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

Vehicle Rollover Avoidance by Parameter-Adaptive Reference Governor

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

TR2021-151 January 11, 2022

Abstract

This paper describes an approach to the vehicle rollover prevention problem that includes estimation of parameters affecting the roll dynamics and a controller accounting for uncertainties in such parameters. We develop an adaptive reference governor (ARG) that modifies the driver steering input based on satisfaction of a rollover avoidance constraint, and state and input constraints. The vehicle dynamics are highly nonlinear and has parametric uncertainties, for which the presented approach ensures rollover prevention. We design a recursive Bayesian estimator that produces confidence estimates of the parameters, including the center-of-gravity height. The confidence estimates are used to construct online constraint admissible sets, which are leveraged by the ARG to ensure rollover prevention. Simulation results on a Fishhook maneuver show that the method robustly avoids rollover prevention, and that the resulting parameter estimates are contained in the confidence sets.

IEEE Conference on Decision and Control (CDC) 2021

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

Vehicle Rollover Avoidance by Parameter-Adaptive Reference Governor

Karl Berntorp, Ankush Chakrabarty, and Stefano Di Cairano

***Abstract*— This paper describes an approach to the vehicle rollover prevention problem that includes estimation of parameters affecting the roll dynamics and a controller accounting for uncertainties in such parameter estimation. We develop a parameter-adaptive reference governor (PARG) that modifies the driver steering input to enforce a rollover avoidance constraint, and state and input constraints. We design a recursive Bayesian estimator that produces confidence estimates of the parameters, including the center-of-gravity height. The confidence estimates inform a supervised learning algorithm, which constructs online constraint admissible sets that are leveraged by the PARG to ensure rollover prevention. Simulation results on a Fishhook maneuver show that the method robustly prvents rollover, and that the resulting parameter estimates are contained in the confidence sets produced by the Bayesian estimator.**

I. Introduction

Rollover accidents are relatively uncommon but constitute a large portion of severe accidents and fatalities [1]. According to [2], usually automotive manufacturers employ robust active road-handling control strategies to account for the unknown and changing center-of-gravity (CoG) location, by designing for the worst-case scenario. Another common approach in the case of Sport Utility Vehicles (SUVs) is to intentionally design the vehicle heavier than usual by adding ballast in the undercarriage [2]. The aim is to lower the CoG height and hence reducing the variation of the CoG location. Such approaches come with drawbacks, such as performance loss under normal driving conditions and reduced efficiency.

The rollover prevention problem has been researched extensively over the last decades using different types of actuation and control architectures. Examples are optimization-based brake-force allocation [3], [4], model-predictive control (MPC) using steer by wire and differential braking [5], robust control of the load-transfer ratio (LTR) through braking [6] and active steering [7], deployment of anti-roll bars in heavy duty vehicles [8], combination of different actuation [9], and joint control of yaw and roll stability [10]. The rollover mitigation system is activated by determination when rollover is imminent. This needs to be done carefully, as rollover prevention affects the operation of the vehicle with respect to other important control objectives, such as yaw stability. In [11], activation is obtained automatically by using reference governors (RGs). The RG [12] is a lightweight add-on scheme that modifies commands to closed-loop control systems to maintain constraint satisfaction despite reference changes. Due to its relatively low computational burden, RGs are well suited for implementation in automotive microcontrollers. In [11] the LTR is a key indicator for the risk of vehicle rollover. Consequently, the RG adjusts the steering-angle command to ensure constraint satisfaction, and hence rollover avoidance, based on the LTR.

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

We propose a constraint-enforcement based rollover prevention method that includes online learning of the CoG location. The method acts as a supervisory controller and adjusts the steering-angle command based on satisfaction of a rollover avoidance constraint, and state and input constraints. Our method estimates the CoG location, and spring stiffness and damping coefficient of a spring-damper model, and accounts for the uncertainty of such parameters in the closed-loop control. We rely on a parameter-adaptive RG (PARG) [13] to avoid rollover. Our PARG includes a support vector machine (SVM) algorithm that dynamically learns robust constraint admissible positive invariant (PI) sets by combining offline data with the online parameter estimates. The parameter estimates are obtained by a computationally efficient marginalized adaptive particle filter that estimates the CoG location, spring stiffness and damping coefficient [14], as well as confidence bounds of the estimates.

We use an RG for achieving rollover avoidance. However, unlike [11], we do not assume a known CoG location, but estimate it online and design a PARG based on the estimates. Another method for robust rollover prevention was presented in [15], where the linearized vehicle model was used to design different proportional-integral controller gains for predefined bounds on the parameter uncertainties, which are then activated based on the risk of rollover. In contrast to [15], we perform online learning of robust PI sets to achieve constraint satisfaction based on the nonlinear vehicle model. Also, our method is designed to only intervene when otherwise rollover would be unavoidable, so that the activation rule is already embedded into the controller.

