying number of particles. The computer is a standard laptop 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 over-estimate 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`. Fig. 14 displays the average computation time per time step, that is, Lines 2–19 in Algorithm 1, for different number of particles. The $\mathcal{O}(N)$ line is also shown to verify 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. Still, judging from Fig. 14 the approach appears real-time feasible, since even considering a platform with 50 times less computing power, which is a reasonable estimate for a mid-level automotive micro-controller, we can execute the algorithm with 500 particles at 20 Hz. Note that by removing the overhead, and by exploiting code and processor-specific optimization, such rate can easily be tripled. It is important to note that the real-time feasibility of the approach is due to the way we structure the estimation problem, and the approximations we use to being able to marginalize out the stiffness parameters to obtain a low-order estimation problem and analytic expressions for the stiffness estimation. While these are indeed approximations, they are motivated by moment matching in the transients and asymptotic convergence, and are the key in achieving real-time feasibility.

VIII. Conclusion

This paper addressed joint tire stiffness and state estimation using wheel-speed and inertial sensors. We defined the problem in a Bayesian framework. The proposed method relies on conjugate priors and moment matching to obtain a computationally efficient marginalized particle filter. The vehicle model and sensor setup leads to dependence between the unknown process noise and the measurement noise. We proposed a real-time viable approach to handle the dependence that was evaluated both in simulation and on several real data sets with promising results. For instance, the results from real data captured on snow shows that the tire stiffness can be estimated within 4% in steady state on regular basis, even for initial uncertainties 50% off from the true values. Furthermore, the results using data for dry asphalt indicate that the estimator provides a mean average error less than 1% in steady state, for driving scenarios corresponding to regular driving.

References

[1] F. Braghin, F. Cheli, and E. Sabbioni, "Environmental effects on Pacejka's scaling factors," *Vehicle System Dynamics*, vol. 44, no. 7, pp. 547–568, Jul. 2006.

[2] J. Svendenius, "Tire modeling and friction estimation," Ph.D. dissertation, Dept. Automatic Control, Lund University, Sweden, Apr. 2007.

[3] C. R. Carlson and J. C. Gerdes, "Consistent nonlinear estimation of longitudinal tire stiffness and effective radius," *IEEE Trans. Contr. Syst. Technol.*, vol. 13, no. 6, pp. 1010–1020, 2005.

[4] S. Di Cairano, H. Tseng, D. Bernardini, and A. Bemporad, "Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain," *IEEE Trans. Contr. Syst. Technol.*, vol. 21, no. 4, pp. 1236–1248, 2013.

[5] F. Gustafsson, "Slip-based tire-road friction estimation," *Automatica*, vol. 33, pp. 1087–1099, 1997.

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

[7] K. Yi, K. Hedrick, and S.-C. Lee, "Estimation of tire-road friction using observer based identifiers," *veh. Syst. Dyn.*, vol. 31, no. 4, pp. 233–261, 1999.

[8] C. Lundquist and T. B. Schön, "Recursive identification of cornering stiffness parameters for an enhanced single track model," in *15th IFAC Symp. System Identification*, Saint-Malo, France, Jul. 2009.

[9] C. Sierra, E. Tseng, A. Jain, and H. Peng, "Cornering stiffness estimation based on vehicle lateral dynamics," *Vehicle System Dynamics*, vol. 44, no. 1, pp. 24–38, 2006.

[10] R. Rajamani, N. Piyabongkarn, J. Lew, K. Yi, and G. Phanomchoeng, "Tire-road friction-coefficient estimation," *IEEE Control Syst. Mag.*, vol. 30, no. 4, pp. 54–69, 2010.

[11] C. Canudas-de Wit and R. Horowitz, "Observers for tire/road contact friction using only wheel angular velocity information," in *38th IEEE Int. Conf. Decision and Control*, Phoenix, AZ, Dec. 1999.

