Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Online Bayesian Tire-Friction Learning by Gaussian-Process State-Space Models

Karl Berntorp *

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

Abstract: The friction dependence between tire and road is highly nonlinear and varies heavily between different surfaces. The tire friction is important for real-time vehicle control, but difficult to learn with automotive-grade sensors as they only provide indirect measurements based on sensing parts of the vehicle state. In this paper we leverage recent advances in particle filtering and Gaussian Processes (GPs), to provide an online method for jointly estimating the vehicle state and subsequently identifying the tire friction as a function of the wheel slip. The unknown function mapping the wheel slip to tire friction is modeled as a GP that is included in a dynamic vehicle model relating the GP to the vehicle state.We illustrate the efficacy of the method using synthetic data on a snow-covered road.

Keywords: Automotive system identification and modelling; Learning and adaptation in autonomous vehicles; Particle filter; Friction estimation; intelligent vehicles

1. INTRODUCTION

As there is an increased demand on technologies for enabling autonomous driving (AD), the sensing and estimation algorithms are required to produce an increasing amount of information about the vehicle and its interaction with the environment to support the control algorithms. For advanced driver-assistance systems (ADAS), such as vehicle steering controllers, the main actuation is done through the interaction between tire and road. Hence, to further increase the AD capabilities in vehicles, it is important to have reliable knowledge of the tire friction.

Various tire models describing the tire friction as a function of wheel slip (i.e., the tire-friction function) have been reported in literature—for example, the Magic formula (Pacejka, 2006), the Burckhardt model (Kiencke and Nielsen, 2005), and the Brush model (Svendenius, 2007). The parametrizations used vary across the different models, but the main characteristics are similar. Knowledge of the tire friction over a range of slip values extending into the saturated region of the tire-friction function is important for AD and ADAS, since the prediction models used in several of the recently proposed control methods rely on such knowledge (Carvalho et al., 2015; Frasch et al., 2013; Quirynen et al., 2018).

A difficulty when addressing the tire-friction estimation problem using automotive-grade sensors is that the amount of sensors is limited, and they are relatively low grade (Gustafsson, 2009). Moreover, not only do the sensors only provide indirect measurements of the friction, they do not even measure the vehicle state, which is nonlinearly dependent on the tire friction and must therefore be known for learning the tire friction. Also, it is worth pointing out that few approaches so far target the estimation of the full tire-friction function only using production-grade sensors.

In this paper, we develop a method for jointly estimating the tire-friction *function* and the vehicle state only using sensors available in production cars, namely wheel-speed sensors and inexpensive accelerometers and gyroscopes. While our primary focus is the lateral dynamics, the method developed here can be applied to either lateral or longitudinal dynamics, or to the two combined. Our approach is fully Bayesian and models the deviations from a nominal tire-friction curve as a Gaussian process (GP) with unknown and time-varying mean and covariance function (Rasmussen and Williams, 2006), leading to a GP state-space model (GP-SSM). GPs (Rasmussen and Williams, 2006) are effective tools for nonparametric modeling of static nonlinear functions and GPs have recently been extended to modeling dynamical system behavior (Frigola et al., 2014; Svensson et al., 2016). In particular, we leverage a recently proposed method for real-time joint state estimation and learning of the state-transition function (Berntorp, 2019a), where a reduced-rank formulation (Svensson et al., 2016) of GP-SSMs is combined with particle filtering (Doucet and Johansen, 2009) for jointly estimating online the state and associated state-transition function. Due to the nonparametric nature of the GP, the method is not subject to specific modeling constraints that various tire models impose. Still, the method is insensitive to overfitting to the data.

The tire-friction identification approaches in literature typically estimate parameters of specific models using different techniques. Batch methods for identifying the parameters of the Brush model based on nonlinear optimization can be found in (Svendenius, 2007). The method in (Goldfain et al., 2019) uses an unscented Kalman filter (UKF) approach that augments the vehicle state and models the Pacejka parameters as random walk processes, which are subsequently estimated in a recursive fashion. The works (Lundquist and Schön, 2009; Lee et al., 2015) employ recursive least-squares for estimating the cornering stiffness (the linear slope of the friction curve), and (Garatti and Bittanti, 2009) performs estimation of the Pacejka tire parameters by generating artificial data associated with different tire parameters and solving for the best fit to measured data. In (Ahn et al., 2012), the Brush tire model and a nonlinear observer is used to estimate the peak friction coefficient under different excitation levels, and in (Berntorp, 2019b) we developed an *offline* method by using the particle Markov-chain Monte-Carlo framework (Svensson and Schön, 2017).

Notation: For a discrete-time signal x with sampling period T_s, $x_k = x(t_k) = x(kT_s)$. 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 x_n denotes the nth component of $\boldsymbol{x}$. Matrices are indicated in capital bold font as $\boldsymbol{X}$, and the element on row i and column j is denoted with X_{ij}. 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 step 0 to time step k given the measurement sequence $\boldsymbol{y}_{0:k} := \{\boldsymbol{y}_0, \dots, \boldsymbol{y}_k\}$, and $\boldsymbol{x}^i_{0:k}$ is the ith realization of $\boldsymbol{x}_{0:k}$. The notation $\boldsymbol{f} \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{\kappa}_{\theta,f}(\boldsymbol{x}, \boldsymbol{x}'))$ means that the function $\boldsymbol{f}(\boldsymbol{x})$ is a realization from a GP prior with a given covariance function $\boldsymbol{\kappa}_{\theta,f}(\boldsymbol{x}, \boldsymbol{x}')$ subject to hyperparameters $\boldsymbol{\theta}$, and $\mathcal{IW}(\nu, \boldsymbol{\Lambda})$ is the inverse-Wishart distribution with degree of freedom ν and scale matrix $\boldsymbol{\Lambda}$. Similarly, $\mathcal{MN}(\boldsymbol{M}, \boldsymbol{Q}, \boldsymbol{V})$ and $\mathcal{T}(\boldsymbol{M}, \boldsymbol{Q}, \boldsymbol{V})$ are the Matrix-Normal and Matrix-t distribution, respectively, with mean $\boldsymbol{M}$, covariance (scale) $\boldsymbol{Q}$, and inverse covariance (scale) $\boldsymbol{V}$.