Notation: We denote vectors with lowercase bold font as $\boldsymbol{x}$ and matrices with uppercase bold font as $\boldsymbol{X}$. The norm ball of center c and radius ρ is $\mathcal{B}(c,\rho) = \{\boldsymbol{x} : \|\boldsymbol{x}-\boldsymbol{c}\| \leq \rho\}$. We denote a discrete time step with k. A set $\mathcal{O}$ is PI for the constrained system $\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k)$ if $\boldsymbol{x}_0 \in \mathcal{O}$ implies $\boldsymbol{x}_k \in \mathcal{O}$ for all $k > 0$. By $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{0:k})$, we denote the posterior density function of the state trajectory $\boldsymbol{x}_{0:k}$ from time index 0 to k given the measurement sequence $\boldsymbol{y}_{0:k} := \{\boldsymbol{y}_0, \dots, \boldsymbol{y}_k\}$. Throughout $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ means that $\boldsymbol{x}$ is Gaussian distributed with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

II. Modeling and Problem Formulation

For control purposes, we model the vehicle using a nonlinear chassis vehicle model describing the motion of the rigid body due to the forces generated at the tires, and

a nonlinear tire model describing the forces that the tires generate depending on the chassis and wheels velocities. The chassis model combines a single-track chassis model with a torsional spring-damper model of the roll dynamics. The resulting model is similar to established models for rollover prevention [5], [11], [15], [16]. With the longitudinal and lateral velocities, v^X, v^Y, yaw rate $\dot{\psi}$, roll angle, ϕ, and roll rate $\dot{\phi}$ as states, the resulting chassis model is described by

$$\dot{v}^X = v^Y\dot{\psi} - mh\dot{\phi}\cos(\phi) + F^X/m, \tag{1a}$$
$$\dot{v}^Y = -v^X\dot{\psi} + h(\ddot{\phi}\cos(\phi) - \dot{\phi}^2\sin(\phi)) + F^Y/m, \tag{1b}$$
$$\ddot{\psi} = M^Z/I_Z, \tag{1c}$$
$$\dot{\phi} = \dot{\phi}, \tag{1d}$$
$$\ddot{\phi} = \frac{h(F^Y + m\sin(\phi))(g + h\dot{\phi}^2) + \tau_\phi}{I_X + mh^2\cos(\phi)}, \tag{1e}$$

where

$$F^X = F_f^x\cos(\phi) - F_f^y\sin(\phi) + F_r^x, \tag{2a}$$
$$F^Y = F_f^x\sin(\phi) + F_f^y\cos(\phi) + F_r^x, \tag{2b}$$
$$M^Z = l_f(F_f^x\sin(\delta) + F_f^y\cos(\phi)) - l_rF_r^y, \tag{2c}$$

g is the gravitational acceleration, h is the distance from the roll axis to the CoG (the CoG distance), m is the vehicle mass, I_X, I_Z, are the vehicle inertias about the X- and Z-axis, respectively, and $\tau_\phi = -K_\phi\tan(\phi) - D_\phi\dot{\phi}\cos(\phi)$, K_ϕ is the spring stiffness and D_ϕ is the damping coefficient. We use the Pacejka tire (Magic Formula) model [17]. To model combined slip we employ similar modeling as in [11], [16],

$$\begin{aligned}\begin{bmatrix}F_i^x & F_i^y\end{bmatrix}^\top &= F_P P\hat{\boldsymbol{s}},\\ P &= \sin(C\text{atan}(s_c/C(1-E) + E\text{atan}(s_c/C)))),\\ s_c &= \frac{C_\alpha\|\boldsymbol{s}\|}{F_P}, \quad C_\alpha = c_1mg(1 - e^{-c_2z_i^Z/(mg)}),\\ c_1 &= \frac{BCD}{4(1-e^{-c_2/4})}, \quad F_P = \frac{F_i^z 1.0527}{1 + (\frac{1.5F_i^z}{mg})^3},\\ \boldsymbol{s} &= \begin{bmatrix}\lambda_i\\ \tan\alpha_i\end{bmatrix}, \quad \hat{\boldsymbol{s}} = \boldsymbol{s}\ \|\boldsymbol{s}\|\end{aligned} \tag{3}$$

where α_i are the slip angles, λ_i are the slip ratios, F_i^z is the normal forces resting on wheel i, $F_f^z = mgl_r/l$, $F_r^z = mgl_f/l$, C_α is the cornering stiffness, and B, C, D, E, and c_2 are tire and road-specific parameters. The slip angles α_i and slip ratios λ_i in (3) are

$$\alpha_i = -\arctan\left(\frac{v_i^y}{v_i^x}\right),\ \lambda_i = \frac{R_w\omega_i - v_i^x}{v_i^x},\ i \in \{f, r\}, \tag{4}$$

where R_w is the wheel radius, ω_i is the wheel angular velocity for wheel i, and v_i^x and v_i^y are the longitudinal and lateral wheel velocities for wheel i in the coordinate system of the respective wheel.

