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

Vehicle Center-of-Gravity Height and Dynamics Estimation with Uncertainty Quantification by Marginalized Particle Filter

Berntorp, Karl; Chakrabarty, Ankush; Di Cairano, Stefano

TR2021-058 June 04, 2021

Abstract

This paper addresses the center-of-gravity height and dynamics estimation problem, which is key for rollover prevention systems in automotive. We model the vehicle as a spring-damper system and develop a Bayesian method that outputs estimates of the center-of-gravity height, suspension stiffness and damping coefficient. We leverage the model structure to design a computationally efficient particle filter, which, combined with Bayesian optimization for particle initialization and a particle-size adaptation scheme, leads to an implementation that provides accurate, smooth estimates of CoG height, stiffness, and damping. A Monte-Carlo simulation study on a standardized maneuver shows that the method almost instantaneously provides reliable estimates that represent well the true parameter values.

American Control Conference (ACC)

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

Vehicle Center-of-Gravity Height and Dynamics Estimation with Uncertainty Quantification by Marginalized Particle Filter

Karl Berntorp, Ankush Chakrabarty, and Stefano Di Cairano

***Abstract*— This paper addresses the center-of-gravity height and dynamics estimation problem, which is key for rollover-prevention systems in automotive. We model the vehicle as a spring-damper system and develop a Bayesian method that outputs estimates of the center-of-gravity height, suspension stiffness and damping coefficient. We leverage the model structure to design a computationally efficient particle filter, which, combined with Bayesian optimization for particle initialization and a particle-size adaptation scheme, leads to an implementation that provides accurate, smooth estimates of CoG height, stiffness, and damping. A Monte-Carlo simulation study on a standardized maneuver shows that the method almost instantaneously provides reliable estimates that represent well the true parameter values.**

I. Introduction

To design roll-dynamics control methods, such as rollover avoidance systems based on differential braking systems or active front steering (AFS), it is imperative to have at least basic understand of the inertial parameters, such as mass and inertia, and the location of the mass, that is, the center of gravity (CoG). Automotive manufacturers can provide values for the inertial parameters and the CoG. However, they are typically for some nominal (usually empty) loading conditions, whereas in reality the loading conditions, hence the CoG, will vary significantly between different drives.

Rollover accidents are relatively uncommon but constitute a large portion of severe accidents and fatalities [1]. Hence, development of improved control principles for avoiding such accidents is a major consideration for automotive manufacturers. According to [2], usually automotive manufacturers employ robust active road-handling control strategies to account for the unknown and changing CoG, by designing for the worst-case scenario. Another approach in the case of Sport Utility Vehicles (SUVs) is to intentionally design the vehicle heavier than usual by adding ballast in the undercarriage, which aims to lower the CoG position while reducing the percent margin of the load variation and thus constraining the variation of the CoG location. While such approaches are successful up to certain extent, they come with drawbacks, such as performance loss under normal driving conditions and reduced efficiency due to added weight.

To avoid designing for the worst-case scenario, the control performance benefits from real-time CoG location estimation. Such estimation can be used to warn the driver or can be integrated into rollover prevention systems.

The CoG estimation problem has been researched concurrently with the rollover avoidance control problem. The methods span from recursive linear least-squares (RLS) [3], multiple-model estimation [2], and extended Kalman filters (EKF) [4]. The method in [3] assumes a model that is linear in the parameters. The approach in [4] utilizes the correlation between the vertical load of a tire and the tire instant effective radius, but needs access to the tire radius. In [2], the sensing assumed can be extracted from modern production vehicles. However, it employs a multiple-model approach that; (i), can be unsuitable to use in control applications due to the inherent chattering heavior that can arise when switching models; (ii), does not include uncertainty quantification and therefore it can be difficult to assess the quality of the estimate when integrated into a control loop; and (iii), needs a priori determined ranges of all involved parameters, which to ensure convergence may need to require a very large number of models in the estimator.

In this paper, we provide a method for estimating the CoG height h. We model the roll dynamics as a torsional spring-damper system, where the roll angle ϕ and roll rate $\dot{\phi}$ are the states and the spring stiffness K and damping D are parameters estimated jointly with the state and CoG height h. We formulate an estimation method based on a marginalized particle filter (MPF, [5]), where K, D, and h are estimated using a PF and conditioned on the particles, the states can be analytically estimated with a conditional Kalman filter (KF). PFs solve nonlinear, non-Gaussian estimation problems by generating random state trajectories and assigning a weight to them according to how well they predict the observations [6]. When there is a tractable substructure present, parts of the state space can be estimated analytically, which results in the MPF. Since the MPF estimates the distribution of the involved variables, the proposed method also provides uncertainty estimates, which can be used for assessing the quality of the estimates. For PFs, it is crucial to have enough number of particles, especially in the transient phase, since otherwise it may lead to poor performance, or divergence of the filter [6]. To cope with this, we design two additional stages of the MPF;

(i) we initialize the method using Bayesian optimization (BO), where we leverage limited archival data to estimate offline a distribution of most likely initial values for the parameters of interest; and
(ii) we implement a control mechanism based on controlling the *effective sample size* in PFs for monitoring the quality of the particles.