2. MODELING AND PROBLEM FORMULATION

In this section we summarize the different models used by the proposed learning method.

2.1 Chassis Modeling

We use a single-track chassis model that includes the lateral velocity v^Y and the yaw rate $\dot{\psi}$. Consequently, in this paper we focus on the lateral dynamics, but the method extends straightforwardly to also handle longitudinal dynamics. The state vector is $\boldsymbol{x} = [v^Y \ \dot{\psi}]^{\mathrm{T}} \in \mathbb{R}^2$. A single-track model is sufficiently accurate for purposes where the tire friction reaches the nonlinear region but the maneuvers are not aggressive enough to result in large roll angles (Quirynen et al., 2018). The presented framework can be extended to handle a double-track model, but it increases computation time and modeling complexity.

The single-track mode lumps together the left and right wheel on each axle, and roll and pitch dynamics are neglected. Thus, the model has two translational and one rotational degrees of freedom. The model dynamics are given by

$$\dot{v}^Y + v^X\dot{\psi} = \frac{1}{m}(F^z_f\mu^y_f\cos(\delta) + F^z_r\mu^y_r + F^z_f\mu^x_f\sin(\delta)), \tag{1a}$$

$$I_{zz}\ddot{\psi} = l_fF^z_f\mu^y_f\cos(\delta) - l_rF^z_r\mu^y_r + l_fF^z_f\mu^x_f\sin(\delta), \tag{1b}$$

where μ^y is the lateral tire friction function and the subscripts f, r stand for front and rear, respectively, m is the vehicle mass, I_{zz} is the vehicle inertia about the vertical axis, and δ is the front-wheel steering angle. By denoting the wheel base with $l = l_f + l_r$, the normal force F^z resting on each front/rear wheel is

$$F^z_f = mg\frac{l_r}{l}, \quad F^z_r = mg\frac{l_f}{l}. \tag{2}$$

2.2 Tire Modeling

The tire friction components μ^y_i, $i \in \{f, r\}$ are modeled as static functions of the slip quantities,

$$\mu^y_i = f^y_i(\alpha_i(\boldsymbol{x})), \tag{3}$$

α is the slip angle, which is defined as (Pacejka, 2006),

$$\alpha_i = -\arctan\left(\frac{v_{y,i}}{v_{x,i}}\right), \tag{4}$$

where $v_{x,i}$ and $v_{y,i}$ are the longitudinal and lateral wheel velocities for wheel i with respect to an inertial system, expressed in the coordinate system of the wheel. The wheel velocities can be computed from a transformation of the longitudinal and lateral vehicle velocities. The lateral velocity is estimated in the proposed method, whereas the longitudinal velocity v^X is determined from the measured wheel-speeds $\{\omega_i\}^4_{i=1}$. For brevity, we define the vector $\boldsymbol{\alpha} = [\alpha_f \ \alpha_r]^{\mathrm{T}}$. We write (3) as

$$\boldsymbol{\mu} = \left[f^y_f \ f^y_r\right]^{\mathrm{T}}, \tag{5}$$

and model the friction vector as a realization from a zero-mean GP prior

$$\boldsymbol{\mu}(\boldsymbol{\alpha}) \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{\kappa}_{\theta,\mu}(\boldsymbol{\alpha}, \boldsymbol{\alpha}')), \tag{6}$$

where the covariance function $\boldsymbol{\kappa}_{\theta,\mu}(\boldsymbol{\alpha}, \boldsymbol{\alpha}')$ is chosen in advance. In this work the hyperparameters $\boldsymbol{\theta}$ are determined a priori and we refer inclusion of the hyperparameters into the learning process as future work.

2.3 Estimation Model

After discretization with sampling period T_s and using $\boldsymbol{u} = [\delta \ v^X]^{\mathrm{T}}$ as the known input vector, the vehicle model (1)–(6) can be written as

$$\boldsymbol{x}_{k+1} = \boldsymbol{a}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{G}(\boldsymbol{x}_k, \boldsymbol{u}_k)\boldsymbol{\mu}(\boldsymbol{\alpha}_k), \tag{7}$$

where $\boldsymbol{a}(\cdot)$ and $\boldsymbol{G}(\cdot)$ are the (known) parts of the vehicle model, and $\boldsymbol{\mu}(\cdot)$ is the unknown tire-friction function.

Our measurement model is based on a setup commonly available in production cars, namely the lateral acceleration a^Y_m and the yaw rate $\dot{\psi}_m$, forming the measurement vector $\boldsymbol{y} = [a^Y_m \ \dot{\psi}_m]^{\mathrm{T}}$. To relate $\boldsymbol{y}_k$ to the vehicle state $\boldsymbol{x}_k$ at each time step k, note that a^Y can be extracted from the right-hand sides of (1a), after dividing the vehicle mass and shifting over the first terms on the right-hand sides. The measurement $\dot{\psi}_m$ is a direct measurement of the yaw rate. We model the measurement noise $\boldsymbol{e}_k$ as zero-mean Gaussian distributed noise with covariance $\boldsymbol{R}$ according to $\boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$. The measurement model is written as

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{D}(\boldsymbol{x}_k, \boldsymbol{u}_k)\boldsymbol{\mu}(\boldsymbol{\alpha}_k) + \boldsymbol{e}_k. \tag{8}$$

The measurement model (8) is decomposed into known parts of the dynamics, $\boldsymbol{h}(\cdot)$ and $\boldsymbol{D}(\cdot)$, and an unknown part, $\boldsymbol{\mu}(\cdot)$. The measurement covariance $\boldsymbol{R}$ is assumed known a priori. This is reasonable, since the measurement noise can oftentimes be determined from prior experiments and data sheets. In addition, when evaluating (8) we assume a prior knowledge of the friction, for example, using the methods in (Berntorp and Di Cairano, 2018; Berntorp and Hiroaki, 2019). This helps in reaching faster convergence and also approximately decouples the motion model and measurement model, and has been proven to give adequate performance in system identification (Svensson and Schön, 2017).