A. Problem Formulation

We consider a setup where a reference wheel steering angle δ_r, determined either by a human driver or an advanced driver-assistance system. The control input is the commanded wheel steering angle. The objective is to design a control method that makes the vehicle avoid rollover while satisfying state and input constraints and tracking the desired reference δ_r as closely as possible. This can be achieved by controlling directly the ßsteering angle, for example, by a steer-by-wire system, or by modifying the current steering angle using various actuators, such as active front steering and electronic stability control. Since the CoG distance h is unknown and varies with the type of vehicle and loading conditions. Fig. 1 shows a schematic of the proposed control strategy. The estimator may use multiple algorithms to estimate the vehicle state $\boldsymbol{x}$ used for control, the spring stiffness K_ϕ, damping coefficient D_ϕ, and CoG distance h, from the vehicle measurements ($\boldsymbol{y}$) . In this paper we present a marginalized particle filter (MPF) that estimates $\boldsymbol{\theta} = (K_\phi, D_\phi, h)$, roll angle ϕ and roll rate $\dot{\phi}$, but there are readily available estimation methods for determining the additional states in $\boldsymbol{x}$ using automotive-grade sensors (e.g., [18]). The estimates of $\boldsymbol{x}$ and $\boldsymbol{\theta}$ are sent to the PARG, which adjusts the steering reference δ_r as needed to produce an adjusted δ_c that avoids rollover. In the following, we set $u = \delta_c$ to emphasize that the steering command is the control input.

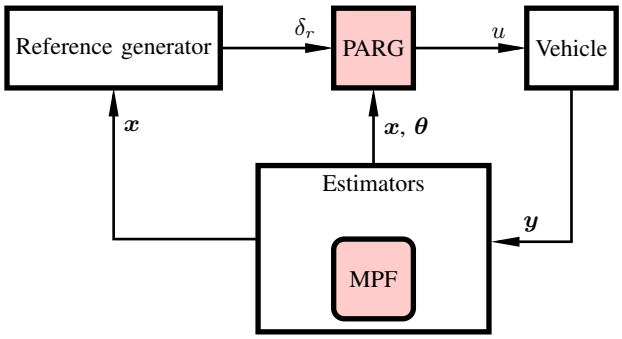

Fig. 1. The proposed control architecture. PARG uses statistics of $\boldsymbol{\theta}$ estimates from the proposed MPF to ensure that the adaptation u of the reference δ_r leads to rollover avoidance. The reference generator can be either a human driver or an ADAS, and the estimator block can, and typically will, include multiple estimators.

III. Learning-Based PARG

After discretization of the vehicle dynamics model consisting of (1)–(4) with sampling period T_s, we can write the resulting model compactly as the nonlinear system

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, u_k) + \boldsymbol{g}(\boldsymbol{x}_k, u_k, \boldsymbol{\theta}_k). \tag{5}$$

A. State and Input Constraints

We enforce both state and input constraints, as well as a rollover avoidance constraint. The state and input constraints are formulated as box constraints,

$$\mathbb{X} = \{\boldsymbol{x} : \boldsymbol{x}_{\min} \le \boldsymbol{x} \le \boldsymbol{x}_{\max}\}, \tag{6a}$$
$$\mathbb{U} = \{u : -\delta_{\lim} \le u \le \delta_{\lim}\}. \tag{6b}$$

The symmetric input constraint is based on the maximum allowed road wheel steering angle. The state constraints are determined based on a tradeoff between safety limits and driving comfort. Following previous work in rollover

avoidance [3], [7], [11], we define the LTR as

$$LTR = \frac{F_L^z - F_R^z}{mg}, \tag{7}$$

where F_L^z, F_R^z, is the left and right vertical load, respectively. The LTR measures the relative load on each side of the vehicle and wheel liftoff occurs when either $LTR > 1$ or $LTR < -1$.[1] The LTR is a function of the roll angle and roll rate and can be calculated from the lateral load transfer [19]. Consequently, we enforce the symmetric LTR constraint

$$\mathbb{Y} = \{LTR : -LTR_{\lim} \leq LTR \leq LTR_{\lim}\}, \tag{8}$$

where $-LTR_{\lim} < 1$ is added as a safety margin. We denote the set of constraints including (6) and (7) by

$$\mathcal{H} = \{(\boldsymbol{x}, u, LTR) : \boldsymbol{x} \in \mathcal{X}, u \in \mathcal{U}, LTR \in \mathcal{Y}\} \tag{9}$$

Remark 1. *The LTR does not capture the transient phase of the rollover, since it is derived without accounting for the roll dynamics [6]. Still, it is the most commonly used indication of rollover and for that reason we also employ it here.*

B. PARG Objective

The objective of an RG is to select a control command u_k as close as possible to a reference r_k while ensuring that constraints are enforced.The commonly treated cases are when the parameters $\boldsymbol{\theta}$ for the considered system is known or when the range is known [12]. In this paper, we consider the case when the parameter vector $\boldsymbol{\theta}$ is unknown, but constant [13]. The PARG we employ in this paper is given by

$$\begin{aligned} u_k &= \bar{\mathcal{G}}(u_{k-1}, \boldsymbol{x}_k, \hat{\boldsymbol{\Theta}}_k, \delta_{r,k}) \\ &= u_{k-1} + \mathcal{G}(u_{k-1}, \boldsymbol{x}_k, \hat{\boldsymbol{\Theta}}_k, \delta_{r,k})(\delta_{r,k} - u_{k-1}), \end{aligned} \tag{10}$$

where $\hat{\boldsymbol{\Theta}}$ is a bounded interval of parameter values, computed by the MPF in the form of confidence intervals such that $\hat{\boldsymbol{\theta}}_k \in \hat{\boldsymbol{\Theta}}_k$ with high certainty. The PARG designs $\mathcal{G}$ such that the closed-loop vehicle model (5) and (10) in combination with the MPF confidence estimate satisfies state and input constraints and tracks the reference δ_r as closely as possible. To enforce constraints under parameter uncertainty, the PARG estimates parameter-robust PI sets for designing the control law $\mathcal{G}$ in (10).