These two additional design steps enable us to start off with a good initial guess thus limiting the number of particles in

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

the MPF while ensuring that we keep the number of particles at a reasonable level that trades-off estimation quality and computational demands.

BO methods solve optimization problems where the objective function is cumbersome or expensive to evaluate over the whole search space, and therefore incorporates an uncertainty on the objective as a distribution function. BO methods have been used for learning system dynamics online (e.g., [7]). For estimation, a recent paper has demonstrated the utility of BO methods for concurrently learning dynamical models from data while maintaining robustness certificates (local input-to-state stability) of the estimation error, even during learning [8]. In the current paper, we leverage limited offline data and BO methods to estimate a distribution function to generate sensible initial guesses for the MPF.

Notation: By $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k})$, 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 $\boldsymbol{y}_{0:k} := \{\boldsymbol{y}_0, \dots, \boldsymbol{y}_k\}$. 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 $\boldsymbol{\Sigma}$. The notation $\hat{\boldsymbol{z}}_{k|m}$ denotes the estimate of $\boldsymbol{z}$ at time index k given measurements up to time index m, $\mathrm{Var}(q)$ is the variance of the variable q, and $\mathrm{ceil}(\cdot)$ is the ceiling function.

II. Modeling

The suspension system is modeled as a torsional spring-damper system (Fig. 1), and has been used before in CoG estimation [2]. We introduce three assumptions.

Assumption 1. *The entire mass is sprung and rolls about a roll axis centered at the ground level of the body.*

Assumption 2. *The roll angle is sufficiently small such that* $\sin\phi \approx \phi$ *and* $\cos\phi \approx 1$.

Assumption 3. *The lateral acceleration a_y, roll angle ϕ, and roll rate $\dot{\phi}$ are measured at every time step.*

While our method does not strictly need Assumption 1, it simplifies modeling slightly. Considering the sprung mass as the whole vehicle mass is an approximation, but for regular passenger vehicles, and SUVs in particular, the unsprung mass can constitute as little as a few percent of the total mass [2], [9]. Assumption 2 basically states that the vehicle either assumes normal driving maneuvers, that is, such that the roll dynamics are not overly excited, or that there is an active rollover protection control system implemented in the vehicle. This assumption can be relaxed, for example, by introducing nonlinear filters also for the roll state estimation. Finally, Assumption 3 is satisfied by vehicles equipped with an electronic stability-control system (lateral acceleration) and a smart suspension system (ϕ, the suspension needs to have a height or elongation sensor). If the vehicle is equipped with a three-dimension inertial measurement unit (IMU), ϕ and $\dot{\phi}$ can instead be determined by a combination of the gyro measurements and the vertical acceleration.

As described in [2], Assumption 1 enables us to model

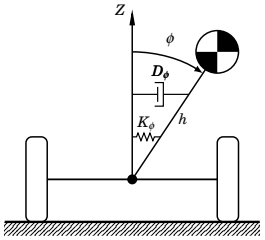

Fig. 1. The torsional spring-damper model for roll dynamics used in this paper, where the chassis roll is modeled with one spring and damper.

the roll-plane dynamics by

$$(I_x + mh^2)\ddot{\phi} + D\dot{\phi} + K\phi = mh(a_y \cos\phi + g\sin\phi), \quad (1)$$

where g is the gravitational acceleration, m is the mass, and I_x is the moment of inertia about the vehicle x-axis. By Assumption 2 and setting $J = I_x + mh^2$, we can write the system on state-space form with the state $\boldsymbol{x} = \begin{bmatrix}\phi & \dot{\phi}\end{bmatrix}^\top$ as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ -\frac{K-mgh}{J} & -\frac{D}{J} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ \frac{mh}{J} \end{bmatrix} a_y, \quad (2)$$

Remark 1. *At steady state $\dot{\boldsymbol{x}} = 0$, we can solve for*

$$h = \frac{K\phi}{m(g\phi + a_y)}.$$

However, K is unknown and will change with loading conditions so this is not practical.

We form the estimation model based on (2). First, note that (2) is linear in $\boldsymbol{x}$, but it is nonlinear in the parameters. Second, all parameters will be time varying in the sense that they will change depending on the loading conditions, but when the vehicle is moving, they are unlikely to have large variations. Hence, it is appropriate to model the parameters as nearly constant position models, which in discrete time for the notation $\boldsymbol{\theta} = \begin{bmatrix}K & D & h\end{bmatrix}^\top$ means

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{w}_{\boldsymbol{\theta},k}, \quad (3)$$

where $\boldsymbol{w}_{\boldsymbol{\theta},k}$ is modeled as zero-mean Gaussian distributed with covariance $\boldsymbol{Q}_{\boldsymbol{\theta}}$ according to $\boldsymbol{w}_{\boldsymbol{\theta},k} \sim \mathcal{N}(0, \boldsymbol{Q}_{\boldsymbol{\theta}})$, with prior distribution $\boldsymbol{\theta}_0 \sim p_0(\boldsymbol{\theta})$.