The estimation model consisting of (7) and (8) is a GP-SSM where the tire friction is a GP. We rely on GP priors for learning the tire friction function, where the covariance function $\boldsymbol{\kappa}(\boldsymbol{x}, \boldsymbol{x}')$ encodes the prior assumptions. A bottleneck in some of the proposed GP-SSM methods is the computational load. In this paper we use the computationally efficient reduced-rank GP-SSM framework presented in (Solin and Särkkä, 2014; Svensson and Schön, 2017). For a thorough derivation and convergence proofs, see (Solin and Särkkä, 2014). Following the notation in (Solin and Särkkä, 2014), isotropic covariance functions (that only depend on the Euclidean norm $\|\boldsymbol{x} - \boldsymbol{x}'\|$) can be approximated in terms of Laplace operators on the form:

$$\boldsymbol{\kappa}_\theta(\boldsymbol{x}, \boldsymbol{x}') \approx \sum_{j_1,\ldots,j_d=1}^{m} \mathcal{S}_\theta(\lambda^{j_1,\ldots,j_d})\phi^{j_1,\ldots,j_d}(\boldsymbol{x})\phi^{j_1,\ldots,j_d}(\boldsymbol{x}'), \tag{9}$$

where we for simplicity assume m basis functions for each state dimension. In (9), $\mathcal{S}_\theta$ is the spectral density of $\boldsymbol{\kappa}_\theta$ and

$$\lambda^{j_1,\ldots,j_d} = \sum_{n=1}^{d}\left(\frac{\pi j_n}{2L_n}\right)^2, \tag{10a}$$

$$\phi^{j_1,\ldots,j_d} = \prod_{n=1}^{d}\frac{1}{\sqrt{L_n}}\sin\left(\frac{\pi j_n(x_n + L_n)}{2L_n}\right), \tag{10b}$$

are the Laplace operator eigenvalues and eigenfunctions, respectively, defined on the intervals $[-L_n, L_n]$. For brevity, we denote $j_1, \ldots, j_d$ with $\boldsymbol{j}$. Note that according to (9), (10), only the spectral density depends on the hyperparameters $\boldsymbol{\theta}$. Furthermore, (9) can be interpreted as an optimal parametric expansion with respect to the covariance function in the GP prior (Svensson and Schön, 2017).

From the approximation (9) using Laplace operators, (Solin and Särkkä, 2014) provides a relation between basis function expansions of a function f and GPs based on the Karhunen-Loeve expansion. Namely, with the basis functions chosen as (10b),

$$f(\boldsymbol{x}) \sim \mathcal{GP}(0, \kappa(\boldsymbol{x}, \boldsymbol{x}')) \Leftrightarrow f(\boldsymbol{x}) \approx \sum_{\boldsymbol{j}} \gamma^{\boldsymbol{j}}\phi^{\boldsymbol{j}}(\boldsymbol{x}), \tag{11}$$

where

$$\gamma^{\boldsymbol{j}} \sim \mathcal{N}(0, S(\lambda^{\boldsymbol{j}}). \tag{12}$$

For a state-space model $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{w}_k$, (11) implies the reduced-rank GP-SSM

$$\boldsymbol{x}_{k+1} = \begin{bmatrix}\gamma_1^1 & \cdots & \gamma_1^m \\ \vdots & & \vdots \\ \gamma_d^1 & \cdots & \gamma_d^m\end{bmatrix}\begin{bmatrix}\phi^1(\boldsymbol{x}_k) \\ \vdots \\ \phi^m(\boldsymbol{x}_k)\end{bmatrix} + \boldsymbol{w}_k, \tag{13}$$

where γ_n^j are the weights to be learned, m is the total number of basis functions (i.e., m^d in (9)), and $\boldsymbol{w}_k$ is zero-mean Gaussian distributed noise with covariance $\boldsymbol{Q}$ according to $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$. In Sec. 3, (13) in combination with particle filtering forms the basis for our joint state estimation and tire-friction learning method.

To get our vehicle model (7) on the form (13), note that by manipulation of (7) and using the basis function expansion approach (11), the model can be written on the form (Berntorp, 2019b)

$$\boldsymbol{\zeta}_{k+1} = \underbrace{\begin{bmatrix}\gamma_1^1 & \cdots & \gamma_1^m \\ \vdots & & \vdots \\ \gamma_d^1 & \cdots & \gamma_d^m\end{bmatrix}}_{\boldsymbol{A}}\underbrace{\begin{bmatrix}\phi^1(\boldsymbol{\alpha}_k) \\ \vdots \\ \phi^m(\boldsymbol{\alpha}_k)\end{bmatrix}}_{\boldsymbol{\varphi}(\boldsymbol{\alpha}_k)} + \boldsymbol{w}_k \tag{14}$$

for some $\boldsymbol{\zeta}_k = [\zeta_{1,k}\ \zeta_{2,k}]^{\mathrm{T}}$. Hence, the original problem of learning the friction function $\boldsymbol{\mu}(\cdot)$ has been transformed to learning $\boldsymbol{A}$ in (14). The noise term $\boldsymbol{w}_k$ accounts for modeling errors and allows for more flexibility in the model and is incorporated into the learning process.