Definition 1. *The set $\mathcal{O}(\hat{\boldsymbol{\Theta}}) \subset \mathcal{H}$ is a parameter-robust PI set for the constrained system* (1) *if, for every initial condition $(\boldsymbol{x}, u) \in \mathcal{O}(\hat{\boldsymbol{\Theta}})$, when $\boldsymbol{x}_0 = \boldsymbol{x}$ and $\delta_0 = u$ for all $k \geq 0$, $(\boldsymbol{x}_k, u_k) \in \mathcal{H}$ for every $\boldsymbol{\theta} \in \hat{\boldsymbol{\Theta}}$ and for all $k > 0$.*

Subsequently, an estimate of a parameter-robust PI set can be used to evaluate the control law (10) by solving

$$\begin{aligned} \mathcal{G}(u_{k-1}, \boldsymbol{x}_k, \hat{\boldsymbol{\Theta}}, \delta_{r,k}) := \quad & \arg\min_{\gamma_t} \ (u_k - \delta_{r,k})^2 && \text{(11a)} \\ \text{subject to} \quad & (u_k, \boldsymbol{x}_k) \in \mathcal{O}(\hat{\boldsymbol{\Theta}}_k), && \text{(11b)} \\ & u_k = u_{k-1} + \gamma_k(\delta_{r,k} - u_{k-1}), && \text{(11c)} \\ & 0 \leq \gamma_k \leq 1, && \text{(11d)} \\ & u_k \in \mathbb{U}_\varepsilon(\hat{\boldsymbol{\Theta}}) && \text{(11e)} \end{aligned}$$

[1]There are some exceptions to this rule, see, e.g., Remark 1 in [11].

at each time step k, where $\mathbb{U}_\varepsilon(\hat{\boldsymbol{\Theta}})$ is the set of references u such that a ball of radius $\varepsilon > 0$ centered at the corresponding steady state $\boldsymbol{x}^{\text{ss}}(u, \boldsymbol{\theta})$ and u lies inside $\mathcal{O}(\hat{\boldsymbol{\Theta}}_k)$,

$$\mathbb{U}_\varepsilon(\hat{\boldsymbol{\Theta}}) \triangleq \left\{ u \in \mathbb{U} : \mathcal{B}_\varepsilon(\boldsymbol{x}^{\text{ss}}(u, \boldsymbol{\theta}), u) \subset \mathcal{O}(\hat{\boldsymbol{\Theta}}_k), \forall\, \theta \in \hat{\Theta} \right\}.$$

IV. Vehicle Rollover Avoidance by Parameter-Adaptive Reference Governor

In this section we describe our control strategy for rollover avoidance, which includes estimating the parameter-robust PI set $\mathcal{O}(\hat{\boldsymbol{\Theta}}_k)$ at each time step k, and subsequently solving (11) to generate an adjusted reference u_k according to (10). Our approach consists of three steps.

1) Offline, we simulate the closed-loop system (5) from different initial states, reference inputs, and parameters, and we classify the trajectories based on whether or not they satisfy the constraints.
2) Online, we recursively estimate the confidence interval $\hat{\boldsymbol{\Theta}}_k$.
3) Then, still online, using the estimate $\hat{\boldsymbol{\Theta}}_k$ we determine the parameter-robust PI set and solve (11).

Next, we go through these three steps and summarize the resulting algorithm.

A. Offline Data Generation

We simulate trajectories of the closed-loop system (5) offline, from different initial states sampled from $\mathbb{X}$, steering angle references sampled from $\mathbb{U}$, and parameters within $\boldsymbol{\Theta}$. Note that $\boldsymbol{\Theta}$ is a priori determined based on physical reasoning about the parameters, and the bounds defining $\boldsymbol{\Theta}$ can be large, as the online estimation of $\hat{\boldsymbol{\Theta}} \subset \boldsymbol{\Theta}$ takes care of reducing conservativeness. We extract N_x unique samples from $\mathbb{X}$ and construct grids on $\mathbb{U}$ and $\boldsymbol{\Theta}$ with N_u and N_θ nodes, respectively. Let $\boldsymbol{x}^i$ denote the i-th sampled state, u^j the j-th sampled wheel steering angle command, and $\boldsymbol{\theta}^m$ the m-th sampled parameter. For each $(\boldsymbol{x}^i, u^j, \boldsymbol{\theta}^m)$, we simulate the closed-loop system (5) forward in time over a finite horizon T_h with a constant reference u^j and parameter $\boldsymbol{\theta}^m$. The horizon T_h is chosen long enough such that the tracking error is small enough by the end of the simulation. For each simulation, we check whether $LTR_k \in \mathbb{Y}$ for every time step of the simulation. We set the corresponding label of the sample $\boldsymbol{x}^i$ as

$$\ell^{i,j,m} = \begin{cases} +1, & \text{if } LTR_k \in \mathbb{Y} \text{ for every } k \in \{0, 1, \ldots, T_s\}, \\ -1, & \text{otherwise.} \end{cases} \tag{12}$$

At the end of the offline data generation, we have a fixed collection of initial $\{x_0^i\}_{i=1}^{N_x}$, and each initial condition x_0^i has a corresponding $N_v \times N_\theta$ matrix of labels

$$\boldsymbol{\ell}^i = \begin{bmatrix} \ell^{i,1,1} & \ldots & \ell^{i,1,N_\theta} \\ \vdots & \ddots & \vdots \\ \ell^{i,N_v,1} & \ldots & \ell^{i,N_v,N_\theta} \end{bmatrix}, \tag{13}$$

from which a labeled set is generated online. Note that from (12), every element in $\boldsymbol{\ell}^i$ is either $+1$ or -1.