Based on (2) and (3) and Assumption 3, the model is

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{F}\boldsymbol{\theta}_k + \boldsymbol{w}_{\boldsymbol{\theta},k}, \quad (4a)$$

$$\boldsymbol{x}^e_{k+1} = \boldsymbol{A}(\boldsymbol{\theta}_k)\boldsymbol{x}^e_k + \boldsymbol{B}(\boldsymbol{\theta}_k)a_y, \quad (4b)$$

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}^e_k + \boldsymbol{e}_k, \quad (4c)$$

where $\boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$ and the lateral acceleration is modeled as $a_y \sim \mathcal{N}(a_{y,m}, Q_a)$, where $a_{y,m}$ is the measurement and Q_a can be determined using standard methods [10].

III. COG Height Estimation by Marginalized Particle Filter

The Bayesian approach to filtering is to compute the posterior density $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k})$ of the state trajectory $\boldsymbol{x}_{0:k}$ given the measurement history $\boldsymbol{y}_{0:k}$. When the dynamics are linear with Gaussian noise, the optimal estimator is given by the KF [11]. Generally, numerical approximations are

required. PFs [12] represent the posterior density with a set of weighted particles. Each particle represents a state trajectory, which gives the approximation

$$p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k}) \approx \sum_{i=1}^{N} q_k^i \delta_{\boldsymbol{x}_{0:k}^i}(\boldsymbol{x}_{0:k}). \tag{5}$$

In (5), $\delta(\cdot)$ is the Dirac delta mass and q_k^i is the associated importance weight for the ith particle given the measurements $\boldsymbol{y}_{0:k}$. The particle weights are updated as $q_k^i \propto q_{k-1}^i p(\boldsymbol{y}_k|\boldsymbol{x}_k^i)$. To avoid having a significant dependence on a few particles with large weights (i.e., particle depletion [10]), a crucial resampling step is carried out, which provides an equally-weighted distribution. To decrease the number of particles and the variance of the estimates, it is advantageous to exploit model structure. This is the idea behind marginalization, where the subset of the state space that allows for analytic expressions is marginalized out. It is therefore possible to use fewer particles. The enabler for the MPF is the factorization

$$p(\boldsymbol{x}_k, \boldsymbol{\theta}_{0:k}|\boldsymbol{y}_{0:k}) = p(\boldsymbol{x}_k|\boldsymbol{\theta}_{0:k}, \boldsymbol{y}_{0:k}) p(\boldsymbol{\theta}_{0:k}|\boldsymbol{y}_{0:k}) \tag{6}$$

The second distribution in (6) is approximated by the PF. Given the nonlinear state trajectory, the first part in (6) is linear Gaussian. Thus, it can be estimated with conditional KFs, one for each particle conditioned on the particle trajectory. Generally, the main difference compared with the standard KF consists of performing an extra measurement update for each KF using the forward propagated $\boldsymbol{\theta}_{k+1}^i$ as an extra measurement. For the base implementation of the MPF used in this paper, model (4) is equivalent to *Model 1* in [5], which gives the weight update

$$\bar{q}_k^i = \mathcal{N}(\boldsymbol{y}_k|\boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1}^i, \boldsymbol{C}\boldsymbol{P}_{k|k-1}^i\boldsymbol{C}^\top + \boldsymbol{R}) \tag{7}$$

and the standard KF measurement update

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k}^i &= \hat{\boldsymbol{x}}_{k|k-1}^i + \boldsymbol{K}_k^i(\boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1}^i), \\ \boldsymbol{P}_{k|k}^i &= \boldsymbol{P}_{k|k-1}^i - \boldsymbol{K}_k^i\boldsymbol{C}\boldsymbol{P}_{k|k-1}^i, \\ \boldsymbol{S}_k^i &= \boldsymbol{C}\boldsymbol{P}_{k|k-1}^i\boldsymbol{C}^\top + \boldsymbol{R}, \\ \boldsymbol{K}_k^i &= \boldsymbol{C}\boldsymbol{P}_{k|k-1}^i(\boldsymbol{S}_k^i)^{-1}. \end{aligned} \tag{8}$$

Due to the model structure, the KF time update is

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k-1}^i &= \boldsymbol{A}(\boldsymbol{\theta}_k^i)\hat{\boldsymbol{x}}_{k-1|k-1}^i + \boldsymbol{B}(\boldsymbol{\theta}_k^i)a_y, \\ \boldsymbol{P}_{k|k-1}^i &= \boldsymbol{A}(\boldsymbol{\theta}_k^i)\boldsymbol{P}_{k-1|k-1}^i\boldsymbol{A}(\boldsymbol{\theta}_k^i)^\top + \boldsymbol{B}(\boldsymbol{\theta}_k^i)Q_a\boldsymbol{B}(\boldsymbol{\theta}_k^i)^\top. \end{aligned} \tag{9}$$

