Parameter-Adaptive Reference Governors with Learned Robust Constraint-Admissible Sets

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

TR2023-005 February 11, 2023

Abstract

Reference governors (RGs) provide an effective method for ensuring safety via constraint enforcement in closed-loop nonlinear control systems. When the system parameters are uncertain but constant, robust formulations of RGs that consider only the worst-case effect may be overly conservative and exhibit poor performance. This paper proposes a parameter-adaptive reference governor (PARG) architecture that is capable of generating safe trajectories in spite of parameter uncertainties, without being as conservative as robust RGs. The proposed approach employs machine learning on a combination of off-line simulations and on-line measurements to estimate parameter-robust constraint-admissible sets (PRCASs) that can be leveraged by the PARG. We illustrate the robust set learning and constraint enforcement qualities of the PARG using a two-dimensional electromagnetic actuator example, and further demonstrate the potential of the PARG on a vehicle case study for preventing rollover despite aggressive maneuvering.

Control Engineering Practice 2023

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

Parameter-Adaptive Reference Governors with Learned Robust Constraint-Admissible Sets

Ankush Chakrabarty[a,1], Karl Berntorp[a], Stefano Di Cairano[a]

[a]*Mitsubishi Electric Research Laboratories, Cambridge, MA, USA*

Abstract

Reference governors (RGs) provide an effective method for ensuring safety via constraint enforcement in closed-loop nonlinear control systems. When the system parameters are uncertain but constant, robust formulations of RGs that consider only the worst-case effect may be overly conservative and exhibit poor performance. This paper proposes a parameter-adaptive reference governor (PARG) architecture that is capable of generating safe trajectories in spite of parameter uncertainties, without being as conservative as robust RGs. The proposed approach employs machine learning on a combination of off-line simulations and on-line measurements to estimate parameter-robust constraint-admissible sets (PRCASs) that can be leveraged by the PARG. We illustrate the robust set learning and constraint enforcement qualities of the PARG using a two-dimensional electromagnetic actuator example, and further demonstrate the potential of the PARG on a vehicle case study for preventing rollover despite aggressive maneuvering.

Keywords: Constrained control, uncertain systems, learning, safe control systems, statistical estimation.

1. Introduction

Reference governors (RGs) are add-on schemes for enforcing pointwise-in-time input and output constraints by modifying the reference input to closed-loop controlled systems [1]. For ensuring repeated constraint satisfaction, RGs leverage particular invariant sets known as constraint admissible sets [2]. Due to their ability of enforcing constraints without requiring a full re-design, and with a relatively low runtime computational burden, RGs have proven useful in multiple application domains; see [3] for an extensive survey.

In particular, reference governance has been applied to the management of large-scale water networks [4], missile guidance [5], and safe handling of vehicles [6], and has been reported to demonstrated excellent performance. The majority of the RG literature focuses on the case where the system dynamics are known, although possibly subject to set-bounded disturbances, but the design of RGs with uncertain system dynamics has been gaining attention. Some recent approaches focus on completely unknown system dynamics, and hence apply black-box safe learning methods [7]. While of very general applicability, black-box approaches often require a significantly larger amount of data than gray-box approaches that estimate specific unknown parameters of structurally known models. When the parameters of the underlying systems are uncertain, robust formulations of RG may be applied based on robust set invariance [8, 9]. On the other hand, such robust designs are made to tackle any uncertainty within an admissible set of uncertain parameters, and as such, if the unknown parameters are constant, these robust designs may be overly conservative.

In spite of many real-world applications comprising piece-wise constant unknown parameters, there are few designs for RGs tailored specifically to this case. To the best of our knowledge, the load governor approach [10] and the neuro-dynamic programming RG [11] are the only RG formulations in the literature that are capable of handling uncertain parameters. However, both those works focus on a single system and are not demonstrated to be general methods for a wide class of methods, which is a distinction from our work. A major reason for the dearth of PARG frameworks is that the computation of robust constraint admissible sets under parameter uncertainty is difficult (even for linear systems) due to complex set geometries and inherent non-convexity of PARG optimization problems. Although these challenges can be somewhat circumvented by local linearization [10], such an approximation may still be conservative and requires online simulation and constraint satisfaction checking of multiple trajectories for various references and unknown parameter values, all of which lead to a possibly significant increase in computational load [12].

Recently proposed sampling-driven supervised machine learning tools can be exploited to efficiently estimate robust constraint admissible sets on-line [13] by off-loading simulation and trajectory generation off-line. However,

Email addresses: achakrabarty@ieee.org (Ankush Chakrabarty), karl.o.berntorp@ieee.org (Karl Berntorp), dicairano@merl.com (Stefano Di Cairano)

[1] Corresponding author. *Phone*: +1 (617) 758-6175.

for constructing robust constraint admissible sets, besides the parameter estimate, the region of the parameter space where the true parameter lies with high probability is also needed. For linear systems with unknown parameters, Kalman filter approaches have been widely reported to be effective [14, 15, 16]. For general nonlinear systems, particle filtering provides an effective solution [17] for this task. Concretely, the state is estimated by predicting state trajectories via particles and weighing them according to the likelihood of the measurements: this has also been extended to parameter estimation [18, 19]. A further extension to the case of dependency between process and measurement noise is discussed in [19]. When recursive parameter estimators are employed, the confidence region of the parameters is time-varying, and hence the robust constraint admissible set needs to be determined by solving a dynamic supervised learning problem. Specifically, for a fixed set of samples, the time-varying confidence intervals result in time-varying labeled/target sets for the learner. Hence, the learner needs to be updated on-line in order to provide improved estimates of the robust constraint admissible sets based on the time-varying confidence intervals.

In this paper, we describe a learning-based PARG that enforces constraints in parameter-uncertain closed-loop systems. Our proposed PARG leverages sampling-based approaches to generate features and labels off-line, and uses these features/labels, along with on-line data used to improve our estimates of the unknown parameters, in order to iteratively update the robust constraint admissible sets on-line. Consequently, these updated sets are used to compute feasible reference inputs. For on-line execution of our PARG, we employ two additional components over a standard RG. The first is a *statistical parameter estimator* for generating confidence intervals around a point estimate of the unknown parameter. The second is a *classifier* that dynamically learns constraint admissible sets by combining off-line sampled trajectory data with on-line data provided by the parameter estimator. In particular, classifiers like deep neural networks and kernel classifiers exhibit good approximation properties that can be employed to efficiently represent highly non-convex robust constraint admissible sets for black-box systems using simulations and systematic sampling, which would be extremely challenging for analytical methods [20].

While adaptive or learning-based RGs have been considered in the literature [21, 22, 23], there are some major differences with these prior works from our proposed PARG. In [21], the RG strategy allows violation of constraints before a robust constraint-admissible set is computed. Since the application is engine control where allowing constraint violations (e.g., misfires) is not catastrophic, this is a viable approach for a non-safety-critical application. However, our proposed PARG can be designed to have a small probability of failure with sufficient data. In [22], an $\mathcal{L}_1$ adaptive controller is cascaded to an RG: this combination is referred to as an 'adaptive RG'. The cascade approaches aim to use RGs as a layer of safety during adaptive control, which is fundamentally different from our method because we are adapting the RG itself, not an exogenous component. Finally, in the paper [23], the authors propose a learning RG, but the fundamental assumptions are different: the proposed approach is black-box rather than our grey-box PARG, and relies directly on output trajectories rather than models. Furthermore, the approach is deterministic; therefore, they do not utilize statistical learning.

The key features and novel contributions of our PARG approach are: (i) combining on-line and off-line learning enabling the application to general nonlinear systems with uncertain parameters; (ii) leveraging confidence intervals around parameter estimates rather than point estimates[2] for generating constraint admissible sets; (iii) providing probabilistic guarantees on the constraint satisfaction of the closed-loop system under PARG operation.

This paper is an extension of the preliminary study [24]. Novel extensions include: (i) a more detailed description of the algorithm, (ii) addition of theoretical guarantees of the components of the PARG algorithm, with extended and modified proofs and (iii) an extended and improved application to a detailed case study on vehicle dynamics, briefly presented in [25]. The rest of the paper is organized as follows. In Section 2, we provide the notation used in the ensuing discussion. In Section 3, we motivate the problem considered in the paper, and describe our proposed method of solving this problem. In Section 4, we present the data generation, estimation, and learning algorithms used in the PARG architecture. Theoretical guarantees are provided in Section 5 and the potential of the approach via a simulation example and a vehicle control case study is demonstrated in Section 6. These examples illustrate that the robust sets we estimate become less conservative with on-line learning and availability of more data, and that our proposed method performs well despite parameter estimation with model mismatch and piece-wise constant command inputs. We present our conclusions in Section 7.