B. Online Determining the Parameter Uncertainty Bounds

For determining the estimates and associated uncertainties of the suspension stiffness, damping coefficient, and CoG distance, all contained in $\boldsymbol{\theta}$, we rely on a recently developed MPF [14]. Next, we briefly outline the estimation model and the algorithm formulation, and refer to [14] for a more comprehensive treatment.

1) Estimation Model: The MPF employs the roll dynamics model given by

$$(I_X+mh^2)\ddot{\phi}+D_\phi\dot{\phi}+K_\phi\phi = mh(a_y\cos\phi+g\sin\phi). \quad (14)$$

If the vehicle is not close to rollover, $\sin\phi \approx \phi$ and $\cos\phi \approx 1$, and we can write the system in state-space form with state $\boldsymbol{x}^e = \begin{bmatrix}\phi & \dot{\phi}\end{bmatrix}^\top$

$$\dot{\boldsymbol{x}}^e = \begin{bmatrix} 0 & 1 \\ -\frac{K_\phi - mgh}{I_x+mh^2} & -\frac{D_\phi}{I_x+mh^2}\end{bmatrix}\boldsymbol{x}^e + \begin{bmatrix}0 \\ \frac{mh}{I_x+mh^2}\end{bmatrix} a_y. \quad (15)$$

First, note that (15) is linear in $\boldsymbol{x}^e$ but nonlinear in the parameters. Second, all parameters are time varying in the sense that they change depending on the loading conditions, but when the vehicle is moving, they are unlikely to have large variations. Thus, it is appropriate to model the parameters as

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{w}_{\boldsymbol{\theta},k}, \quad (16)$$

where $\boldsymbol{w}_{\boldsymbol{\theta},k}$ is 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 $\boldsymbol{\theta}_0 \sim p_0(\boldsymbol{\theta})$. After zero-order hold sampling of (15) and combining with (16), the estimation model becomes

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{w}_{\boldsymbol{\theta},k}, \quad (17a)$$
$$\boldsymbol{x}^e_{k+1} = \boldsymbol{A}(\boldsymbol{\theta}_k)\boldsymbol{x}^e_k + \boldsymbol{B}(\boldsymbol{\theta}_k)a_y, \quad (17b)$$
$$\boldsymbol{y}_k = \boldsymbol{x}^e_k + \boldsymbol{e}_k, \quad (17c)$$

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

2) Formulation of the CoG Estimation Problem: The state is measured in (17c) but the observation of $\boldsymbol{\theta}$ is implicit through the roll-dynamics model (17b). Hence, to estimate $\boldsymbol{\theta}$ we also estimate $\boldsymbol{x}^e$. Since (17) is linear in the state and nonlinear in the parameters, a suitable estimation algorithm is rooted in the MPF, where we can exploit linearity with respect to the vehicle state to give a semi-analytic estimator.

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, or Rao-Blackwellization, where the subset of the state space that allows for analytic expressions is marginalized out. The sampled state space is then smaller and it is therefore possible to use fewer particles, which is key in applications with stringent computational and timing requirements. The enabler for the MPF in [14] is the factorization

$$p(\boldsymbol{x}^e_k, \boldsymbol{\theta}_{0:k}|\boldsymbol{y}_{0:k}) = p(\boldsymbol{x}^e_k|\boldsymbol{\theta}_{0:k}, \boldsymbol{y}_{0:k})p(\boldsymbol{\theta}_{0:k}|\boldsymbol{y}_{0:k}) \quad (18)$$

The second distribution in (18) is approximated by the PF. Conditioned on the nonlinear state trajectory, the first factor on the right-hand side in (18) is linear Gaussian. Thus, it can be estimated with conditional KFs, one for each particle conditioned on the particle trajectory. The main difference compared with the standard KF consists of performing an extra measurement update for each KF using the forward propagated $\boldsymbol{x}^{e,i}_{k+1}$ as the extra measurement. From the particle distribution that generates (18), we can construct confidence bounds by numerical integration. Alternatively, a simplification can be made to estimating the confidenece bounds by using first and second moments.

For the PARG constraint satisfaction property to hold, we need to ensure that our confidence intervals do not expand with more available data, that is, $\hat{\boldsymbol{\Theta}}_{k+1} \subseteq \hat{\boldsymbol{\Theta}}_k$, which is not guaranteed for the MPF and needs to be checked during runtime. Hence, we explicitly enforce nonexpansion of confidence intervals. Specifically, if the filter computes an updated confidence interval $\tilde{\boldsymbol{\Theta}}_{k+1}$, we set

$$\hat{\boldsymbol{\Theta}}_{k+1} := \begin{cases} \hat{\boldsymbol{\Theta}}_k \cap \tilde{\boldsymbol{\Theta}}_{k+1} & \text{if } \hat{\boldsymbol{\Theta}}_{k+1} \cap \tilde{\boldsymbol{\Theta}}_k \neq \emptyset \\ \hat{\boldsymbol{\Theta}}_k, & \text{otherwise} \end{cases} \quad (19)$$

to ensure nonexpansion of $\hat{\boldsymbol{\Theta}}_k$ for all $k \geq 0$.