A. Initialization and Speedup

The MPF for our model structure provides asymptotic convergence guarantees, meaning that as the number of particles $N \to \infty$, the MPF estimates converge to the true state both in the mean-square sense and in distribution [10]. A key question is what value of N is sufficient large to get estimates that are accurate enough to be of practical value. For our estimation problem, the initial possible range of values of the spring stiffness, damping coefficient, and CoG height, varies heavily between vehicle type and loading conditions. Hence, to ensure sensible results coming out of the estimator, without any additional knowledge about the parameter range at initialization time, a very large number of particles is needed to make sure that the initial guess has any reasonable chance to converge to the true parameters. We now provide two add-on steps to the standard MPF. First, we desribe a procedure to warm-start the MPF using Bayesian optimization, which provides *a priori* information of how to sample the initial guess of the parameter set. Second, we include a sample-size adaptation of the MPF to decrease the number of particles needed while maintaining reliable estimation performance.

B. Warm Starting the MPF using Bayesian Optimization

BO methods solve optimization problems of the form

$$\min_{\boldsymbol{\theta}\in\Theta} \mathcal{J}(\boldsymbol{\theta}),$$

where the objective function $\mathcal{J}(\boldsymbol{\theta})$ is usually too expensive to evaluate over the admissible search space Θ (assumed to be known), and an analytical form of the function is unavailable, making it impossible to derive analytical gradients. To prevent prohibitive expenditure in evaluating the cost, the BO framework models the uncertainty on $\mathcal{J}$, at $\boldsymbol{\theta}$ values that have not yet been evaluated, as a probability distribution. In particular, BO can employ Gaussian-process regression [13] to construct a surrogate $\tilde{\mathcal{J}}$ of the true objective function that can be evaluated relatively cheaply and, therefore, evaluated often. At each iteration of the BO algorithm, the Gaussian process regressor $\tilde{\mathcal{J}}$ is used to inform which $\boldsymbol{\theta} \in \Theta$ is the next best candidate for evaluation; upon selecting this next best candidate $\boldsymbol{\theta}^*$, the corresponding true function value $\mathcal{J}(\boldsymbol{\theta}^*)$ is evaluated, and the Gaussian process updates its posterior belief by appending the newly-obtained data pair $\boldsymbol{\theta}^*, \mathcal{J}(\boldsymbol{\theta}^*)$ to the prior dataset. While re-learning the Gaussian process regressor involves solving a nonconvex problem and inversion of a square matrix of growing dimension, inference is cheap [14]. Iterating and re-learning the surrogate map results in a balance of exploration and exploitation that is automatic in the BO framework [15].

A crucial component of BO methods is the acquisition of the next candidate $\boldsymbol{\theta}^\star$ based on the surrogate cost function. Typically, this is done via an acquisition function which is designed to yield regions in the admissible search space Θ, based on the Gaussian process, where the optimal solution most likely lies. While many suitable acquisition functions have been investigated in the literature, we use the expected improvement function, which is explained hereafter.

Suppose we have run the BO algorithm for m iterations, and let $\boldsymbol{\theta}^+$ denote the optimal solution obtained thus far. That is, $\boldsymbol{\theta}^+ := \arg\min\{\mathcal{J}(\boldsymbol{\theta}_t)\}_{t=1}^m$. The expected improvement acquisition function for a Gaussian prior at any $\boldsymbol{\theta} \in \Theta$ is given by $\mathcal{A}_{\mathrm{EI}}(\boldsymbol{\theta}) = \tilde{\sigma}(\boldsymbol{\theta})\big(z\Phi_{\mathrm{G}}(z) + \phi_{\mathrm{G}}(z)\big)$, where

$$z = \frac{\tilde{\mu}(\boldsymbol{\theta}) - \mathcal{J}(\boldsymbol{\theta}^+)}{\tilde{\sigma}(\boldsymbol{\theta})}.$$

Here, Φ_{G} and ϕ_{G} denote the cdf and pdf of the standard (zero mean, unit variance) Gaussian distribution, and $\tilde{\mu}$ and

$\tilde{\sigma}$ are the mean and variance obtained from the surrogate $\tilde{J}$. Using the acquisition function $\mathcal{A}_{\mathrm{EI}}$ one can obtain the next best candidate $\boldsymbol{\theta}^* := \arg\min_{\Theta} \mathcal{A}_{\mathrm{EI}}(\boldsymbol{\theta})$, with which the function value $\mathcal{J}(\boldsymbol{\theta}^*)$ can be evaluated.

After N_{BO} iterations of the BO algorithm, we have accumulated a history $\{\boldsymbol{\theta}_t^*, \mathcal{J}(\boldsymbol{\theta}_t^*)\}_{t=1}^{N_{\mathrm{BO}}}$. This history comprises both $\boldsymbol{\theta}^*$ values which were found during exploration, and, in later iterations of the BO procedure, $\boldsymbol{\theta}^*$ values that had a high likelihood of being the optimum. Since both exploration and exploitation information is embedded in this history, and assuming that N_{BO} is not very small, we can estimate a probability density function on $\boldsymbol{\theta}^*$ values using, for example, kernel density estimators [8], from which we can obtain an initial density $p_0(\boldsymbol{\theta})$ by taking a high confidence interval, e.g., the 95% confidence interval. Algorithm 1 provides pseudo-code of the BO warm-start procedure, where $\mathcal{Y}$ is a set of measurements (the data) over a predefined time span.