[12] G. Baffet, A. Charara, and D. Lechner, "Estimation of vehicle sideslip, tire force and wheel cornering stiffness," *Control Eng. Pract.*, vol. 17, no. 11, pp. 1255–1264, 2009.
[13] S. Lee, K. Nakano, and M. Ohori, "On-board identification of tyre cornering stiffness using dual Kalman filter and GPS," *Veh. Syst. Dyn.*, vol. 53, no. 4, pp. 437–448, 2015.
[14] R. Wang and J. Wang, "Tire-road friction coefficient and tire cornering stiffness estimation based on longitudinal tire force difference generation," *Control Eng. Pract.*, vol. 21, no. 1, pp. 65–75, 2013.
[15] C. Ahn, H. Peng, and H. E. Tseng, "Robust estimation of road frictional coefficient," *IEEE Trans. Contr. Syst. Technol.*, vol. 21, no. 1, pp. 1–13, 2013.
[16] S. Hong, G. Erdogan, K. Hedrick, and F. Borrelli, "Tyre–road friction coefficient estimation based on tyre sensors and lateral tyre deflection: modelling, simulations and experiments," *Veh. Syst. Dyn.*, vol. 51, no. 5, pp. 627–647, 2013.
[17] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovsky, Eds. Oxford University Press, 2009.
[18] K. Berntorp, "Joint wheel-slip and vehicle-motion estimation based on inertial, GPS, and wheel-speed sensors," *IEEE Trans. Contr. Syst. Technol.*, vol. 24, no. 3, pp. 1020–1027, 2016.
[19] C. Lundquist, R. Karlsson, E. Özkan, and F. Gustafsson, "Tire radii estimation using a marginalized particle filter," *IEEE Trans. Intell. Transport. Syst.*, vol. 15, no. 2, pp. 663–672, 2014.
[20] A. Eidehall, T. B. Schön, and F. Gustafsson, "The marginalized particle filter for automotive tracking applications," in *IEEE Proc. Intelligent Veh. Symp*, Las Vegas, NV, USA, Jun. 2005.
[21] T. B. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear nonlinear state-space models," *IEEE Trans. Signal Processing*, vol. 53, pp. 2279–2289, 2005.
[22] C. M. Bishop, *Pattern Recognition and Machine Learning*. NJ, USA: Springer-Verlag New York, 2006.
[23] S. Saha and F. Gustafsson, "Particle filtering with dependent noise processes," *IEEE Trans. Signal Processing*, vol. 60, no. 9, pp. 4497–4508, 2012.
[24] E. Özkan, V. Šmídl, S. Saha, C. Lundquist, and F. Gustafsson, "Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters," *Automatica*, vol. 49, no. 6, pp. 1566–1575, 2013.
[25] K. Berntorp and S. Di Cairano, "Process-noise adaptive particle filtering with dependent process and measurement noise," in *IEEE Int. Conf. Decision and Control*, Las Vegas, NV, Dec. 2016.
[26] ——, "Tire-stiffness estimation by marginalized adaptive particle filter," in *IEEE Int. Conf. Decision and Control*, Las Vegas, NV, Dec. 2016.
[27] R. Rajamani, *Vehicle Dynamics and Control*. Berlin Heidelberg: Springer-Verlag, 2006.
[28] E. Schindler, *Fahrdynamik: Grundlagen Des Lenkverhaltens Und Ihre Anwendung Für Fahrzeugregelsysteme*. Renningen, Germany: Expert-Verlag, 2007.
[29] H. F. Grip, L. Imsland, T. A. Johansen, J. C. Kalkkuhl, and A. Suissa, "Vehicle sideslip estimation," *IEEE Control Syst. Mag.*, vol. 29, no. 5, pp. 36–52, 2009.
[30] F. Gustafsson, *Statistical Sensor Fusion*. Lund, Sweden: Utbildningshuset/Studentlitteratur, 2010.
[31] K. P. Murphy, "Conjugate Bayesian analysis of the Gaussian distribution," UBC, Tech. Rep., 2007.
[32] V. Peterka, "Bayesian system identification," *Automatica*, vol. 17, no. 1, pp. 41–53, 1981.
[33] C. R. Rao, *Linear Statistical Inference and its Applications*. Wiley, 2001.
[34] C. Berg and C. Vignat, "On the density of the sum of two independent Student t-random vectors," *Statistics & Probability Letters*, vol. 80, no. 13, pp. 1043–1055, 2010.
[35] M. P. Deisenroth, M. F. Huber, and U. D. Hanebeck, "Analytic moment-based Gaussian process filtering," in *Proc. 26th int. conf. machine learning*, Montreal, Canada, 2009.
[36] M. Roth, T. Ardeshiri, E. Özkan, and F. Gustafsson, "Robust Bayesian filtering and smoothing using student's t distribution," *ArXiv e-prints*, Mar. 2017.

Karl Berntorp received the M.Sc. degree in Engineering Physics in 2009 and the Ph.D. degree in Automatic Control in 2014, from Lund University, Lund, Sweden. In 2014 he became a research scientist at the Mitsubishi Electric Research Laboratories in Cambridge, MA. His research is on statistical signal processing, sensor fusion, and optimization-based control, with applications to automotive, aerospace, transportation systems, and communication systems. His work includes design and implementation of nonlinear filtering, constrained control, and motion-planning algorithms. Dr. Berntorp is the author of more than 35 peer-reviewed international papers and several patents, and he is a founder/co-founder of two engineering consultancy companies.

(SM'08) Stefano Di Cairano received the Master (Laurea), and the PhD in Information Engineering in '04 and '08, respectively, from the University of Siena, Italy. He has been visiting student at the Technical University of Denmark and at the California Institute of Technology. During 20082011, he was with Powertrain Control R&A, Ford Research and Adv. Engineering, Dearborn, MI. Since 2011, he is with Mitsubishi Electric Research Laboratories, Cambridge, MA, where he is now the Senior Team Leader for Optimization-based Control, and a Senior Principal Researcher in Mechatronics. His research is on optimization-based control strategies for complex mechatronic systems, in automotive, factory automation, transportation systems and aerospace. His research interests include model predictive control, constrained control, networked control systems, hybrid systems, optimization.

Dr. Di Cairano has authored/co-authored more than 130 peer reviewed papers in journals and conference proceedings and 25 patents. He was the Chair of the IEEE CSS Technical Committee on Automotive Controls 2012-2015, is the Chair of IEEE Standing Committee on Standards since 2016, and an Associate Editor of the IEEE Transactions on Control Systems Technology.