2.4 Problem Formulation

We want to estimate both the nonlinear function $\boldsymbol{\mu}(\boldsymbol{\alpha}_k) \approx \boldsymbol{A}\boldsymbol{\varphi}(\boldsymbol{\alpha}_k)$ describing the tire friction and the vehicle state $\boldsymbol{x}_k$ online at each time step k. We approach this problem as follows. Given the system model (7), (8), and a GP prior (6) on the tire friction resulting in the GP-SSM (14), we want to infer the posterior distribution of $\boldsymbol{x}_k$ and $\boldsymbol{A}$ given a set of measurement data $\boldsymbol{y}_{0:k}$,

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k}), \tag{15a}$$

$$p(\boldsymbol{A}|\boldsymbol{y}_{0:k}). \tag{15b}$$

We also include $\boldsymbol{Q}$ in the learning process, and since the tire-friction estimate will depend on the vehicle state, and vice versa, we solve for (15) by approximating the joint posterior

$$p(\boldsymbol{A}, \boldsymbol{Q}, \boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k}) \tag{16}$$

at each time step k, from which we can extract (15).

3. JOINT VEHICLE STATE AND FRICTION-FUNCTION LEARNING

The objective is to estimate the posterior distribution (15) of the unknown function $\boldsymbol{\mu}(\boldsymbol{\alpha}_k) \approx \boldsymbol{A}\boldsymbol{\varphi}(\boldsymbol{\alpha}_k)$ and the vehicle state $\boldsymbol{x}_k$ at each time step k. To approximate (15), we decompose (16) into conditional densities as

$$p(\boldsymbol{A}, \boldsymbol{Q}, \boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k}) = p(\boldsymbol{A}, \boldsymbol{Q}|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{0:k})p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k}). \tag{17}$$

In what follows, we describe how to recursively approximate the two densities on the right-hand side of (17).

3.1 State Estimation with Particle Filtering

We approximate the posterior of the state trajectory $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k})$ in (17) by a set of N weighted state trajectories as

$$p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k}) \approx \sum_{i=1}^{N} q_k^i \delta_{\boldsymbol{x}_{0:k}^i}(\boldsymbol{x}_{0:k}), \tag{18}$$

where q_k^i is the importance weight of the ith state trajectory $\boldsymbol{x}_{0:k}^i$ and $\delta(\cdot)$ is the Dirac delta mass. The particle filter recursively estimates (18) by repeated application of Bayes' rule as

$$p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k}) \propto p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{0:k-1})p(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{0:k-1}) \cdot p(\boldsymbol{x}_{0:k-1}|\boldsymbol{y}_{0:k-1}). \tag{19}$$

In general, the particles are sampled from a user-designed *proposal distribution* $\pi(\cdot)$ as

$$\boldsymbol{x}_k \sim \pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{0:k}). \tag{20}$$

Inserting (18) into (19) and accounting for the proposal, importance weight q_k^i is obtained as

$$q_k^i \propto q_{k-1}^i \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k^i)p(\boldsymbol{x}_k^i|\boldsymbol{x}_{0:k-1}^i, \boldsymbol{y}_{0:k-1})}{\pi(\boldsymbol{x}_k^i|\boldsymbol{x}_{0:k-1}^i, \boldsymbol{y}_{0:k})}. \tag{21}$$

In this work we choose the proposal as the predictive density, $\pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{0:k}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{0:k-1})$, which leads to the simplified weight update

$$q_k^i \propto q_{k-1}^i p(\boldsymbol{y}_k|\boldsymbol{x}_k^i). \tag{22}$$

The particle filter iterates between prediction and weight update, combined with a resampling step that removes particles with low weights and replaces them with more likely particles.

3.2 Bayesian Learning of the Tire-Friction Function

The distribution of $\boldsymbol{A}$ and $\boldsymbol{Q}$ in (17) is computed conditioned on the realization of the state and measurement trajectories. For a realization $\boldsymbol{x}_{0:k}^i$, the posterior density of $\boldsymbol{A}$ and $\boldsymbol{Q}$ can be written into a likelihood and prior according to Bayes' rule,

$$p(\boldsymbol{A}, \boldsymbol{Q}|\boldsymbol{x}_{0:k}^i) \propto p(\boldsymbol{x}_k^i|\boldsymbol{x}_{0:k-1}^i, \boldsymbol{A}, \boldsymbol{Q})p(\boldsymbol{A}, \boldsymbol{Q}|\boldsymbol{x}_{0:k-1}^i) \tag{23}$$

The first term on the right-hand side of (23) is assumed Gaussian distributed and the joint prior (i.e., the second term on the right-hand side in (23)) is assumed $\mathcal{MNIW}$ distributed with the hierarchical structure

$$\mathcal{MNIW}(\boldsymbol{A}, \boldsymbol{Q}|\boldsymbol{0}, \boldsymbol{V}, \boldsymbol{\Lambda}_0, \nu_0) = \mathcal{MN}(\boldsymbol{A}|\boldsymbol{0}, \boldsymbol{Q}, \boldsymbol{V})\mathcal{IW}(\boldsymbol{Q}|\nu_0, \boldsymbol{\Lambda}_0). \tag{24}$$

For a Gaussian joint likelihood, the joint prior (24) is a conjugate prior (Dawid, 1981; Murphy, 2007; Svensson and Schön, 2017). Hence, for any $k = 1, 2, \ldots$, the posterior (23) is $\mathcal{MNIW}$ distributed when conditioning on a realization $\boldsymbol{x}_{0:k}^i$ and $\boldsymbol{y}_{0:k}$. In (24), $\boldsymbol{V}$ is a diagonal matrix with entries $S(\lambda^{\boldsymbol{j}})$, that is, $\boldsymbol{V}$ encodes prior knowledge of the weights (12).