Algorithm 1 Pseudo-code of the BO warm-start procedure

Input: Measured sequence $\mathcal{Y}$, Estimation model $\mathcal{E}$, Initial guess $\boldsymbol{\theta}_0^*$ of (K, D, h), Admissible range Θ, Acquisition function $\mathcal{A}$, Confidence interval $\beta \in (0, 1)$.

for $t \leftarrow 0$ to N_{BO} **do**
  $\hat{\mathcal{Y}}_t \leftarrow$ output of $\mathcal{E}$ with parameters set to θ_t^*
  $\mathcal{J}(\boldsymbol{\theta}_t^*) \leftarrow \mathcal{L}_2$ (or other error metric) between $\mathcal{Y}$, $\hat{\mathcal{Y}}_t$
  Update history $\{\boldsymbol{\theta}_j^*, \mathcal{J}(\boldsymbol{\theta}_j^*)\}_{j=0}^{t}$
  Update Gaussian process model with history
  $\boldsymbol{\theta}_{t+1}^* \leftarrow \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{A}(\theta)$
end for
Estimate density π from $\boldsymbol{\theta}^*$ history using (kernel) density estimation
$p_0(\boldsymbol{\theta}) \leftarrow$ compute β-confidence interval from π

C. Controlling the Effective Sample Size

PFs generally exhibit convergence rates that are inversely proportional to the square root of the number of particles [16]. However, for finite N, convergence rates are usually problem-dependent and difficult to predict before implementation. There are several methods for adapting the number of particles N during runtime (c.f. [17], [18]), which all look at different metrics for controlling the number of particles in response to the estimation quality.

We adapt the method in [17], which controls the number of particles by monitoring the *effective sample size* (ESS). The ESS N_{eff} gives an indication of the degree of lack of particles corresponding to the true state and is usually defined as $N_{\mathrm{eff}} = \frac{N}{1+\mathrm{Var}(q_k)}$. By predefining a lower bound N^* on the ESS, the number of particles needed at time step k to achieve an ESS of N^* can be calculated as $N_k = \mathrm{ceil}(N^*(1 + \mathrm{Var}(q_k)))$, and the variance $\mathrm{Var}(q_k)$ can be expanded as

$$\mathrm{Var}(q_k) = \mathbb{E}(q_k^2) - (\mathbb{E}(q_k))^2. \tag{10}$$

Solving for (10) involves solving a set of integral equations and cannot be done in closed-form since they require the state, which the particle filter approximates. However, they can be solved numerically, for example, by using Monte-Carlo integration. An approximation suggested in [17] is to utilize that $N_k \geq N^*$. Hence, the first N^* particles generated by the particle filter sampling step can be used to approximate the integrals. Another approach, which we use in this paper, is to use the Monte-Carlo integration resulting from the N_{k-1} particles generated in the previous time step.

Algorithm 2 provides the pseudo-code of the proposed estimation algorithm. As with all identification methods, the system needs to be excited properly for the parameters to be identifiable uniquely from the data. The degree of excitation can be tested at runtime and the estimator can be turned on and off in response to this test. For a simple, yet effective procedure for activating/deactivating the method in repsonse to excitation, see, for example, [19].

Algorithm 2 Pseudo-code of the estimation algorithm

Warm-Start: Execute Algorithm 1 to get an initial distribution $p_0(\boldsymbol{\theta})$.
Initialize: Set N_{-1}, $\{\boldsymbol{x}_0^i\}_{i=1}^{N_{-1}} \sim p_0(\boldsymbol{x}_0)$, $\{q_0^i\}_{i=1}^{N_{-1}} = 1/N_{-1}$, $\{\boldsymbol{\theta}_0^i\}_{i=1}^{N_{-1}} \sim p_0(\boldsymbol{\theta})$, and define desired ESS N^*.

for $k \leftarrow 0$ to $\ldots$ **do**
  for each particle $i \in \{1, \ldots, N_{k-1}\}$ **do**
    Update weights according to (7).
  end for
  Normalize weights as $q_k^i = \bar{q}_k^i / (\sum_{i=1}^{N_{k-1}} \bar{q}_k^i)$.
  Approximate (10) to get $\mathrm{Var}(q_k)$ (Sec. III-C and [17]).
  Determine $N_k = \mathrm{ceil}(N^*(1 + \mathrm{Var}(q_k)))$.
  Resample N_k particles and copy the corresponding statistics. Set $\{q_k^i\}_{i=1}^{N_k} = 1/N_k$.
  for each particle $i \in \{1, \ldots, N_k\}$ **do**
    Determine $\boldsymbol{x}_{k|k}^i$ and $\boldsymbol{P}_{k|k}^i$ using (8).
  end for
  for each particle $i \in \{1, \ldots, N_k\}$ **do**
    Determine $\boldsymbol{\theta}_{k+1}^i$ using (3).
    Determine $\boldsymbol{x}_{k+1|k}^i$ and $\boldsymbol{P}_{k+1|k}^i$ using (9).
  end for
end for