2. Notation

The symbols $\mathbb{R}$, $\mathbb{Z}$, $\mathbb{Z}_+$, and $\mathbb{N}$ denote the set of reals, integers, non-negative integers, and natural numbers, respectively. We denote the norm ball of center c and radius ρ by $\mathcal{B}(c,\rho) := \{x : \|x-c\| \leq \rho\}$, where unless explicitly said differently, the 2-norm is used. We denote the transpose of a vector/matrix a by $a^\top$. Given two vectors a, b we denote the "stacking" of the vectors by $(a,b) = [a^\top\ b^\top]^\top$. The diagonal and anti-diagonal of a square $m \times m$ matrix M are denoted by $\text{diag}(M)$ and $\text{antidiag}(M)$, respectively. The probability of an event E is denoted by $\mathsf{Pr}(E)$; note

[2] An advantage of using confidence intervals rather than relying solely on point estimates of the parameter is that the intervals can exhibit certain properties such as monotonicity that are crucial to ensure performance guarantees on the PARG.

that we also abuse notation and use Pr to denote the outer measure. We denote a normal distribution with mean m and covariance Σ as $N(m, \Sigma)$. For sets A and B, the notation $A \subseteq B$ ($A \subset B$) indicates that A is (strictly) contained in B, and $|A|$ denotes the cardinality of A. The notation $\mathsf{grid}\ A$ describes a grid of samples drawn from a set A.

3. Motivation

3.1. Problem Statement

We consider the class of parametric discrete-time nonlinear systems

$$\begin{aligned} x_{t+1} &= f(x_t, v_t) + \theta^\top g(x_t, v_t), && \text{(1a)} \\ y_t &= h(x_t, v_t), && \text{(1b)} \end{aligned}$$

where $t \in \mathbb{Z}_+$ denotes the time-index, $x_t \in \mathbb{X} \subset \mathbb{R}^{n_x}$ is the measured system state, $v_t \in \mathbb{V} \subset \mathbb{R}$ is the reference input, $y_t \in \mathbb{R}$ is the output that should track a desired reference, and f, g, h are nonlinearities that represent the model and output dynamics. The vector $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ models a set of system parameters. The output y_t must satisfy constraints described by the set $\mathbb{Y} \subset \mathbb{R}$ for each instant of time, that is, we require

$$y_t \in \mathbb{Y} \text{ for every } t \geq 0. \tag{2}$$

Assumption 1. *The sets $\mathbb{X}$, $\mathbb{V}$, $\mathbb{Y}$ and Θ are compact and known. The sets $\mathbb{X}$, $\mathbb{V}$, and $\mathbb{Y}$ contain the origin in their interiors, and $\mathbb{V}$ is convex.*

Assumption 1 is fairly mild. Although the classical reference governor literature does not always require boundedness of $\mathbb{X}$, we make this an explicit assumption because we will use sampling-based methods to characterize robust constraint admissible sets. In order to determine where the set needs to be approximated most precisely, the domain over which the samples are drawn must be known and bounded. Furthermore, the reference v is assumed to be scalar, which is the standard formulation for the reference governor. The results of this manuscript can be extended almost directly to vector-valued v, which is the case for vector reference governor and command governor[3].

A common assumption for reference governors is that, in the absence of constraints, (1) exhibits good tracking performance, possibly because it models the actual plant already in closed-loop with a stabilizing controller.

Assumption 2. *When $\mathbb{Y} = \mathbb{R}$, (1) is asymptotically stable in $\mathbb{X}$. Furthermore, for any $r \in \mathbb{V}$, if $v_t = r_t \equiv r$ for all $t \geq 0$, then $y_t \to r$ as $t \to \infty$.*

The objective of a reference governor is to select v_t as close as possible to r_t, while ensuring that (2) is enforced. In the literature [3], the commonly studied cases are when θ is known or when it is unknown and constantly varying within a given range. In this paper, we consider the case when the parameter vector $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ is unknown, but constant or at most piece-wise constant. In many applications such as industrial motors [26], autonomous vehicles [27], and buildings [28], exact values of the model parameters are often not known, but they are basically fixed, or are slowly drifting due to aging or environmental effects. However, the set Θ is generally known from experience and statistical data.

Our proposed PARG is given by the control law

$$\begin{aligned} v_t &= \bar{\mathcal{G}}\big(v_{t-1},\, x_t,\, r_t,\, \mathcal{O}(\hat{\Theta}_t)\big) \\ &= v_{t-1} + \mathcal{G}\big(v_{t-1},\, x_t,\, r_t,\, \mathcal{O}(\hat{\Theta}_t)\big)\ (r_t - v_{t-1})\,, \end{aligned} \tag{3a}$$

where $\hat{\Theta}_t \subset \Theta$ is a bounded interval of parameter values, computed by a parameter estimator with the functional form

$$\hat{\Theta}_t = \mathcal{E}\big(v_{t-1},\, x_t,\, \hat{\Theta}_{t-1}\big), \tag{3b}$$

and

$$\mathcal{O}(\hat{\Theta}_t) = \Psi\big(\hat{\Theta}_t, x_t, v_{t-1}\big), \tag{3c}$$

is a robust constraint-admissible set (which we define formally in the following subsection) that is estimated using a machine learning algorithm Ψ by combining data generated both from offline simulations and online parameter estimates.

Our objective is to design $\mathcal{G}$, Ψ, and $\mathcal{E}$ such that the closed-loop system (1) with (3) satisfies constraints (2) in spite of parametric uncertainty and, when possible, tracks the desired reference r_t. A schematic diagram representation of the proposed PARG architecture is shown in Figure 1.

3.2. Proposed Solution

Constructing a PARG for the uncertain system (1) poses two major difficulties. First, parameter estimators generally converge asymptotically to the true parameters, implying that the current estimate of the parameter at any arbitrary finite time t is not necessarily correct, and in fact, could be quite far from the true value. Thus, designing references by trusting point estimates $\hat{\theta}_t \neq \theta_t$ at each time instant t could result in constraint violations. Second, model-based analytical methods for estimating parameter-robust invariant sets on-line for employment in a reference governor is computationally expensive because the underlying constraint admissible invariant sets are typically nonconvex with respect to the parameters; analytical methods are often limited to simplified dynamics, and commonly employed for systems with low-dimensional state-spaces.

In order to address the first difficulty, we propose using parameter estimators that generate not only a point-estimate $\hat{\theta}_t$, but also an interval $\hat{\Theta}_t$ which contains the true θ with certainty (e.g., using deterministic interval estimators) or with high probability (e.g., using Bayesian estimators). The advantage of using confidence intervals instead of point estimates is that they can be made to exhibit certain useful properties such as non-expansivity as more

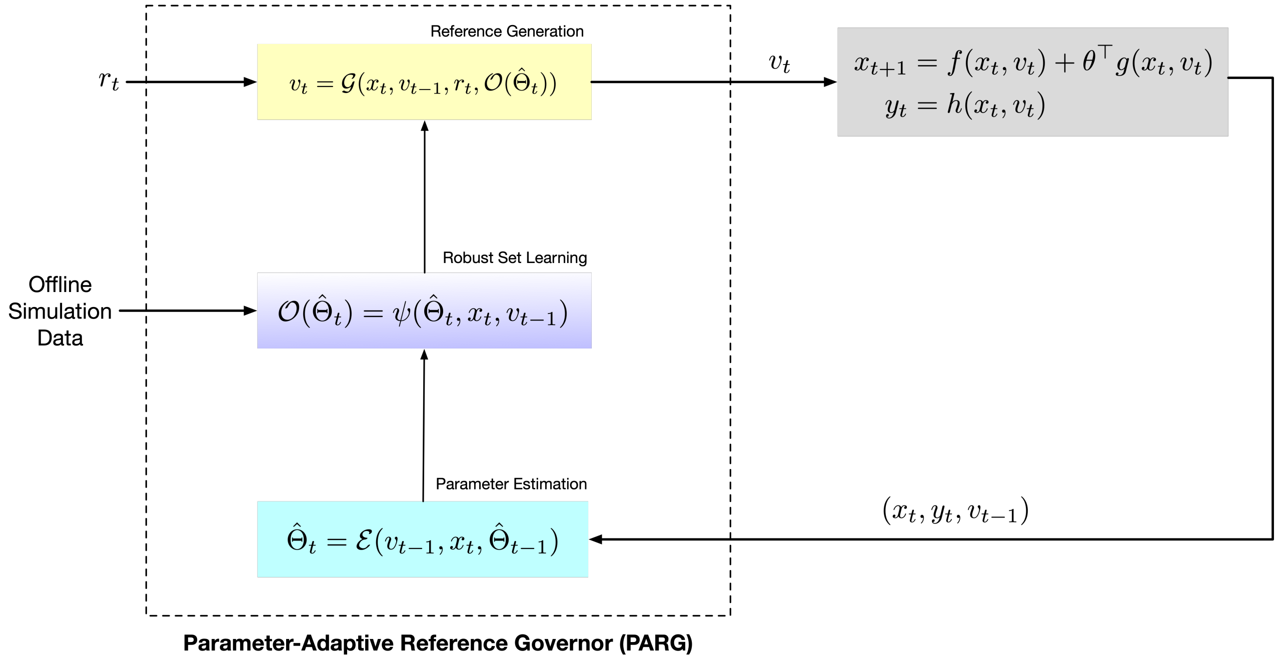
Reference Generation
r_t
$v_t = \mathcal{G}(x_t, v_{t-1}, r_t, \mathcal{O}(\hat{\Theta}_t))$
v_t
$x_{t+1} = f(x_t, v_t) + \theta^\top g(x_t, v_t)$
$y_t = h(x_t, v_t)$
Robust Set Learning
Offline Simulation Data
$\mathcal{O}(\hat{\Theta}_t) = \psi(\hat{\Theta}_t, x_t, v_{t-1})$
Parameter Estimation
(x_t, y_t, v_{t-1})
$\hat{\Theta}_t = \mathcal{E}(v_{t-1}, x_t, \hat{\Theta}_{t-1})$
Parameter-Adaptive Reference Governor (PARG)