Furthermore, the posterior distributions of $\boldsymbol{A}$ and $\boldsymbol{Q}$ for a model on the form (13) conditioned on trajectories $\boldsymbol{x}_{0:T}$ and $\boldsymbol{y}_{0:T}$ are (Svensson et al., 2016; Svensson and Schön, 2017)

$$p(\boldsymbol{Q}|\boldsymbol{x}_{0:T}^i, \boldsymbol{y}_{0:T}) = \mathcal{IW}(\boldsymbol{Q}|T + \nu_0, \boldsymbol{\Lambda}_T), \tag{25a}$$

$$p(\boldsymbol{A}|\boldsymbol{Q}, \boldsymbol{x}_{0:T}, \boldsymbol{y}_{0:T}) = \mathcal{MN}(\boldsymbol{A}|\boldsymbol{M}_T, \boldsymbol{Q}, (\boldsymbol{\Sigma}_T + \boldsymbol{V})^{-1}), \tag{25b}$$

where $\boldsymbol{M}_T = \boldsymbol{\Psi}_T(\boldsymbol{\Sigma}_T + \boldsymbol{V})^{-1}$, $\boldsymbol{\Lambda}_T = \boldsymbol{\Lambda}_0 + \boldsymbol{\Phi}_T - \boldsymbol{M}_T\boldsymbol{\Psi}_T^{\mathrm{T}}$,

$$\boldsymbol{\Phi}_T = \sum_{k=1}^{T-1} \boldsymbol{x}_{k+1}\boldsymbol{x}_{k+1}^{\mathrm{T}}, \tag{26a}$$

$$\boldsymbol{\Psi}_T = \sum_{k=1}^{T-1} \boldsymbol{x}_{k+1}\boldsymbol{\varphi}(\boldsymbol{x}_k)^{\mathrm{T}}, \tag{26b}$$

$$\boldsymbol{\Sigma}_T = \sum_{k=1}^{T-1} \boldsymbol{\varphi}(\boldsymbol{x}_k)\boldsymbol{\varphi}(\boldsymbol{x}_k)^{\mathrm{T}}. \tag{26c}$$

To get recursive update equations suitable for online learning using the model (14), we note that from (26), we can write

$$\boldsymbol{\Phi}_{k+1} = \boldsymbol{\Phi}_k + \boldsymbol{\zeta}_{k+1}\boldsymbol{\zeta}_{k+1}^{\mathrm{T}}, \tag{27a}$$

$$\boldsymbol{\Psi}_{k+1} = \boldsymbol{\Psi}_k + \boldsymbol{\zeta}_{k+1}\boldsymbol{\varphi}(\boldsymbol{\alpha}_k)^{\mathrm{T}}, \tag{27b}$$

$$\boldsymbol{\Sigma}_{k+1} = \boldsymbol{\Sigma}_k + \boldsymbol{\varphi}(\boldsymbol{\alpha}_k)\boldsymbol{\varphi}(\boldsymbol{\alpha}_k)^{\mathrm{T}}. \tag{27c}$$

Hence, the statistics necessary to determine (25) for the model (14) can be recursively updated as measurements arrive as

$$\boldsymbol{M}_{k|k} = \boldsymbol{\Psi}_{k|k}(\boldsymbol{\Sigma}_{k|k} + \boldsymbol{V})^{-1}, \tag{28a}$$

$$\boldsymbol{\Sigma}_{k|k} = \boldsymbol{\Sigma}_{k|k-1} + \boldsymbol{\varphi}(\boldsymbol{\alpha}_{k-1})\boldsymbol{\varphi}(\boldsymbol{\alpha}_{k-1})^{\mathrm{T}}, \tag{28b}$$

$$\boldsymbol{\Phi}_{k|k} = \boldsymbol{\Phi}_{k|k-1} + \boldsymbol{\zeta}_k\boldsymbol{\alpha}_k^{\mathrm{T}}, \tag{28c}$$

$$\boldsymbol{\Psi}_{k|k} = \boldsymbol{\Psi}_{k|k-1} + \boldsymbol{\zeta}_k\boldsymbol{\varphi}(\boldsymbol{\alpha}_{k-1})^{\mathrm{T}}, \tag{28d}$$

$$\boldsymbol{\Lambda}_{k|k} = \boldsymbol{\Lambda}_0 + \boldsymbol{\Phi}_{k|k} - \boldsymbol{M}_{k|k}\boldsymbol{\Psi}_{k|k}^{\mathrm{T}}, \tag{28e}$$

$$\nu_{k|k} = \nu_{k|k-1} + 1, \tag{28f}$$

with the statistics of the predictive distributions given by the time-update step

$$\boldsymbol{\Phi}_{k|k-1} = \lambda\boldsymbol{\Phi}_{k-1|k-1}, \tag{29a}$$

$$\boldsymbol{\Psi}_{k|k-1} = \lambda\boldsymbol{\Psi}_{k-1|k-1}, \tag{29b}$$

$$\boldsymbol{\Sigma}_{k|k-1} = \lambda\boldsymbol{\Sigma}_{k-1|k-1}, \tag{29c}$$

$$\nu_{k|k-1} = \lambda\nu_{k-1|k-1}. \tag{29d}$$

The scalar real-valued number $\lambda \in [0, 1]$ provides exponential forgetting in the data that allows the algorithm to adapt to (slowly time-varying) changes in $\boldsymbol{A}$ and $\boldsymbol{Q}$ over time, and also mitigates path degeneracy (Özkan et al., 2013). To find the posterior distribution of $\boldsymbol{A}$ and $\boldsymbol{Q}$, we marginalize out the state trajectory as

$$p(\boldsymbol{A}, \boldsymbol{Q}|\boldsymbol{y}_{0:k}) = \int p(\boldsymbol{A}, \boldsymbol{Q}|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{0:k})p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k})\,\mathrm{d}\boldsymbol{x}_{0:k} \approx \sum_{i=1}^{N} q_k^i p(\boldsymbol{A}, \boldsymbol{Q}|\boldsymbol{x}_{0:k}^i, \boldsymbol{y}_{0:k}), \tag{30}$$

from where the tire-friction function can be extracted.