IV. Simulation Study

We use the Sine-with-dweel maneuver, standardized by NHTSA. The estimation model used for Algorithm 2 is the quite simple linear-in-state model presented in Sec. II. However, the simulation model is a nonlinear five-state vehicle model that includes lateral and longitudinal dynamics, nonlinear tire forces including combined slip, and suspension dynamics, which includes both sprung and unsprung mass models. This model is substantially more complex than the estimation model, and the results therefore also indicate the robustness of the method to modeling errors. The model and the parameters used are the same as in [20], where the parameters are set to be similar to a North-American SUV.

For the BO warm-start procedure, we simulate the system for the entirety of the Sine-with-dwell maneuver. With this

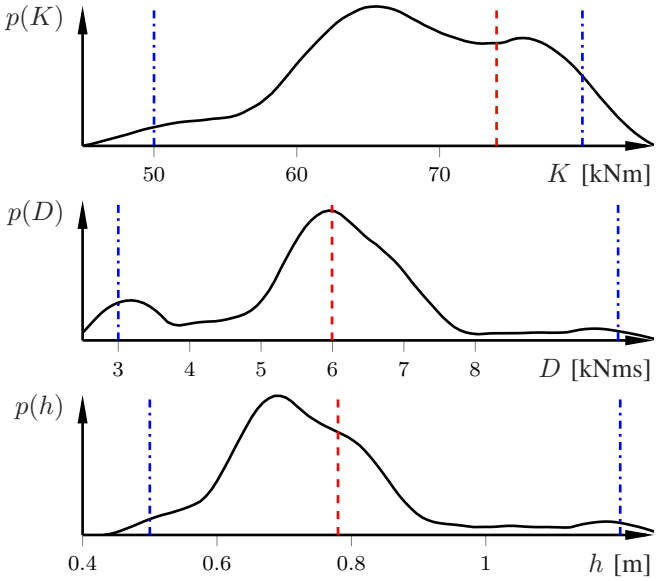

Fig. 2. Resulting estimated initial distribution using the Bayesian optimization procedure in Sec. III-B. True underlying parameter values are indicated by the red vertical dashed lines and the initial uniform ranges are indicated by the blue vertical dash-dotted lines.

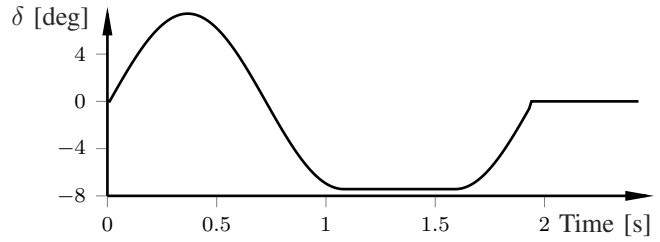

Fig. 3. The road-wheel steering angle for the Sine-with-dwell maneuver.

single trajectory, the BO algorithm is iterated 100 times using the estimation model described in Sec. II with different $\boldsymbol{\theta}$ learned by the BO procedure. Initially, we sample parameters K, D, h from the set $[50000, 80000] \times [3000, 10000] \times [0.5, 1.2]$, and the resulting estimates of the distribution of parameters using the BO procedure in Sec. III-B are shown in Fig. 2. The BO warm-start procedure shrinks the range of likely values around the true values, and the true parameters all lie quite close to the maximum-likelihood estimate, and certainly within the 95% confidence intervals induced by the respective density functions, as shown in Fig. 2. In Algorithm 2, the distributions in Fig. 2 correspond to $p_0(\boldsymbol{\theta})$ and are the distributions from which we sample the initial guess in the MPF. To partition out the initial particles we sample 20 values according to the probablities for each of the dimensions in Fig. 2, which gives $N_{-1} = 20^3 = 8000$ particles initially. For the particle adaptation, $N^* = 1000$.

Remark 2. *The offline estimated distributions in Fig. 2 using Algorithm 1 are highly non-Gaussian and there is even a tendency of bimodality, motivating the use of a PF.*

We have executed 1000 Monte-Carlo runs of the Sine-with-dwell maneuver, see Fig. 3. Fig. 4 shows the estimation results for one representative Monte-Carlo realization. Due to the BO warm-starting, the sampled particle range initially covers the true parameter range, and, hence, we immediately

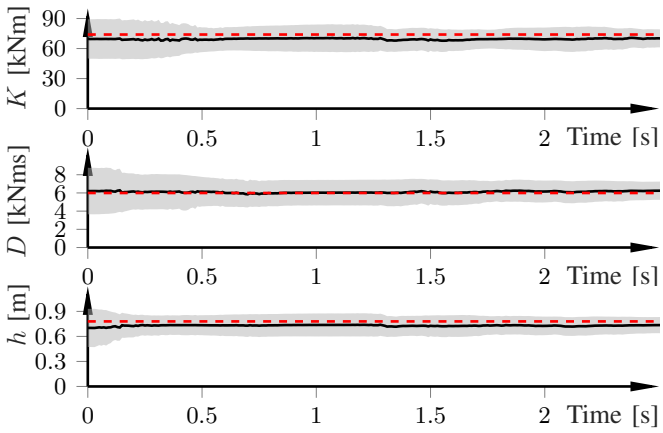

Fig. 4. Parameter estimation results for one Monte-Carlo execution for the Sine-with-dwell maneuver in Fig. 3. True values in red dashed, estimates in black, and 3σ area in gray.