Figure 1: Block diagram representation of the parameter-adaptive reference governor (PARG) added-on to the unconstrained system.

measured data becomes available on-line. Unlike point estimates, which can be time-varying and unpredictable, confidence intervals can be designed to exhibit more predictable dynamics, making them effective for constraint enforcement. Unfortunately, replacing a point-estimate with a confidence interval exacerbates the second difficulty as one now needs to estimate constraint-admissible or invariant sets that are robust to not only to parameter values, but ranges of parameter values. One of the key contributions of this paper is to estimate these *parameter-robust constraint-admissible sets* (PRCAS) in a computationally efficient manner by exploiting sampling and classification. Let $\mathcal{H} = \{(x, v) \in \mathbb{X} \times \mathbb{V} : h(x, v) \in \mathbb{Y}\}$ denote the set of state and reference inputs for which the output y satisfies the constraint (2).

Definition 1 (PRCAS)**.** *The set $\mathcal{O}(\hat{\Theta}) \subset \mathcal{H}$ is a parameter-robust constraint-admissible set for* (1) *if, for every initial condition $(x, v) \in \mathcal{O}(\hat{\Theta})$, when $x_0 = x$ and $v_t = v$ for all $t \geq 0$, $(x_t, v_t) \in \mathcal{H}$ for every $\theta \in \hat{\Theta}$, for all $t > 0$.*

Since the set $\mathcal{O}(\hat{\Theta})$ is invariant, we use the terminology parameter-robust invariant sets interchangeably.

In order to generate estimates of PRCASs, we will sample trajectories of the closed-loop system off-line and use the collected data to construct a dataset for a classifier that can learn these sets on-line. Such an approximation of a PRCAS can subsequently be used to evaluate the control law (3a) by solving at each time instant t the following optimization problem:

$$\mathcal{G}(v_{t-1}, x_t, \hat{\Theta}_t, r_t) := \underset{\gamma_t}{\arg\min}\ (v_t - r_t)^2 \tag{4a}$$

subject to:

$$(v_t, x_t) \in \mathcal{O}(\hat{\Theta}_t), \tag{4b}$$
$$v_t = v_{t-1} + \gamma_t (r_t - v_{t-1}), \tag{4c}$$
$$0 \leq \gamma_t \leq 1, \tag{4d}$$
$$v_t \in \mathbb{V}_\varepsilon(\hat{\Theta}_t) \tag{4e}$$

where $\mathbb{V}_\varepsilon(\hat{\Theta}_t)$ denotes the set of references v such that a ball of radius $\varepsilon > 0$ centered at the corresponding steady state $x^{\text{ss}}(v, \theta)$ and v lies inside $\mathcal{H}$, that is,

$$\mathbb{V}_\varepsilon(\hat{\Theta}_t) := \left\{ v \in \mathbb{V} : \mathcal{B}_\varepsilon(h(x^{\text{ss}}(v, \theta), v)) \subset \mathcal{H}, \forall \theta \in \hat{\Theta} \right\}.$$

4. Design of Parameter Adaptive Reference Governor

In this section, we discuss the off-line dataset construction phase, the on-line parameter estimation phase, and how the two can be combined to estimate PRCASs required to compute feasible reference inputs using our proposed PARG.

4.1. Data generation

We begin by simulating trajectories of the closed loop system (1) off-line, from different initial states sampled from $\mathbb{X}$, reference inputs sampled from $\mathbb{V}$, and parameters in Θ. At the end of each off-line simulation, if an initial condition $x_i \in \mathbb{X}$ tracks a desired reference input $v_i \in \mathbb{V}$

without violating the constraint (2) at any time in the simulation, for a parameter θ_i sampled within Θ, the combination (x_i, v_i) is labeled '+1' to indicate it resides within the PRCAS $\mathcal{O}(\{\theta_i\})$. Conversely, if the constraint is violated at any time point within the simulation, the feature (x_i, v_i) is labeled '−1' to indicate it resides outside $\mathcal{O}(\theta_i)$. In this way, we can construct a dataset consisting of states, reference inputs, parameters, and labels (± 1) that can be used by a classifier to estimate constraint-admissible sets.

Formally, we extract N_x unique samples from $\mathbb{X}$ and construct grids (not necessarily equidistantly spaced) on $\mathbb{V}$ and Θ with N_v and N_θ nodes, respectively. Let x_i denote the i-th sampled state, v_j the j-th reference input grid node, and θ_k the k-th parameter grid node. For each $(x_i, v_j, \theta_k) \in \mathsf{grid}\,\mathbb{X} \times \mathsf{grid}\,\mathbb{V} \times \mathsf{grid}\,\Theta$, we simulate the model (1) forward in time over a finite horizon T_s with a constant reference v_j and parameter θ_k. The horizon T_s is chosen long enough that the final tracking error can be smaller than a threshold (for example, $< 10^{-6}$) by the end of the simulation, for a feasible combination of (x_i, v_j, θ_k). For each simulation, we check whether $y_t \in \mathbb{Y}$ for every time instant. We set the corresponding label

$$\ell_i^{j,k} = \begin{cases} +1, & \text{if } y_t \in \mathbb{Y} \text{ for every } t \in \{0, 1, \dots, T_s\}, \\ -1, & \text{otherwise.} \end{cases} \tag{5}$$

At the end of such off-line data generation procedure, we have a dataset comprising initial conditions $\{x_i\}_{i=1}^{N_x}$, and each initial condition x_i has a corresponding matrix of labels $\ell_i \in \{-1, +1\}^{N_v \times N_\theta}$ described by

$$\ell_i = \begin{bmatrix} \ell_i^{1,1} & \cdots & \ell_i^{1,N_\theta} \\ \vdots & \ddots & \vdots \\ \ell_i^{N_v,1} & \cdots & \ell_i^{N_v,N_\theta} \end{bmatrix},$$

from which a classification problem can be cast on-line to determine a PRCAS. See Algorithm 1 for the pseudocode.

Remark 1. Note that for this sampling and data collection step, we do not need a causal, closed-form representation of the system (1). Even if one has a black-box representation of the dynamics (such as a software simulator), such a simulator can be used for generating closed-loop trajectories from, and therefore, sampling x_i and computing ℓ_i.

4.2. Confidence intervals from parameter estimator $\mathcal{E}$

Our proposed PARG utilizes confidence intervals around parameter estimates to learn PRCASs. While one may use any deterministic interval observer or stochastic estimator for this purpose, in this paper we demonstrate the performance using both Kalman filters and adaptive particle filters [18], which have demonstrated effectiveness on parameter estimation problems for nonlinear systems. We reformulate (1) in a probabilistic framework where θ is treated as an unknown disturbance with stochastic properties.

Algorithm 1 PARG Data Collection (Offline)

```
Require: Samples $\{x_i\}_{i=1}^{N_x}$ on state-space $\mathbb{X}$
Require: Grid $\{v_j\}_{k=1}^{N_v}$ on reference inputs $\mathbb{V}$
Require: Grid $\{\theta_k\}_{k=1}^{N_\theta}$ on parameter set $\Theta$
Require: Constraint set $\mathbb{Y}$
Require: Simulation model $\mathcal{M}$ of closed-loop dynamics (1) for time-horizon $T_s$ (black-box)
for each $(x_i, v_j, \theta_k)$ do
    $\{y_t\}_{t=0}^{T_s} \leftarrow$ simulate $\mathcal{M}$ with initial condition $x_i$, fixed $v_j$ and $\theta_k$
    $\ell_i^{j,k} \leftarrow$ obtain label using (5) using $\{y_t\}$
end for
    return data tuples $(x, v, \theta)$, label matrix $\ell_i$
```

If x_t is known, we can exploit the linearity of the system (1) with respect to θ and use a Kalman filter for estimating θ and its confidence interval Θ_t. The approach can be extended to the case when the state vector is not completely known and has to be estimated together with the parameter[27]. Specifically, we can reformulate (1a) as

$$\theta_t = \theta_{t-1} + w_{\theta,t}, \tag{6a}$$

$$\bar{y}_t = g(x_{t-1}, v_{t-1})^\top \theta_t + e_t, \tag{6b}$$

where $\bar{y}_t = x_t^\top - f^\top(x_{t-1}, v_{t-1})$, that is, the dynamical system (1a) for x_t now plays the role of the measurement (output) equation in the Kalman filter, $w_{\theta,t} \sim \mathcal{N}(0, Q_{\theta,t})$ and $e_t \sim \mathcal{N}(0, R)$ are Gaussian noise variables.