3.3 Noise Marginalization

We predict the state trajectory by sampling from the predictive density $p(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^i, \boldsymbol{y}_{0:k-1})$, which depends on $\boldsymbol{A}$ from (7). From marginalization of the unknown quantities,

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^i, \boldsymbol{y}_{0:k-1}) = \int p(\boldsymbol{x}_k|\boldsymbol{A}, \boldsymbol{Q}, \boldsymbol{x}_{k-1}^i) p(\boldsymbol{A}, \boldsymbol{Q}|\boldsymbol{x}_{0:k-1}^i, \boldsymbol{y}_{0:k-1})\mathrm{d}\boldsymbol{A}\mathrm{d}\boldsymbol{Q}. \tag{31}$$

By assumption the second density in the integrand of (31) is $\mathcal{MNIW}$ distributed and the first term is Gaussian, which implies that (31) is $\mathcal{MNT}$ distributed. Using the lemma on transformation of variables in probability density functions (Rao, 2001; Berntorp and Di Cairano, 2018), we can generate states $\boldsymbol{x}_k^i$ by sampling $\boldsymbol{A}_{k-1}^i$ as

$$\boldsymbol{A}_{k-1}^i \sim \mathcal{MNT}(\nu_{k|k-1}^i - 1, \boldsymbol{M}^*, \boldsymbol{\Lambda}^*, \boldsymbol{\Sigma}^*), \tag{32}$$

with

$$\boldsymbol{M}^* = \boldsymbol{\Psi}_{k|k-1}(\boldsymbol{\Sigma}_{k|k-1} + \boldsymbol{V})^{-1},$$
$$\boldsymbol{\Sigma}^* = (\boldsymbol{\Sigma}_{k|k-1} + \boldsymbol{V})^{-1},$$
$$\boldsymbol{\Lambda}^* = \boldsymbol{\Lambda}_0 + \boldsymbol{\Phi}_{k|k-1} - \boldsymbol{\Psi}_{k|k-1}(\boldsymbol{\Sigma}^*)^{-1}\boldsymbol{\Psi}_{k|k-1}^{\top}.$$

The samples $\{\boldsymbol{x}_k^i\}_{i=1}^N$ are then generated by inserting $\boldsymbol{A}_{k-1}^i\boldsymbol{\varphi}(\boldsymbol{\alpha}_{k-1}^i)$ into (7). The resulting samples are used to compute the weights (22) and to update the statistics

(28). In the proposed method, each particle i retains its own estimate of the unknown parameters $\boldsymbol{A}^i$ and $\boldsymbol{Q}^i$. Algorithm 1 summarizes the proposed method.

Algorithm 1 Pseudo-code of proposed algorithm

Initialize: Set $\{\boldsymbol{x}_0^i\}_{i=1}^N \sim p_0(\boldsymbol{x}_0)$, $\{q_{-1}^i\}_{i=1}^N = 1/N$, $\{\nu_0^i, \boldsymbol{\Lambda}_0^i\}_{i=1}^N$, $\boldsymbol{V}$

for $k = 0, 1, \dots$ **do**
  for $i \in \{1, \dots, N\}$ **do**
    Update weight $\bar{q}_k^i$ using (22):
$$\bar{q}_k^i = q_{k-1}^i p(\boldsymbol{y}_k | \boldsymbol{x}_k^i)$$
    Update relevant statistics using (28).
  end for
  Normalize weights as $q_k^i = \bar{q}_k^i / (\sum_{i=1}^N \bar{q}_k^i)$.
  Compute $N_{\text{eff}} = 1/(\sum_{i=1}^N (q_k^i)^2)$
  if $N_{\text{eff}} \leq N_{\text{thr}}$ **then**
    Resample particles and copy the corresponding statistics. Set $\{q_k^i\}_{i=1}^N = 1/N$.
  end if
  Compute state estimate $\hat{\boldsymbol{x}}_k = \sum_{i=1}^N q_k^i \boldsymbol{x}_k^i$.
  Compute friction estimate $\hat{\boldsymbol{\mu}}_k = \sum_{i=1}^N q_k^i \boldsymbol{M}_{k|k}^i \boldsymbol{\varphi}(\boldsymbol{\alpha}_k^i)$.
  for $i \in \{1, \dots, N\}$ **do**
    Predict relevant statistics using (29).
    Predict $\boldsymbol{x}_{k+1}^i$ by sampling from (32) and inserting into (7).
  end for
end for

4. RESULTS PRELIMINARIES

We evaluate Algorithm 1 on simulated measurement data, but with control inputs generated experimentally. We have used a mid-size SUV to gather data and collected several different data sets using the same vehicle setup on a snow-covered track, all data sets roughly 250s long, and the maneuvers are such that the nonlinear region of the tire-force curve is excited at times. The parameters of the vehicle model have been extracted from data sheets and bench testing. In Sec. 5 we present the results of a simulation using the experimental control inputs.