C. Online Determining Parameter-Robust PI Sets

For enabling online learning of the robust PI sets and subsequent solution of (11), we grid the input constraint set $\mathbb{U}$ and combine with machine learning [13]. Gridding the input constraint set $\mathbb{U}$ in (6b), along with the constraints (11c) and (11d), implies that the solution to (11) is contained within the sub-grid of $\mathbb{U}$ defined by

$$\tilde{\mathbb{U}}_k := \big\{\min\{\delta_{r,k}, u_{k-1}\}, \ldots, \max\{\delta_{r,k}, u_{k-1}\}\big\}, \quad (20)$$

with $|\tilde{\mathbb{U}}_k| = J$, where J is the number of grid points of u. We solve (11) approximately as the grid-search problem

$$u_k := \quad \arg\min_{u\in\tilde{\mathbb{U}}_k} (u-\delta_{r,k})^2 \quad (21a)$$
$$\text{subject to} \quad (u, \boldsymbol{x}_k) \in \mathcal{O}(\hat{\boldsymbol{\Theta}}_k) \quad (21b)$$
$$u \in \mathbb{U}_\varepsilon(\hat{\boldsymbol{\Theta}}_k) \quad (21c)$$

To learn the parameter-robust PI set $\mathcal{O}(\hat{\boldsymbol{\Theta}}_k)$, for each $u^j \in \tilde{\mathbb{U}}_k$ described in (20), and each $\boldsymbol{x}^i \in \{\boldsymbol{x}^i\}_{i=1}^{N_x}$ sampled offline according to Sec. IV-A, we assign the label

$$z^{i,j}(\hat{\boldsymbol{\Theta}}_k) = \min_{m\in\mathcal{I}_{i,j}(\hat{\boldsymbol{\Theta}}_k)} \ell_i^{j,m}, \quad (22)$$

where $\mathcal{I}^{i,j}(\hat{\boldsymbol{\Theta}}_k) := \Big\{m : \boldsymbol{\theta}^m \in \hat{\boldsymbol{\Theta}}_k\Big\}$ is the index set of parameters contained in the current confidence interval $\hat{\boldsymbol{\Theta}}_k$. With the training data $D := \{(\boldsymbol{x}^i, u^j), z^{i,j}\}$, we construct classifiers φ^j, where $j = 1, \ldots, J$. For each u^j, a classifier is trained on features $\{\boldsymbol{x}^i\}$ and their corresponding labels $\{z^{i,j}\}$. These classifiers need to be inner approximations of the robust PI sets. We achieve this by selecting sublevel sets of the decision boundary $\varphi^m = 0$ of the classifier until no infeasible sample is contained in the interior of the sublevel set [21]. Solving (21) is then identical to selecting the node u^j on the grid $\tilde{\mathbb{U}}_k$ that minimizes the cost (21a),

while ensuring that $\varphi^j(\boldsymbol{x}_k) > 0$; that is, the current state is predicted by the j-th classifier to belong to the parameter-robust PI set induced by $\hat{\boldsymbol{\Theta}}_k$. In this paper, we use a 2-norm soft margin SVM classifier (c.f. [13]) trained on D.

The proposed parameter-adaptive rollover avoidance control strategy is summarized in Algorithm 1.

Algorithm 1 Proposed rollover-avoidance control strategy

Offline: Simulate closed-loop trajectories according to Sec. IV-A to get labeled data matrix (13).
Online:
while true **do**
  Estimate the parameter mean $\hat{\boldsymbol{\theta}}_k$ and distribution $p(\boldsymbol{\theta}_k|\boldsymbol{y}_{0:k})$ using the approach in [14].
  From $\hat{\boldsymbol{\theta}}_k$ and distribution $p(\boldsymbol{\theta}_k|\boldsymbol{y}_{0:k})$, determine confidence interval $\hat{\boldsymbol{\Theta}}_k$ (i.e., (19)).
  Determine admissible steering angles $\tilde{\mathbb{U}}_k$ with (20).
  for each $i = 1, \ldots, N_x$ **do**
    for each $j = 1, \ldots, |\tilde{\mathbb{U}}_k| < N_u$ **do**
      Assign label $z^{i,j}$ according to (22).
    end for
  end for
  Construct classifiers φ^j, $j \in \{1, \ldots, , |\tilde{\mathbb{U}}_k|\}$.
  Solve the grid-search problem (21) and apply u_k.
  $k \leftarrow k + 1$
end while

V. Simulation Results

In the simulation study we use the Fishhook maneuver, which is a vehicle maneuver standardized by NHTSA and commonly used for evaluating roll stability. We keep the vehicle velocity when entering the maneuver to 80km/h. Here, the only actuation is the wheel steering angle. The PARG prediction model is the nonlinear vehicle model described by (1)–(4) that includes lateral and longitudinal dynamics, nonlinear tire forces including combined slip, and suspension dynamics, which includes both sprung and unsprung mass models. The estimation model, however, is the relatively simple roll-dynamics model (17), 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 [11], where according to the authors the parameters are set to be similar to a North-American SUV. The estimator tuning parameters are similar as in [14] and the initial estimates of the parameters $\{K_\phi, D_\phi, h\}$ are sampled from the uncertainty interval

$$[30000, 80000] \times [3000, 10000] \times [0.5, 1.2]. \qquad (23)$$

We extract 5000 samples drawn from Halton sequences on $\mathbb{X}$, and construct equispaced grids on $\mathbb{U}$ and Θ with 100 and 150 samples, respectively.