Eq. 6 is an estimation model applicable to many applications. However, for some applications a linear-in-parameter model does not sufficiently represent true underlying system. The estimator estimates θ based on knowledge of the estimated state x^e which does not need to be equal to the state x in the PARG. Sometimes, not only is θ unknown but also the state x^e is unknown and only implicitly measured. Hence, the estimator needs to resolve θ from the hidden state x^e. In a Bayesian framework the estimation model for a generic system can be written as

$$\theta_{t+1} = \theta_t + w_{\theta,t}, \tag{7a}$$

$$x_{t+1}^e = f^e(x_t^e, \theta_t, v_t) + w_{x^e,t}, \tag{7b}$$

$$\bar{y}_t = g^e(x_t^e) + e_t, \tag{7c}$$

where $w_{x^e,t} \sim \mathcal{N}(0, Q_{x^e})$ and $e_t \sim \mathcal{N}(0, R)$ are zero-mean Gaussian distributed with covariance Q_{x^e} and R, respectively.

The reason to address the parameter estimation problem in a Bayesian framework is that, even if the state x_t^e is known, for instance from measurements $\bar{y}_t$, such knowledge is typically imperfect due to inherent noise in the sensors measuring the state. Furthermore, a Bayesian framework provides a systematic approach to work with confidence intervals in recursive estimators. We address the parameter estimation problem by recursively estimating the posterior density function of the parameter θ_t, given by

$$\begin{aligned} p(\theta_t|\bar{y}_{0:t}) &= \int p(\theta_t, x^e_{0:t}|\bar{y}_{0:t})\,\mathrm{d}x^e_{0:t} \\ &= \int p(\theta_t|x^e_{0:t}, \bar{y}_{0:t})p(x^e_{0:t}|\bar{y}_{0:t})\,\mathrm{d}x^e_{0:t}. \end{aligned} \tag{8}$$

using the measurement history $\bar{y}_{0:T} = \{\bar{y}_0, \cdots, \bar{y}_T\}$.

The Bayesian updates for solving (8) can be summarized in the prediction and update equations

$$p(\theta_t|\bar{y}_{0:t-1}) = \int p(\theta_t|\theta_{t-1})p(\theta_{t-1}|\bar{y}_{0:t-1})\,\mathrm{d}\theta_{t-1}, \tag{9a}$$

$$p(\theta_t|\bar{y}_{0:t}) = \frac{p(\bar{y}_t|\theta_t)p(\theta_t|\bar{y}_{0:t-1})}{p(\bar{y}_t|\bar{y}_{0:t-1})}, \tag{9b}$$

where $p(\bar{y}_t|\bar{y}_{0:t-1})$ is a normalization constant. If the process noise w_t and measurement noise e_t are Gaussian distributed and if f^e and g^e are linear, the Bayesian update recursions (9) result in the Kalman filter equations that estimate the parameter mean $\hat{\theta}_t$ and associated covariance P_t. However, in other cases nonlinear filters, such as particle filters or linear-regression Kalman filters, need to be employed. Using the covariance, we estimate the confidence interval $\tilde{\Theta}_{t+1}$ as

$$\tilde{\Theta}^j_{t+1} = [\hat{\theta}^j_t - \beta P^{j,j}_t, \hat{\theta}^j_t + \beta P^{j,j}_t] \tag{10}$$

for each element j in the parameter vector θ_t and $\beta > 0$.

In order to provide theoretical guarantees on the PARG, we need to ensure that our confidence intervals do not expand over time, when more data becomes available, that is, $\hat{\Theta}_{t+1} \subseteq \hat{\Theta}_t$. While this is a natural consequence of applying Kalman filters to linear-in-parameter systems[29] such as (6), in general, exploration using nonlinear filters such as in particle filters could result in a violation of this condition. In such scenarios, we explicitly enforce contraction of confidence intervals. Specifically, if the filter computes an updated confidence interval $\tilde{\Theta}_{t+1}$, we set

$$\hat{\Theta}_{t+1} := \begin{cases} \hat{\Theta}_t \cap \tilde{\Theta}_{t+1} & \text{if } \tilde{\Theta}_{t+1} \cap \hat{\Theta}_t \neq \emptyset \\ \hat{\Theta}_t, & \text{otherwise.} \end{cases} \tag{11}$$

This forces non-expansion of $\hat{\Theta}_t$ for all $t \geq 0$.

Remark 2. If the state vector is available at every time instant t, a linear estimator can provide the confidence intervals, and, therefore, a more general approach using Bayesian recursions (9) is not needed. However, if the state is unavailable, the updates (9) can be employed to generate joint estimates of states and parameters via nonlinear recursive estimators. In Section 6.2, we give an example of how our approach applies to a realistic vehicle rollover-avoidance example.

4.3. Learning PRCAS using Ψ

We efficiently, albeit approximately, solve the problem (4b) by using machine learning and gridding $\mathbb{V}$. Gridding $\mathbb{V}$ and considering only the nodes, and taking into account the constraints (4c) and (4d), we deduce that the solution to (4) must be restricted to the sub-grid

$$\tilde{\mathbb{V}}_t := \big[\min\{r_{t,k}, v_{t,k}\}, \max\{r_{t,k}, v_{t,k}\}\big]. \tag{12}$$

Therefore, we can recast the problem (4) as a grid search,

$$\begin{aligned} v_t := \quad & \operatorname*{arg\,min}_{v\in\tilde{\mathbb{V}}_t} \,(v - r_t)^2 && \text{(13a)} \\ \text{subject to:} \quad & (v, x_t) \in \mathcal{O}(\hat{\Theta}_t), && \text{(13b)} \\ & v \in \mathbb{V}_\varepsilon(\hat{\Theta}_t). && \text{(13c)} \end{aligned}$$

In order to solve (13), we require an estimate of the set $\mathcal{O}(\hat{\Theta}_t)$. To this end, we formulate a time-varying binary classification problem where the inputs to the learner remain constant, but the labels change with time[3], as more data become available. Specifically, the PRCAS changes with time because the confidence interval $\hat{\Theta}_t$ is time-varying; this implies that a state x that was infeasible for $\hat{\Theta}_t$ may become feasible in a shrunken set $\hat{\Theta}_{t+1}$, even if v_t is fixed.

We set up the classification problem as follows. At time instant t, suppose the confidence interval $\hat{\Theta}_t$ is provided by the parameter estimator. Let

$$\mathcal{I}_{i,j}(\hat{\Theta}_t) := \Big\{ s : \theta_s \in \hat{\Theta}_t \cap \mathsf{grid}\,\Theta \Big\}$$

be the index set of parameters contained in the current confidence interval $\hat{\Theta}_t$ and the grid nodes $\{\theta_j\}_{j=1}^{N_\theta}$. Then, for each $v_j \in \tilde{\mathbb{V}}_t$ described in (12), and each $x_i \in \{x_i\}_{i=1}^{N_x}$ sampled off-line, we assign the label

$$z_{i,j}(\hat{\Theta}_t) = \min_{k\in\mathcal{I}_{i,j}(\hat{\Theta}_t)} \ell^{j,k}_i. \tag{14}$$

Taking the minimum in (14) ensures that the estimated set is robust to *all* parameters within $\hat{\Theta}_t$. That is, if even one θ_t is infeasible for the particular v_j and x_i, then x_i does not belong to the PRCAS corresponding to $\hat{\Theta}_t$. With the dataset $\mathcal{D} = \{(x_i, v_j), z_{i,j}\}$, we can construct classifiers ψ_j, where $j = 1, \ldots, |\tilde{\mathbb{V}}_t|$. For each v_j, a classifier is trained on inputs $\{x_i\}$ and their corresponding target labels $\{z_{i,j}\}$.