For generating synthetic data, we use a single-track model with longitudinal velocity and steering angle as control inputs. The control inputs are from one of the experimental test-drives. The single-track model and associated measurement equation (8) use the Pacejka tire model Pacejka (2006), with parameters from (Olofsson et al., 2013).

We use 10 basis functions each for the front and rear tire, which gives $m = 100$ basis functions in total. The sampling period is $T_s = 40$ms. The number of particles is $N = 200$. We use a squared exponential covariance function (Rasmussen and Williams, 2006) $\kappa(r) = s_f \exp(-r^2/(2\ell^2))$ with spectral density $S(s) = s_f\sqrt{2\pi\ell^2}\exp(-(\pi^2\ell^2 s^2)(2))$, where $s_f = 500$, $L = 30\pi/180$, and $\ell = 2\pi/180$. Prior knowledge of the tire friction can be used to initialize the algorithm and therefore improve convergence. Initializing the estimate with $\boldsymbol{\mu}(\boldsymbol{\alpha}_k) = \boldsymbol{0}$ can result in large transients as there is no information incorporated the tire-friction function. We can split up the friction function into two parts, $\boldsymbol{\mu}(\boldsymbol{\alpha}_k) = \tilde{\boldsymbol{\mu}}(\boldsymbol{\alpha}_k) + \Delta\boldsymbol{\mu}(\boldsymbol{\alpha}_k)$, where $\tilde{\boldsymbol{\mu}}(\boldsymbol{\alpha}_k)$ is the prior information. In this paper we initialize $\tilde{\boldsymbol{\mu}}(\boldsymbol{\alpha}_k)$ using a Pacejka model with parameters corresponding to a surface with peak friction $\mu_{\max} = 0.05$. The proposed method is robust to the initialization, and it is merely there to inform the estimator that the initial slope of the tire-friction function is positive and crosses the origin.

5. SIMULATION RESULTS

Fig. 1 displays the results from our method for $t = 4, 20, 40, 50$s, respectively. The estimates converge as the number of measurements increases and a larger region of the state space is explored. Note that the times of the snapshots do not imply anything about the convergence of the algorithm. Rather, the tire-friction function estimates are at all times very close to the true underlying Pacejka model for the region in which data have been obtained.

6. CONCLUSION

We presented a method for jointly learning the nonlinear function describing the dependence between wheel slip and tire friction, together with the vehicle state. The method is fully Bayesian and combines a truncated basis-function formulation of Gaussian processes into a particle-filter framework. This gives an online algorithm where each particle retains its own estimate of the unknown tire-friction function and vehicle state. A key feature is that the method only uses sensors that are typically installed in production vehicles.

The simulation results show that the method is capable of accurately learning online the tire-friction function, but it is future work to verify the method experimentally. More interesting future work could be how to incorporate the considered approach for providing less conservative vehicle control.

REFERENCES

Ahn, C., Peng, H., and Tseng, H.E. (2012). Robust estimation of road friction coefficient using lateral and longitudinal vehicle dynamics. *Veh. Syst. Dyn.*, 50(6), 961–985.

Berntorp, K. (2019a). Bayesian inference and learning of Gaussian-process state-space models. In *Eur. Control Conf.* Naples, Italy.

Berntorp, K. (2019b). Bayesian tire-friction learning by Gaussian-process state-space models. In *Eur. Control Conf.* Naples, Italy.

Berntorp, K. and Di Cairano, S. (2018). Tire-stiffness and vehicle-state estimation based on noise-adaptive particle filtering. *IEEE Trans. Control Syst. Technol.*, 27(3), 1100–1114.

Berntorp, K. and Hiroaki, K. (2019). Bayesian learning of tire friction with automotive-grade sensors by Gaussian-process state-space models. In *IEEE Conf. Decision and Control.* Nice, France.

Carvalho, A., Leféve, S., Schildbach, G., Kong, J., and Borrelli, F. (2015). Automated driving: The role of forecasts and uncertainty - a control perspective. *Eur. J. Control*, 24, 14–32.

Dawid, A.P. (1981). Some matrix-variate distribution theory: notational considerations and a Bayesian application. *Biometrika*, 68(1), 265–274.

Doucet, A. and Johansen, A.M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan and B. Rozovsky (eds.), *Handbook of Nonlinear Filtering*. Oxford University Press.

Frasch, J.V., Gray, A., Zanon, M., Ferreau, H.J., Sager, S., Borrelli, F., and Diehl, M. (2013). An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles. In *Eur. Control Conf.* Zurich, Switzerland.