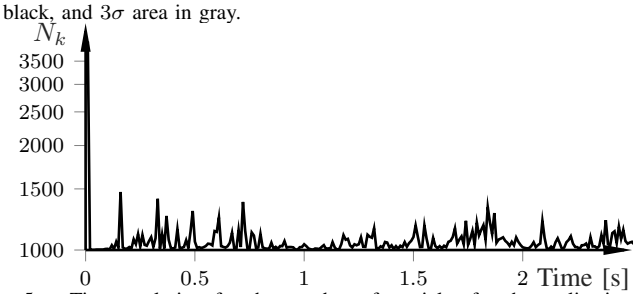

Fig. 5. Time evolution for the number of particles for the realization corresponding to Fig. 4.

get estimates that are very close to the true values. However, as time increases, the certainty about the estimates increase over time. The average CoG height estimate over the time span is $\hat{h} \approx 0.74$m, which is 0.04m away from the true value. Fig. 5 displays the time evolution of the number of particles needed to achieve an ESS of $N^* = 1000$. The number of particles decreases within a few samples, from $N_{-1} = 8000$.

Figs. 6–8 compare the estimation results of Algorithm 2 with a version of Algorithm 2 that uses adaptive particle size selection but initializing the filter from the initial, uniform range of parameter values, denoted by NONBO, and a version with BO for warm-starting but without adaptive particle size selection, denoted by NONADAP. For NONADAP, we have chosen the number of particles N to equal the average of the number of particles used in Algorithm 2, that is, an average of Fig. 5 for all Monte-Carlo runs in this example. Clearly, not using BO for initialization of the MPF, as is the case for NONBO, leads to much larger variance of the estimator, even though the mean after the initial transients is very similar to Algorithm 2. On the other hand, by fixing the number of particles beforehand as in NONADAP, we get a considerably biased estimator that suffers from particle depletion, implied by the small estimated variance in combination with that the mean is biased for all three parameters.

V. CONCLUSION

We developed a method for estimating the CoG height h, along with the suspension parameters K, D, when the vehicle roll-dynamics are modeled as a torsional spring-damper system. We leveraged that the roll dynamics are

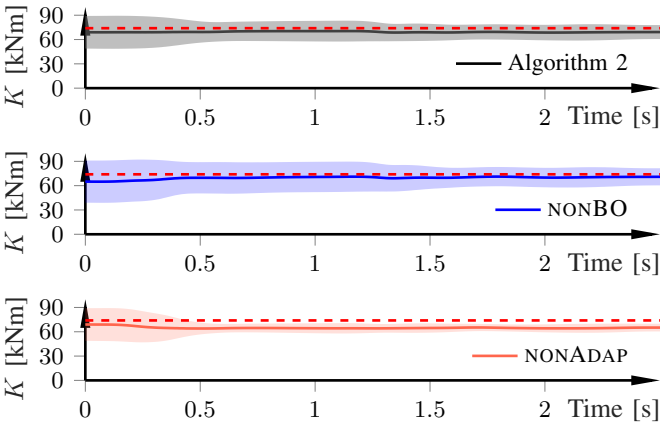

Fig. 6. Averaged estimation results of the spring stiffness K for 1000 Monte-Carlo executions for the Sine-with-dwell maneuver in Fig. 3, with 3σ shown as shaded areas. Algorithm 2 and NONBO have similar mean but the uncertainty is significantly larger for NONBO. NONADAP does not represent well the underlying value.

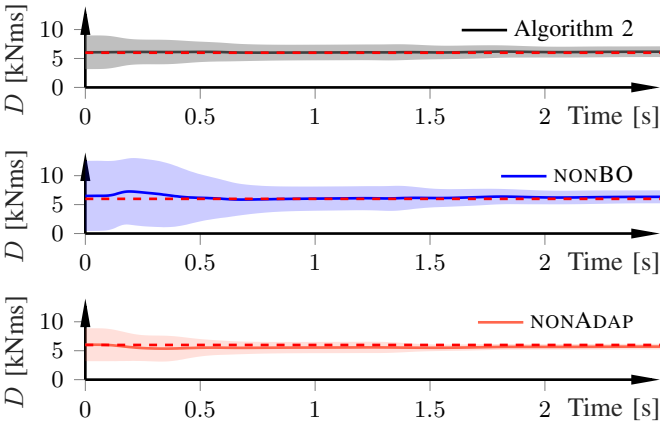

Fig. 7. Averaged estimation results of the damping coefficient D for 1000 Monte-Carlo executions for the Sine-with-dwell maneuver in Fig. 3, with 3σ shown as shaded areas. Algorithm 2 and NONBO have similar mean but the uncertainty is significantly larger for NONBO. NONADAP does not represent well the underlying value.