While any nonlinear classifier can be used here, as an example, we consider a 1-norm soft margin support vector machine (SVM) classifier in order to exploit generalization error bounds to provide theoretical guarantees on the PARG performance. A typical 1-norm SVM is trained on the dataset $(x_i, z_{i,j})$, for a given j, by solving the opti-

[3] In the machine learning literature, this is referred to learning with concept drift [30].

mization problem

$$(w_j^\star, b_j^\star) := \operatorname*{arg\,min}_{w,b,\xi\geq 0} \; w^\top w + c\,\|\xi\|_1 \tag{15a}$$

$$\text{subject to: } z_{i,j}\,(w^\top\varphi(x_i)+b) \geq 1-\xi_i, \quad \forall\, i=1,\ldots,N_x. \tag{15b}$$

Here, $c>0$ is a regularization constant, w quantifies the margin of separation, b is a bias term, ξ are slack variables, and φ is a continuous feature map into a reproducing kernel Hilbert space (RKHS) on which a kernel function $\mathcal{K}$ is defined. The decision function of the 1-norm SVM, whose zero level set describes the boundary of the PRCAS, is given by

$$\psi_j(x) = \mathsf{sgn}\left((w_j^\star)^\top\varphi(x)+b_j^\star\right). \tag{16}$$

Since the classifier may not estimate a true inner approximation of the set with finite data, one may select sub-level sets of the decision boundaries of the classifier until no infeasible sample is contained in the interior of the sub-level set [20]. A specific heuristic that can be employed to ensure constraint satisfaction is by choosing a small $\varepsilon>0$ and checking that $\psi_{j^\star}(x_t)>\varepsilon$ rather than $\psi_{j^\star}(x_t)>0$. This forces the state to lie in the interior of the set rather than on the boundary. In this way, the hyperparameter ε trades-off safety and performance. In the PARG context, this is the equivalent of the inner approximation of the PRCAS to achieve finite determination, in the standard reference governor [3].

Once these classifiers are trained, solving (13) becomes equivalent to selecting the node v_j on the grid $\tilde{\mathbb{V}}_t$ that minimizes (13a), while ensuring that the current state is predicted by the j-th classifier to belong to the PRCAS induced by $\hat\Theta_t$, that is: $\psi_j(x_t)>0$. The full pseudocode for the online PARG steps is provided in Algorithm 2.

Algorithm 2 PARG Implementation (Online)

```
Require: Parameter estimator $\mathcal{E}$
Require: Classifier $\psi$
Require: Samples $\mathsf{grid}\,\mathbb{X} := \{x_i\}_{i=1}^{N_x}$, $\mathsf{grid}\,\mathbb{V} := \{v_j\}_{j=1}^{N_v}$, and $\mathsf{grid}\,\Theta := \{\theta_k\}_{k=1}^{N_\theta}$
Require: Initial bounds on parameter estimate $\hat\Theta_0$
for $t = 0 : \infty$ do
    Compute current state $x_t$
    $\tilde\Theta_{t+1} \leftarrow$ compute confidence interval (CI) with $\mathcal{E}$
    $\hat\Theta_{t+1} \leftarrow$ enforce non-expansion using (11)
    for each $x_i, v_j$ do
        $\mathcal{I}_{i,j}(\hat\Theta_{t+1}) \leftarrow$ compute index set of parameter grid nodes within CI
        $z_{i,j}(\hat\Theta_{t+1}) \leftarrow$ assign label based on confidence using (14)
        $\psi_j \leftarrow$ construct classifier based on features $x_i$ and labels $z_{i,j}$
    end for
    $v_t \leftarrow$ minimize (13a) with constraint $\psi_j(x_t) > 0$
end for
```

Remark 3. A common way of implementing classifiers with good approximation properties in an on-line manner is by incrementally updating their hyperparameters based on changes in the updated label set rather than re-learning from scratch [31]. Fortunately, we will show later in Section 5 that a state will remain feasible for a fixed v_t, if it starts within a PRCAS corresponding to that v_t. This indicates that the set $\mathcal{O}(\hat\Theta_t)$ does not have to be updated at every time iteration. Thus, this method can be used for systems with small sampling times by learning the robust sets asynchronously.

Remark 4. Note that the learning of these robust sets are also performed without model knowledge, that is, in a purely data-driven manner that is agnostic to the system dynamics.

5. Idealized Performance Guarantees

In this section, we provide formal guarantees on both the PRCAS learning quality and the closed-loop performance of the PARG.

5.1. Bounding the SVM generalization errors

Recall that the optimization problem to be solved for training the soft-margin SVM is described in (15). Without loss of generality, we remove the bias term $b^\star$ from this discussion and denote the optimal solution $w^\star$. The margin of the SVM is given by

$$M^\star = 1/\|w^\star\|_2.$$

Since $\varphi : \mathbb{X} \to \Phi \subset \mathbb{R}^{n_\varphi}$ is a continuous feature map, and by assumption $\mathbb{X}$ is compact, the image $\Phi := \varphi(\mathbb{X})$ is compact. Therefore, there exists some positive scalar $R_\varphi > 0$ such that a n_φ-dimensional ball with radius R_ϕ covers Φ. The following discussion demonstrates that the misclassification rate of the robust invariant set classifier on unknown test datasets can be quantified based on training performance.

Lemma 1 (*Misclassification Error*)**.** *Let $\mathcal{D}_{\rm test}$ denote any distribution on $\Phi\times\{-1,+1\}$. Let $D_{\rm train}^{N_x}$ denote a training dataset comprising $N_x\in\mathbb{N}$ independent samples drawn from $\mathcal{D}_{\rm test}^{N_x}$. If there exists a $w^\star\in\mathbb{R}^{n_\varphi}$ such that $\|w^\star\|_2\leq 1$ and the classifier* (16) *has margin at least $M^\star$ on all the training examples in $D_{\rm train}^{N_x}$, then there exists some constant $C_0>0$ such that for every $\varepsilon_\psi\in(0,1)$, it holds with probability at least $1-\varepsilon_\psi$ over the training samples that the misclassification error rate on the test dataset satisfies the bound*

$$\mathsf{Pr}_{(\varphi(x),\,z)\sim\mathcal{D}_{\rm test}}\left[z\cdot(\varphi(x)^\top w^\star)\leq 0\right] \leq \check{C}_0(N_x,\varepsilon_\psi,M^\star), \tag{17}$$

where $\check{C}_0(N_x,\varepsilon_\psi,M^\star) := C_0\frac{(R_\varphi/M^\star)^2\ln N_x+\ln(1/\varepsilon_\psi)}{N_x}$.

Proof. Let

$$\mathcal{L}_{\rm test}(w) = \mathsf{Pr}_{(\varphi(x),\,z)\sim\mathcal{D}_{\rm test}}\left[z\cdot(\varphi(x)^\top w)\leq 0\right]$$

and

$$\mathcal{L}_{\text{train}}(w, N_x, M) = \mathsf{Pr}_{(\varphi(x),z)\sim\mathcal{D}_{\text{train}}^{N_x}}\left[z\cdot(\varphi(x)^\top w)\le M\right].$$

For every $\|w\|\le 1$, and margin $M = 1/\|w\|_2$, we know from [32] that there exist constants $C_0 > 0$ and $C_1 > 0$ such that the probability of

$$\mathcal{L}_{\text{test}}(w) \le \mathcal{L}_{\text{train}}(w, N_x, M) + \check{C}_1 + \check{C}_2, \tag{18}$$

is at least $1-\varepsilon_\psi$, where

$$\check{C}_1 := C_0\frac{(R_\varphi/M)^2\ln N_x + \ln(1/\varepsilon_\psi)}{N_x},$$

$$\check{C}_2 := C_1\left(\frac{(R_\varphi/M)^2\ln N_x\ln(1/\varepsilon_\psi)}{N_x}\cdot\mathcal{L}_{\text{train}}(w, N_x, M)\right)^{\frac{1}{2}}$$

For the training data $\mathcal{D}_{\text{train}}^{N_x}$ that yields $w^\star$ with margin $M^\star$ separating all training samples, we can write

$$\mathcal{L}_{\text{train}}(w^\star, N_x, M^\star) = 0.$$

Replacing this in (18) yields (17), which concludes the proof for finite-dimensional RKHS. For infinite-dimensional RKHS, one can apply Johnson-Lindenstrauss transforms [33] to first project onto general finite-dimensional Hilbert spaces and then identical arguments can be made to prove the result. □

Remark 5. Note that the conditions on the dataset in Lemma 1 are often not possible to check computationally; nevertheless, these conditions appear often in the literature and can be considered standard, c.f. [32].

Lemma 2. *Suppose the conditions of Lemma 1 hold and ε_ψ be a fixed, small quantity. If the classifier has margin at least $M^\star$ on all training samples for some sufficiently large $N_x^0\in\mathbb{N}$, then with probability $1-\varepsilon_\psi$ over training sets, for every $\pi_\psi\in(0,1)$ the classifier correctly classifies any test data sampled from $\mathcal{D}_{\text{test}}$ with probability at least π_ψ.*

Proof. Invoking (17), as long as N_x^0 satisfies

$$\frac{(R_\varphi/M)^2\ln N_x^0 + \ln(1/\varepsilon_\psi)}{N_x^0} \le \frac{1-\pi_\psi}{C_0}, \tag{19}$$

the misclassification error rate $\mathcal{L}_{\text{test}}(w^\star)\le 1-\pi_\psi$ with a probability $1-\varepsilon_\psi$ over training sets. For every π_ψ, such an N_x^0 exists for a given R_φ, $M^\star$, C_0, and ε_ψ because the left hand side of the inequality can be made arbitrarily small by increasing N_x^0. Since the misclassification probability is at most $1-\pi_\psi$, the probability of correct classification is at least π_ψ, which concludes the proof. □

An interpretation of Lemma 2 is that if the soft margin SVM optimization problem (15) possesses a feasible solution for arbitrarily large training sets, and the training data is separable by some (possibly infinite-dimensional) kernel function, then the generalization error can be made arbitrarily small by the same kernel.