For evaluation, we have executed 100 Monte-Carlo runs of the Fishhook maneuver with different values of $\boldsymbol{\theta}$ for each run, sampled as a uniform distribution with variation of $\pm 5\%$ around the mean value. However, in the training, we have used the mean value of the parameters. This setup therefore indicates the robustness of the proposed method to imperfect training data. The compared controllers are

1) PARG: The proposed control strategy in Algorithm 1.
2) RRG: A robust RG that uses the uncertainty interval (23) to offline construct the PI set.
3) NOMINAL: A controller that uses the reference command directly to control the vehicle.

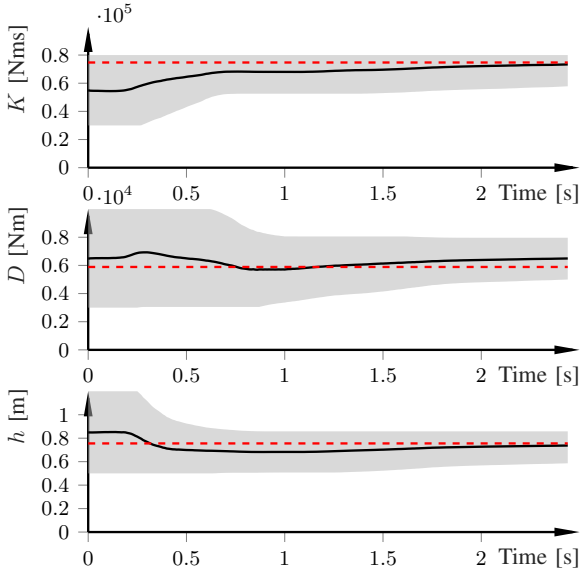

Fig. 2. Parameter estimation results for one Monte-Carlo execution for the Fishhook maneuver. True values in red dashed, estimates in black, and 2σ area in gray.

Fig. 2 shows the closed-loop estimation results for one representative realization. After the initial transients, the spring stiffness K, damping coefficient D, and CoG distance h all converge close to their respective true values. The estimates are contained within the confidence intervals, which are tightening with time as more data are gathered.

The corresponding closed-loop constraint satisfaction and resulting steering angle command for the same realization as in Fig. 2 are shown in Fig. 3. For the online calculation of the PI sets, we employ a nonlinear SVM with radial basis function kernel. The SVM penalizes false positives more strongly than false negatives through an asymmetric cost, and constraints on SVM coefficients are removed. Small coefficients are clipped, and cross-validation yields best regularization parameters. The proposed PARG satisfies the constraints, as it adjusts the steering angle. Using the nominal reference steering angle, there are numerous large constraint violations, indicating wheel liftoff.

Next, we compare the proposed PARG with the nominal steering profile (NOMINAL) and the robust RG (RRG). As performance metric, we use the number of constraint violations and the quadratic cost in (21a),

$$\text{Cost} = \sum_k (u_k - \delta_{r,k})^2, \qquad (24)$$

which both PARG and RRG aim to minimize. Note that the offline simulation in Algorithm 1 to get the labeled matrix (13) is done for one fixed value of the parameters and is not aware of the different parameter values.

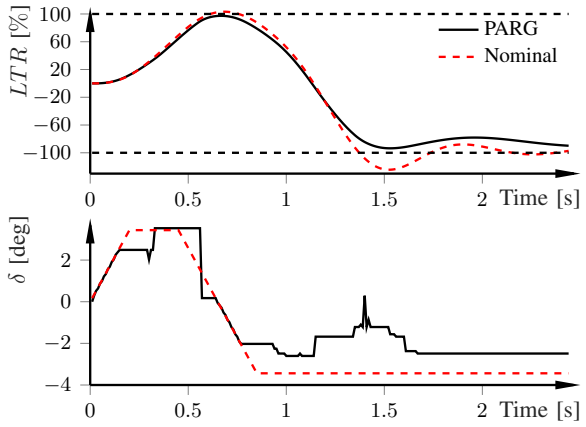

Fig. 3. Closed-loop control for the resulting LTR and the adjusted steering angle for the estimation results in Fig. 2. The PARG satisfies the LTR constraint, whereas the nominal steering reference violates the constraints.

TABLE I

RESULTS FOR 100 MONTE-CARLO RUNS.