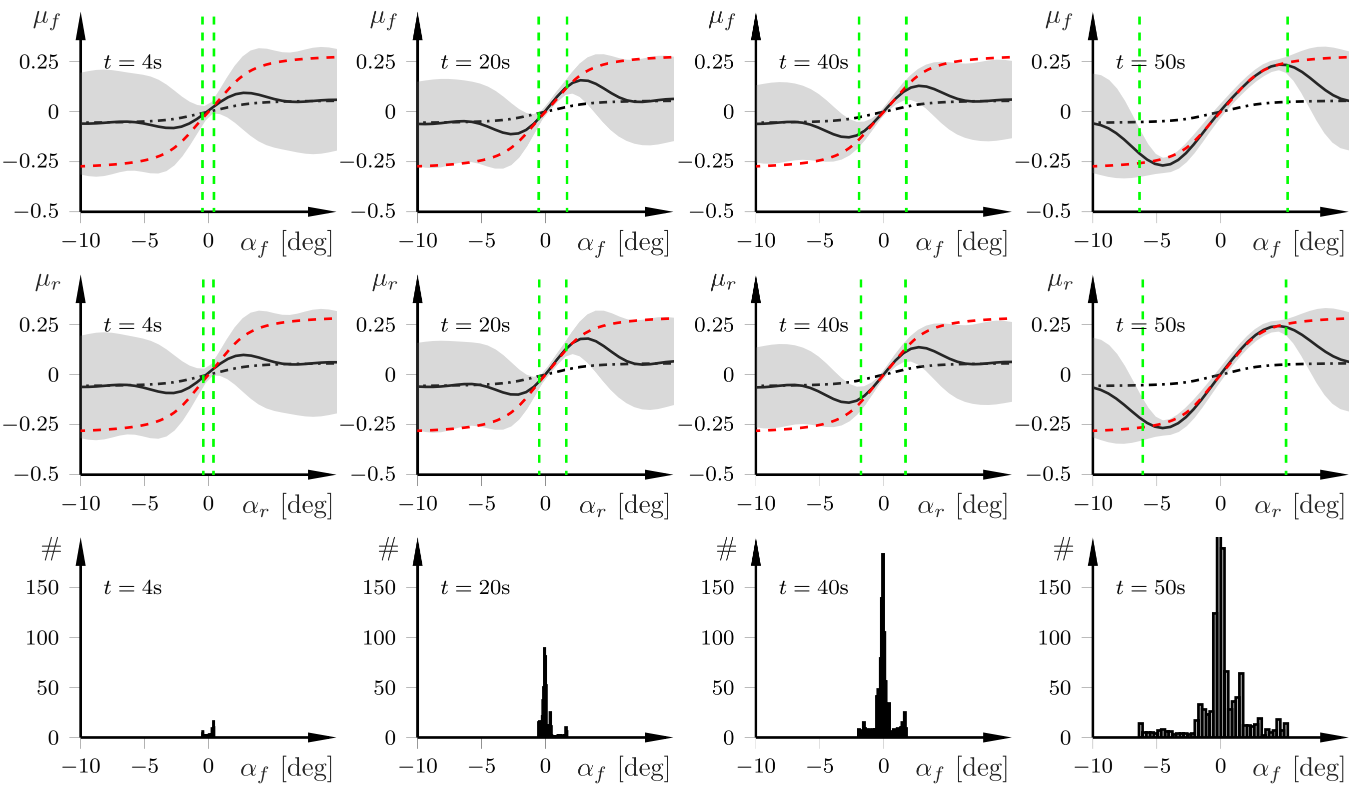

Fig. 1. The posterior estimates of our proposed method, for the simulated tire-road friction estimation example in Sec. 5. The upper plots show the front tire friction and the middle plots show the rear tire friction. Estimated function in black, true function in dashed red, posterior uncertainty in gray, and the initial estimate in black dash-dotted. The lower plots display the accumulated data points of the underlying slip-angle for the front tire (not used in the learning), and the green vertical dashed lines in the upper two plots indicate the range of the data.

Frigola, R., Chen, Y., and Rasmussen, C.E. (2014). Variational Gaussian process state-space models. In *Advances in Neural Information Processing Systems*. Montreal, Canada.

Garatti, S. and Bittanti, S. (2009). Parameter estimation in the Pacejka's tyre model through the TS method. In *IFAC Symp. System Identification*. Saint-Malo, France.

Goldfain, B., Drews, P., You, C., Barulic, M., Velev, O., Tsiotras, P., and Rehg, J.M. (2019). Autorally: An open platform for aggressive autonomous driving. *IEEE Control Syst. Mag.*, 39(1), 26–55.

Gustafsson, F. (2009). Automotive safety systems. *IEEE Signal Processing magazine*, 26(4), 32–47.

Kiencke, U. and Nielsen, L. (2005). *Automotive Control Systems—For Engine, Driveline and Vehicle*. Springer-Verlag, Berlin Heidelberg, 2nd edition edition.

Lee, S., Nakano, K., and Ohori, M. (2015). On-board identification of tyre cornering stiffness using dual Kalman filter and GPS. *Veh. Syst. Dyn.*, 53(4), 437–448.

Lundquist, C. and Schön, T.B. (2009). Recursive identification of cornering stiffness parameters for an enhanced single track model. In *15th IFAC Symp. System Identification*. Saint-Malo, France.

Murphy, K.P. (2007). Conjugate Bayesian analysis of the Gaussian distribution. Technical report, UBC.

Olofsson, B., Lundahl, K., Berntorp, K., and Nielsen, L. (2013). An investigation of optimal vehicle maneuvers for different road conditions. In *IFAC Symp. Adv. Automotive Control*. Tokyo, Japan.

Özkan, E., Šmídl, V., Saha, S., Lundquist, C., and Gustafsson, F. (2013). Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters. *Automatica*, 49(6), 1566–1575.

Pacejka, H.B. (2006). *Tire and Vehicle Dynamics*. Butterworth-Heinemann, Oxford, United Kingdom, 2nd edition edition.

Quirynen, R., Berntorp, K., and Di Cairano, S. (2018). Embedded optimization algorithms for steering in autonomous vehicles based on nonlinear model predictive control. In *Amer. Control Conf.* Milwaukee, WI.

Rao, C.R. (2001). *Linear Statistical Inference and its Applications*. Wiley.

Rasmussen, C. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, USA.

Solin, A. and Särkkä, S. (2014). Hilbert space methods for reduced-rank Gaussian process regression. *arXiv preprint arXiv:1401.5508*.

Svendenius, J. (2007). *Tire modeling and Friction Estimation*. Ph.D. thesis, Dept. Automatic Control, Lund University, Sweden.

Svensson, A. and Schön, T.B. (2017). A flexible state-space model for learning nonlinear dynamical systems. *Automatica*, 80, 189–199.

Svensson, A., Solin, A., Särkkä, S., and Schön, T.B. (2016). Computationally efficient Bayesian learning of Gaussian process state space models. In *Int. Conf. Artificial Intelligence and Statistics*. Cadiz, Spain.