Remark 6. Since we have $j = 1,\dots,N_v$ independent classifiers, the condition of Lemma 2 hold for $N_x^{0,j}$ for the j-th classifier. In such a case, one could take $N_x^0 := \max_j N_x^{0,j}$ to make the classification error probability uniformly at most $1-\pi_\psi$ over each classifier. Understanding that each classification is made independently, the probability of correct classification can remain at least π_ψ. In practice, however, choosing a very large N_x might be computationally challenging due to SVM's typically quadratic to cubic training complexity.

5.2. Guarantees on the PARG

We begin with the following lemma, which ensures that the true parameter lies in each confidence interval $\hat{\Theta}_t$ with probability determined by β in (10).

Lemma 3 (*Parameter Estimation Error*)**.** *If the estimator* (6) *uses the update* (11)*, then there exists a scalar $\pi_\theta\in(0,1)$ such that*

$$\mathsf{Pr}[\theta\in\hat{\Theta}_t, \forall t\ge 0]\ge\pi_\theta. \tag{20}$$

Proof. Let $\hat{\Theta}_\infty = \lim_{t\to\infty}\hat{\Theta}_t$. This limit exists because the sequence of sets is non-expanding and bounded. From (6), we note that our measurement output is linear in θ. Invoking the properties of the Kalman filter [29], there exists a scalar $\pi_\theta\in[0,1]$ such that $\mathsf{Pr}[\theta\in\hat{\Theta}_\infty]\ge\pi_\theta$. By the update (11), $\theta\in\hat{\Theta}_\infty$ implies $\theta\in\hat{\Theta}_t$ for all $t\ge 0$, because $\hat{\Theta}_t\subseteq\hat{\Theta}_{t-1}$. Thus, $\mathsf{Pr}[\theta\in\hat{\Theta}_t,\forall t\ge 0]\ge\mathsf{Pr}[\theta\in\hat{\Theta}_\infty]\ge\pi_\theta$. □

The following lemma ensures that the PRCASs do not contract for a fixed reference input, as long as the confidence intervals are non-expansive.

Lemma 4. *Recall that $\mathcal{H} = \{(x,v)\in\mathbb{X}\times\mathbb{V} : h(x,v)\in\mathbb{Y}\}$. The update* (11) *ensures $\mathcal{O}(\hat{\Theta}_t)\supseteq\mathcal{O}(\hat{\Theta}_{t-1})$ and $\mathcal{O}(\hat{\Theta}_t)\subseteq\mathcal{H}$ for all $t\ge 0$.*

Proof. For every $t\ge 0$, $\mathcal{O}(\hat{\Theta}_t)\subseteq\mathcal{H}$ by construction. From (11), $\hat{\Theta}_t\subseteq\hat{\Theta}_{t-1}$. Then, for any $x\in\mathbb{X}$ and any $v\in\mathbb{V}$ the set of possible trajectories of (1) for $\hat{\Theta}_t$ is a subset of those for $\hat{\Theta}_{t-1}$. The result follows from Definition 1. □

Lemmas 1–4 enable the following guarantees on the constraint satisfaction performance of the closed loop of (1) with the PARG (3a), (3b).

Theorem 1. *Suppose Assumptions 1, 2 hold and the conditions for Lemma 2 hold. Let ε_ψ, π_ψ, and π_θ be as defined in* (17) *and* (20)*, respectively. Let $t_0\ge 0$ denote a time instant at which $(x_{t_0}, v)\in\mathcal{O}(\hat{\Theta}_{t_0})$ for some $v\in\mathbb{V}_\varepsilon(\hat{\Theta}_{t_0})$. Then, the closed-loop system* (1)*,* (4)*,* (11) *satisfies the constraints* (2) *for any $r(t)\in\mathbb{V}$ and every $t\ge t_0$, with probability at least $\pi_\theta\pi_\psi$.*

Proof. At time t_0, let $\theta \in \hat{\Theta}_t$. Then, from Lemma 2 the constraints are satisfied with probability π_ψ because $(x_{t_0}, v) \in \mathcal{O}(\hat{\Theta}_{t_0}) \subseteq \mathcal{H}$. From Definition 1, $\mathcal{O}(\hat{\Theta}_{t_0})$ is invariant when $v_t = v_{t_0}$ for all $t \geq t_0$, and hence $(x_t, v) \in \mathcal{O}(\hat{\Theta}_{t_0}) \subseteq \mathcal{H}$ for all $t \geq t_0$. Since, from Lemma 4, we know that $\mathcal{O}(\hat{\Theta}_t) \supseteq \mathcal{O}(\hat{\Theta}_{t-1})$, thus $(x_t, v) \in \mathcal{O}(\hat{\Theta}_t)$ for all $t \geq t_0$. Therefore, $v_t = v$ is a recursively feasible solution of (4), and the set of solutions of (4) is never empty, for $t \geq t_0$. Since any feasible solution v_t of (4) satisfies $(x_t, v) \in \mathcal{O}(\hat{\Theta}_t)$ due to (4b), the property holds for any value v_t applied by (3a). From Lemma 3, the probability of $\theta \in \hat{\Theta}_t$ for all $t \geq 0$ is at least π_θ, and from Lemma 2 the probability of correctly reconstructing $\mathcal{O}(\hat{\Theta})$ by SVM classification is at least π_ψ. Then, assuming that the parameter and PRCAS estimation events are independent, the probability of the result above holding is at least $\pi_\theta \pi_\psi$. □

Theorem 2. *Let the conditions of Theorem 1 hold. Let $r(t) = r$ for all $t \geq 0$, and let there be a finite time $\hat{t}$ such that $r \in \mathbb{V}_\varepsilon(\hat{\Theta}_{\hat{t}})$. Then, there exists a finite $\bar{t} \geq \hat{t}$ such that $v(\bar{t}) = r$, with probability at least $\pi_\theta \pi_\psi$.*

Proof. For a standard reference governor with known parameter vector θ, and fixed $\mathcal{O}(\{\theta\})$, v_t converges in finite time to $r \in \mathbb{V}_\varepsilon(\{\theta\})$ by the asymptotic stability of (1), Assumption 2, and $v \in \mathbb{V}_\varepsilon(\{\theta\})$. Due to these, in a finite time $\tilde{t}$, $(x_{\tilde{t}}, v) \in \mathcal{B}((x^{\mathrm{ss}}(v), v), \varepsilon/2)$, and the updated reference $v + \frac{\varepsilon}{2|r-v|}(r - v)$ is feasible and reduces the distance $(r - v)$ by $\varepsilon/2$. Thus, there are finite decreases of the reference tracking error in finite time intervals, which eventually lead to $v_t = r$ for a finite time t [34, 3].

Consider the case $\theta \in \hat{\Theta}_t$, $\hat{\Theta}_t = \hat{\Theta}_{t-1}$ for all $t \geq 0$. Assumption 2 provides asymptotic stability for any value of $\theta \in \Theta$. Thus, there exists $\tilde{t} < \infty$ such that $(x_{\tilde{t}}, v) \in \mathcal{B}((x^{\mathrm{ss}}(v), v), \varepsilon/2)$ when $v_t = v$ for all $t \in [0, \tilde{t})$, and hence at time $\tilde{t} < \infty$, $v_{\tilde{t}}$ can be selected to reduce the tracking error by at least $\varepsilon/2 > 0$. Since $\hat{\Theta}_t \subseteq \hat{\Theta}_{t-1}$ by Lemma 4, $\mathbb{V}_\varepsilon(\hat{\Theta}_t) \supseteq \mathbb{V}_\varepsilon(\hat{\Theta}_{t-1})$, and hence if $v \in \mathbb{V}_\varepsilon(\hat{\Theta}_{t-1})$, then $v \in \mathbb{V}_\varepsilon(\hat{\Theta}_t)$, which means that v_t, $t \in [0, \tilde{t}]$ is still feasible when Θ_t is updated by (11). Thus, the finite-time interval to see the reference tracking error reduction may only shorten, and the size of the reference tracking error reduction may only increase due to the updates to the parameter confidence interval. Finally, we prove that the reference r satisfies (4e) over an infinite interval. Again, because of the parameter update, if $r \in \mathbb{V}_\varepsilon(\hat{\Theta}_{\hat{t}})$, then $r \in \mathbb{V}_\varepsilon(\hat{\Theta}_t)$, for all $t \geq \hat{t}$, which means that r will be steady state admissible at any time past $\hat{t}$. For ensuring that the governor continues operating along the entire trajectory, it is sufficient that $\theta \in \hat{\Theta}_t$ for all $t \geq 0$. As for Theorem 1, these conditions hold with probability $\pi_\theta \pi_\psi$, due to the confidence of $\theta \in \hat{\Theta}_t$, and the probability of correctly estimating $\mathcal{O}(\theta)$. □

Roughly, Theorem 1 states that if the learning algorithm has sufficiently small misclassification error probability, and the confidence interval induced by the parameter estimator contains the true parameter with sufficiently high probability, then as long as the closed-loop system satisfies constraints after a finite time $t_0 \geq 0$, it will remain constraint-feasible for every subsequent time instant with a sufficiently high probability as well. Theorem 2 implies a form of recursive feasibility, wherein if a particular reference r results in a constraint-feasible closed-loop system, then keeping that reference fixed for all subsequent time is guaranteed (with some probability) to keep the closed-loop system constraint-feasible.