	PARG	RRG	NOMINAL
# runs $\|LTR\| > 1$	**0/100**	**0/100**	100/100
Mean Cost	**0.130**	0.135	-
Std Cost	**0.010**	0.017	-

Table I displays a summary of the results, where we show the number of Monte-Carlo runs with constraint violation, the cost (24) averaged over the Monte-Carlo runs (Mean Cost), and the standard deviation of the cost over the Monte-Carlo runs (Std Cost). Both the proposed rollover-avoidance strategy and RRG succeed in providing constraint satisfaction in all of the Monte-Carlo runs. However, the nominal command violates multiple times the constraints at every Monte-Carlo run, as indicated in Fig. 3. PARG produces a slightly lower cost (3%) than RRG on average. The real benefit is seen from the variation of the cost over runs, where the proposed adaptation to the estimated parameters reduce the standard deviation (Std Cost) of the cost with 60%.

VI. CONCLUSION

We presented a rollover avoidance control strategy based on a parameter-adaptive reference governor that avoids rollover, even when having uncertainties in the parameters related to the suspension dynamcis, for example, the CoG height. The method combines machine learning with marginalized particle filtering for determining online the robust positive invariant sets that ensure rollover avoidance.

The simulation results show that the method effectively avoids wheel liftoff according to the LTR metric and that while the average cost is a few percent, the reduction in performance variation is 60% compared to a robust reference governor implementation. The Monte-Carlo study when varying the true parameters indicates that the method is robust not only to uncertain parameters, but also avoids rollover when the simulation model we use for learning is uncertain. The next step is to verify the real-time applicability of the method in high-fidelity hardware-in-the-loop simulations.

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] B. Schofield and T. Hägglund, "Optimal control allocation in vehicle dynamics control for rollover mitigation," in *Amer. Control Conf.*, Seattle, WA, Jun. 2008.

[4] M. B. Alberding, J. Tjønnås, and T. A. Johansen, "Integration of vehicle yaw stabilisation and rollover prevention through nonlinear hierarchical control allocation," *Veh. Syst. Dyn.*, vol. 52, no. 12, pp. 1607–1621, 2014.

[5] C. R. Carlson and J. C. Gerdes, "Optimal rollover prevention with steer by wire and differential braking," in *ASME International Mechanical Engineering Congress and Exposition*, 2003.

[6] S. Solmaz, M. Corless, and R. Shorten, "A methodology for the design of robust rollover prevention controllers for automotive vehicles: Part 1-differential braking," in *Conf. Decision and Control*, San Diego, CA, Dec. 2006.

[7] ——, "A methodology for the design of robust rollover prevention controllers for automotive vehicles: Part 2-active steering," in *Amer. Control Conf.*, New York City, NY, Jul. 2007.

[8] H. Yu, L. Güvenc, and Ü. Özgüner, "Heavy duty vehicle rollover detection and active roll control," *Veh. Syst. Dyn.*, vol. 46, no. 6, pp. 451–470, 2008.

[9] A. Y. Lee, "Coordinated control of steering and anti-roll bars to alter vehicle rollover tendencies," *J. Dyn. Sys., Meas., Control*, vol. 124, no. 1, pp. 127–132, 2002.

[10] R. Rajamani and D. N. Piyabongkarn, "New paradigms for the integration of yaw stability and rollover prevention functions in vehicle stability control," *IEEE Trans. Intell. Transport. Syst.*, vol. 14, no. 1, pp. 249–261, 2012.

[11] R. Bencatel, R. Tian, A. R. Girard, and I. Kolmanovsky, "Reference governor strategies for vehicle rollover avoidance," *IEEE Trans. Control Syst. Techn.*, vol. 26, no. 6, pp. 1954–1969, 2017.

[12] E. Garone, S. Di Cairano, and I. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 100, no. 75, pp. 306–328, 2017.

[13] A. Chakrabarty, K. Berntorp, and S. Di Cairano, "Learning-based parameter-adaptive reference governors," in *Amer. Control Conf.*, Denver, Colorado, Jul. 2020.

[14] K. Berntorp, A. Chakrabarty, and S. Di Cairano, "Vehicle center-of-gravity height and dynamics estimation with uncertainty quantification by marginalized particle filter," in *Amer. Control Conf.*, New Orleans, LA, May 2021, acccepted.

[15] S. Solmaz, M. Corless, and R. Shorten, "A methodology for the design of robust rollover prevention controllers for automotive vehicles with active steering," *Int. J. Control*, vol. 80, no. 11, pp. 1763–1779, 2007.

[16] J. Zhou, "Active safety measures for vehicles involved in light vehicle-to-vehicle impacts," Ph.D. dissertation, Univ. Michigan, 2009.

[17] H. B. Pacejka, *Tire and Vehicle Dynamics*, 2nd ed. Oxford, United Kingdom: Butterworth-Heinemann, 2006.

[18] 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.

[19] K. Berntorp, B. Olofsson, K. Lundahl, and L. Nielsen, "Models and methodology for optimal trajectory generation in safety-critical road–vehicle manoeuvres," *Veh. Syst. Dyn.*, vol. 52, no. 10, pp. 1304–1332, 2014.

[20] 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.

[21] A. Chakrabarty, V. Dinh, M. J. Corless, A. E. Rundell, S. H. Żak, and G. T. Buzzard, "Support vector machine informed explicit nonlinear model predictive control using low-discrepancy sequences," *IEEE Trans. Autom. Control*, vol. 62, no. 1, pp. 135–148, 2016.