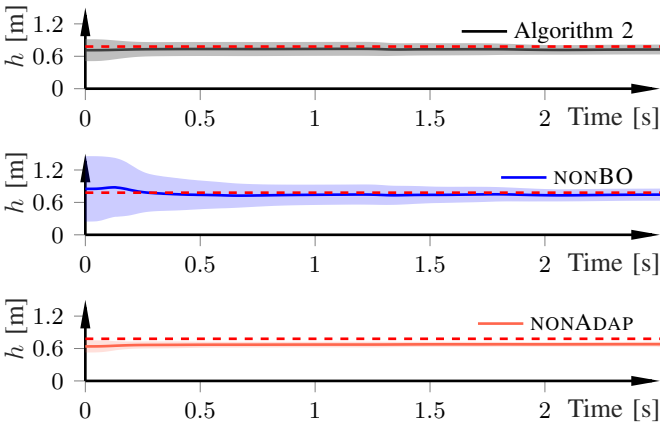

Fig. 8. Averaged estimation results of the CoG height h for 1000 Monte-Carlo executions for the Sine-with-dwell maneuver in Fig. 3, with 3σ shown as shaded areas. Algorithm 2 and NONBO have similar mean but the uncertainty is significantly larger for NONBO. NONADAP does not represent well the underlying value.

linear in the states but nonlinear in the parameters, which makes it possible to design an MPF. To avoid having to pre-set the number of particles to a very high prescribed value, we developed a BO-based procedure to provide an initial guess of the parameter distribution. This, along with a mechanism to adaptively select the number of particles during runtime, leads to an implementation of the particle filter that is both computationally efficient and gives reliable, yet not too conservative, uncertainty estimates.

REFERENCES

[1] National Highway Traffic Safety Administration, "Traffic safety facts 2004: A compilation of motor vehicle crash data from the fatality analysis reporting system and the general estimates system," NHTSA, Tech. Rep., 2006.

[2] S. Solmaz, M. Akar, R. Shorten, and J. Kalkkuhl, "Real-time multiple-model estimation of centre of gravity position in automotive vehicles," *Veh. Syst. Dyn.*, vol. 46, no. 9, pp. 763–788, 2008.

[3] A. Vahidi, A. Stefanopoulou, and H. Peng, "Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments," *Veh. Syst. Dyn.*, vol. 43, no. 1, pp. 31–55, 2005.

[4] X. Huang and J. Wang, "Real-time estimation of center of gravity position for lightweight vehicles using combined AKF–EKF method," *IEEE Trans. Veh. Technol.*, vol. 63, no. 9, pp. 4221–4231, 2014.

[5] T. B. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear nonlinear state-space models," *IEEE Trans. Signal Process.*, vol. 53, pp. 2279–2289, 2005.

[6] 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.

[7] T. Beckers, S. Bansal, C. J. Tomlin, and S. Hirche, "Closed-loop model selection for kernel-based models using bayesian optimization," in *Conf. Decision and Control*, Nice, France, 2019.

[8] A. Chakrabarty and M. Benosman, "Safe learning-based observers for unknown nonlinear systems using Bayesian optimization," *arXiv preprint arXiv:2005.05888*, 2020.

[9] U. Kiencke and L. Nielsen, *Automotive Control Systems—For Engine, Driveline and Vehicle*, 2nd ed. Berlin Heidelberg: Springer-Verlag, 2005.

[10] F. Gustafsson, *Statistical Sensor Fusion*. Lund, Sweden: Utbildning-shuset/Studentlitteratur, 2010.

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME–J. Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[12] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[13] C. K. Williams and C. E. Rasmussen, "Gaussian processes for regression," in *Advances in neural information processing systems*, 1996.

[14] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*, 2010.

[15] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012.

[16] A. Bain and D. Crisan, *Fundamentals of Stochastic Filtering*, 1st ed., ser. Stochastic Modelling and Applied Probability. Springer-Verlag New York, 2009.

[17] O. Straka and M. Šimandl, "Particle filter adaptation based on efficient sample size," in *IFAC Symp. System ID.*, 2006.

[18] V. Elvira, J. Míguez, and P. M. Djurić, "Adapting the number of particles in sequential Monte Carlo methods through an online scheme for convergence assessment," *IEEE Transactions on Signal Processing*, vol. 65, no. 7, pp. 1781–1794, 2016.

[19] K. Berntorp and S. Di Cairano, "Tire-stiffness and vehicle-state estimation based on noise-adaptive particle filtering," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 1100–1114, 2018.

[20] R. Bencatel, R. Tian, A. R. Girard, and I. Kolmanovsky, "Reference governor strategies for vehicle rollover avoidance," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 6, pp. 1954–1969, 2017.