6. Simulation Examples

In this section we verify the proposed control method using two case studies: a lower dimensional mass-spring-damper system to showcase the evolution of the robust invariant sets during on-line learning, and a vehicle system example to demonstrate the effectiveness of the approach on real-world applications.

6.1. Example 1: Electromagnetically-actuated mass-spring-damper

We consider a second-order nonlinear electromagnetically actuated mass-spring damper system [1] for testing the PARG. The closed-loop system without the reference governor is given by the forward Euler discretization of

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\frac{c + c_d}{m} x_2 + \theta \left(\frac{1}{m} v - \frac{1}{m} x_1 \right), \\ y &= x_1, \end{aligned}$$

with sampling time $\tau = 1$ ms.

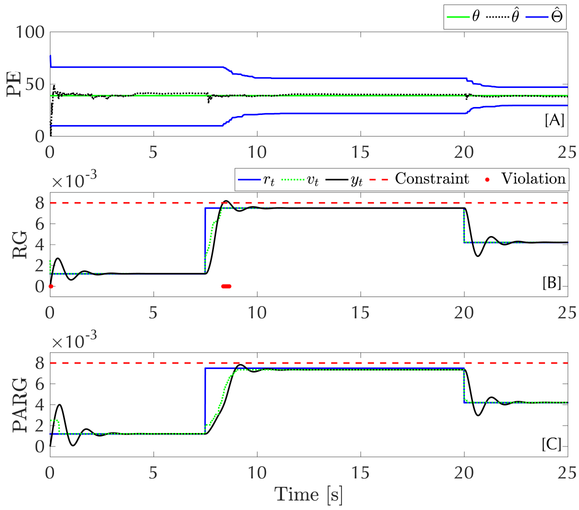

Figure 2: [A] Mean parameter estimate, the true parameter value, and the 99% confidence interval. [B] Measured output of closed-loop system with conventional RG with incorrect parameter estimate. [C] Measured output of closed-loop system with learning-based PARG. No constraints are violated. PE = parameter estimator, RG = reference governor without adaptation, PARG = parameter-adaptive reference governor.

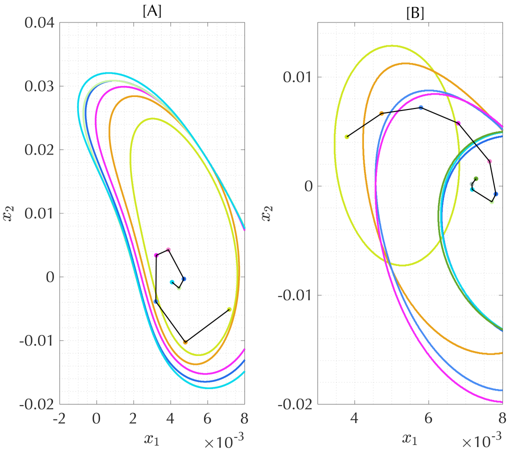

Figure 3: Phase plots of system with PARG-in-the-loop. [A] Evolution of the states and $\mathcal{O}(\hat{\Theta}_t)$ with fixed v_t and varying $\hat{\Theta}_t$. [B] Evolution of the states and $\mathcal{O}(\hat{\Theta}_t)$ with varying v_t and varying $\hat{\Theta}_t$.

The parameter $\theta = 38.94$ is the unknown spring constant. The values of the other parameters are $c_d = 4.00$, $c = 0.66$, and $m = 1.54$. The set $\mathbb{Y}$ is described by the constraints $x_1 \leq 8 \times 10^{-3}$ and $0 \leq u(x, v) \leq 0.3$, where

$$u(x, v) = \frac{1}{\alpha}(\theta v - c_d x_2)(d_0 - x_1)^{\nu}$$

is the legacy tracking controller whose structure and parameters θ, c_d, $\alpha = 4.5 \times 10^{-5}$, $d_0 = 1.02 \times 10^{-2}$, and $\nu = 1.99$, cannot be altered. At design time, we know that $\theta \in [10, 90]$. Thus, we initialize $\hat{\Theta}_0 = [10, 90]$, and we have $\mathbb{V} = [0.5 \times 10^{-3}, 7.5 \times 10^{-3}]$, and $\mathbb{X} = [-8 \times 10^{-3}, 8 \times 10^{-3}] \times [-4 \times 10^{-2}, 4 \times 10^{-2}]$.

For off-line dataset generation, we select $N_x = 2000$ Halton low-discrepancy samples on $\mathbb{X}$, and construct regular grids on Θ and $\mathbb{V}$ using 50 and 80 equidistant nodes, respectively. Empirically, we computed that the 95% settling time of the mass-spring-damper system was 3.5 second, for each (x_i, v_j, θ_k) we simulate the closed-loop dynamics forward in time for $T_s = 5$ seconds. Consequently, we check whether the constraints were violated at any time point on $t \in [0, 5]$s for generating labels.

We note that the mass-spring-damper system can be written in the linear-in-parameter form (7), and thus a Kalman filter can be designed to provide estimates of the parameter θ. The Kalman filter covariance matrices are set to $R_f = \mathrm{diag}\left(\left[10^{-4}, 10^{-8}\right]\right)$ and $Q_f = 10^{-8} I$. The 99% confidence interval $\hat{\Theta}_t$ generated by the parameter estimator is used to learn the PRCASs online. Concretely, we can compute $\hat{\Theta}_t$, described in (14), and use support vector machine (SVM) bi-classifiers to estimate the PRCAS as described in Section 4.3. The SVM employs a radial basis function kernel, with no box constraints, a sequential minimal optimization [35] (SMO) solver[4], and kernel length-scales automatically selected by MATLAB R2021a. In order to promote the generation of strictly feasible sets, where no point labeled '-1' is contained in the set, we use an asymmetric cost function with cost matrix $\left[\begin{smallmatrix}0 & 1\\ 100 & 0\end{smallmatrix}\right]$.

We compare the performance of the learning-based PARG to a non-adaptive RG that assumes a parameter value of $\tilde{\theta} = 45$, which is the point estimate $\hat{\theta}$ after 100 measurement have been collected online. We will show that such a point estimate can lead to constraint violations, which the PARG can prevent by the use of robust sets. The output of the parameter estimator is illustrated in Figure 2[A] (dotted line) along with the true parameter value (green continuous line). The point estimate $\hat{\theta}$ converges to a small neighborhood around θ within 1 s, and the 99% confidence intervals (blue continuous lines) start contracting to a tight set around θ around 20 s. Note that $\hat{\Theta}_t$ contracts when the desired reference r_t jumps and v_t varies, since these jumps excite the closed-loop system, leading to satisfaction of weak persistence of excitation conditions. Figures 2[B] and 2[C] illustrate that the robust sets learned by the statistical learning algorithm is not conservative, capable of generating irregular geometries, and adapts quickly with change in the confidence intervals computed by the Kalman filter. Figure 2[B] confirms that using only a point-estimate and not adapting with measurements leads to constraint violations, because the constraint-admissible set is generated based on an incorrect estimate $\tilde{\theta}$. Conversely, as evident from Figure 2[C], the PARG, which is informed by the learned PRCASs, does not violate constraints over the length of the simulation.

Figure 3 depicts the time-evolution of the robust sets. In Figure 3[A], we see that, for a fixed v_t, the sets expand with time; each updated state shares the same colored dot as the corresponding invariant set. The expansions occur because the intervals $\hat{\Theta}_t$ contract and so the invariant sets become less conservative for the same v_t. When v_t changes, as in Figure 3[B], the shapes of the sets alter according to how close they are to the constraints. However, as expected, the states always lie within the PRCASs, which implies that the constraints are never violated.

6.2. Example 2: Vehicle Rollover Avoidance

This second case study is to demonstrate the potential of the PARG algorithm on realistic application with larger dimension (5 states, 3 uncertain parameters) and varying reference, namely, vehicle rollover prevention. Rollover accidents, while infrequent, contribute to a significant percentage of severe accidents and fatalities [36]. Development of control principles for avoiding such accidents has therefore been a major topic of research. The center-of-gravity (CoG) location is the most prominent factor in rollover occurrence. However, before vehicle operation, the CoG location can only be determined under nominal conditions, and therefore a control strategy benefits from the capability of estimating in real-time the CoG location. It is important to note that most rollover prevention control

[4] We observed that other commonly used SVM solvers such as L1QP or ISDA significantly increase the training time.

methods assume a known CoG location [37, 38, 39], but in practice the CoG location changes with loading conditions and the maneuver [40].

6.2.1. Modeling

We use a control-oriented model of the vehicle based on a nonlinear chassis 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 for the roll dynamics. 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

$$
\begin{aligned}
\dot{v}^X &= v^Y\dot{\psi} - mh\dot{\phi}\cos(\phi) + F^X/m, && \text{(21a)}\\
\dot{v}^Y &= -v^X\dot{\psi} + h(\ddot{\phi}\cos(\phi) - \dot{\phi}^2\sin(\phi)) + F^Y/m, && \text{(21b)}\\
\ddot{\psi} &= M^Z/I_Z, && \text{(21c)}\\
\dot{\phi} &= \dot{\phi}, && \text{(21d)}\\
\ddot{\phi} &= \frac{h(F^Y + m\sin(\phi))(g + h\dot{\phi}^2) + \tau_\phi}{I_X + mh^2\cos(\phi)}, && \text{(21e)}
\end{aligned}
$$

where

$$
\begin{aligned}
F^X &= F_f^x\cos(\delta) - F_f^y\sin(\delta) + F_r^x, && \text{(22a)}\\
F^Y &= F_f^x\sin(\delta) + F_f^y\cos(\delta) + F_r^x, && \text{(22b)}\\
M^Z &= l_f(F_f^x\sin(\delta) + F_f^y\cos(\delta)) - l_rF_r^y, && \text{(22c)}\\
\tau_\phi &= -K_\phi\tan(\phi) - D_\phi\dot{\phi}\cos(\phi), && \text{(22d)}
\end{aligned}
$$

g is the gravitational acceleration, h is the CoG distance, the distance from the roll axis to the CoG, m is the vehicle mass, I_X, I_Z, are the vehicle inertias about the X- and Z-axis, respectively, K_ϕ is the spring stiffness, D_ϕ is the damping coefficient, and δ is the road-wheel angle of the front wheels, which is used as the reference input to be designed. In this work the vector of uncertain parameters is $\theta = \{K_\phi,\, D_\phi,\, h\}$.

We use the Pacejka tire model [41] for the tire-road interaction. We model combined slip similar to [39],

$$
\begin{aligned}
&\begin{bmatrix} F_i^x & F_i^y \end{bmatrix}^\top = F_P P\hat{s}, \quad s = \begin{bmatrix} \lambda_i \\ \tan\alpha_i \end{bmatrix}, l\hat{s} = s\ \|s\|,\\
&P = \sin\left(C\,\mathrm{atan}\left(s_c/C(1-E) + E\,\mathrm{atan}(s_c/C)\right)\right),\\
&s_c = \frac{C_\alpha\|s\|}{F_P},\ C_\alpha = c_1mg(1 - e^{-c_2z_i^Z/(mg)}), && \text{(23)}\\
&c_1 = \frac{BCD}{4(1 - e^{-c_2/4})},\ F_P = \frac{F_i^z 1.0527}{1 + \left(\frac{1.5F_i^z}{mg}\right)^3},
\end{aligned}
$$

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 (23) are

$$
\alpha_i = -\arctan\left(\frac{v_i^y}{v_i^x}\right),\ \lambda_i = \frac{R_w\omega_i - v_i^x}{v_i^x},\ i \in \{f, r\}, \quad (24)
$$

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.

The input constraint is symmetric and 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 [38, 39], we express the load transfer ratio (LTR) as a percentage, given by

$$
\mathsf{LTR} = 100 \times \frac{F_L^z - F_R^z}{mg}, \quad (25)
$$

where F_L^z, F_R^z, are the left and right vertical loads, respectively. The LTR measures the relative load on each side of the vehicle and wheel liftoff occurs when either $\mathsf{LTR} > 100\%$ or $\mathsf{LTR} < -100\%$.[5] The LTR is a function of the roll angle and roll rate and can therefore be calculated from the lateral load transfer [42].

6.2.2. Bayesian CoG Estimation

From (21), we observe that the roll dynamics are linear in the vehicle states but nonlinear in the parameters. We formulate an estimation method based on a marginalized[17, 43] adaptive particle filter (MAPF), where $\theta = \{K_\phi,\, D_\phi, h\}$ is estimated using a particle filter, which adapts its number of particles in response to convergence monitoring. Conditioned on the particles, the states can be analytically estimated with a conditional Kalman filter. We briefly summarize the estimator next and refer to [6] for more details. We model the roll-plane dynamics by

$$
(I_x + mh^2)\ddot{\phi} + D\dot{\phi} + K\phi = mh(a_y\cos\phi + g\sin\phi), \quad (26)
$$

and the parameters as nearly constant position models with prior distribution $\theta_0 \sim p_0(\theta)$. Consequently, we can write the system in state-space form with state $x^e = \begin{bmatrix}\phi & \dot{\phi}\end{bmatrix}^\top$ similar to (7),

$$
\begin{aligned}
\theta_{t+1} &= \theta_t + w_{\theta,t}, && \text{(27a)}\\
x_{t+1}^e &= A(\theta_t)x_t^e + B(\theta_t)a_y, && \text{(27b)}\\
\bar{y}_t &= x_t^e + e_t, && \text{(27c)}
\end{aligned}
$$

where $e_k \sim N(0, R)$ and the lateral acceleration is modeled as $a_y \sim N(a_{y,m}, Q_a)$, where $a_{y,m}$ is the measurement and Q_a can be determined using standard sensor calibration methods [29].

The estimation of θ is based on the factorization

$$
p(x_t^e, \theta_{0:t}|\bar{y}_{0:t}) = p(x_t^e|\theta_{0:t}, \bar{y}_{0:t}).p(\theta_{0:t}|\bar{y}_{0:t}). \quad (28)
$$

[5] There are some exceptions to this rule, c.f. [39, Remark 1].

The second distribution in (28) is approximated by a particle filter. Given the nonlinear state trajectory $\theta_{0:t}$, the first distribution in (28) is Gaussian and hence it can be estimated with conditional KFs, one for each particle i conditioned on the i-th particle trajectory. Then, (8) is computed from (28) by approximating the integral using the particles. The main difference compared with the standard KF is an extra measurement update for each KF using the forward propagated θ^i_{t+1} as an extra measurement.

For the implementation of the MAPF used in this paper, (7) is equivalent to *Model 1* in [43], which for the rollover avoidance application is the roll-dynamics model (27). For particle adaptation, we initialize the filter with $N = 8000$ particles, and the minimum number of particles in the particle adaptation is set to $N = 1000$. This gives a balance between execution time and performance. For more details about the MAPF implementation, we refer the reader to our prior work [6].

6.2.3. PRCAS Learning

The first step of the learner design is the offline data collection. To this end, we extract $N_x = 2000$ samples on a 5-dimensional admissible state space selected based on domain expertise,

$$\mathbb{X} = [15, 25] \times [-1.5, 1.5] \times [-8\pi/18, 8\pi/18]^3.$$

We also construct grids for parameters and references, where the bounds on these spaces are

$$\Theta = \hat{\Theta}_0 = [4 \times 10^4, 8 \times 10^4] \times [4 \times 10^3, 1 \times 10^4] \times [0.5, 1.2]$$

and

$$\mathbb{V} = [-0.075, 0.075],$$

with $N_\theta = 20^3$ and $N_v = 40$ nodes, respectively; the reference angle is in radian. We find that the settling time of the vehicle is approximately 1.5s, so for each sample (x_i, v_j, θ_k), we simulate the vehicle dynamics (21)–(22) for 2s with sampling period of 10ms, and compute the LTR constraint via (25) in order to assign a label to the sample. To promote caution on this practical system, we label constraint violation if $|\mathsf{LTR}| > 98\%$. The prediction model is (21)–(24), which includes lateral and longitudinal dynamics, nonlinear tire forces including combined slip, and suspension dynamics, which includes both sprung and unsprung mass models. The simulation model and the parameters used are the same as in [39], where are said to be similar to a North-American SUV.

Online, we use an SVM with the same implementation details as in the previous example, with the addition that the inputs to the SVM are first standardized due to the varying orders of magnitude. The asymmetric cost matrix has the same form as the previous example, with 100 replaced by 10^6 to heavily penalize any infeasible sample being classified as feasible. We also get comparable results using a polynomial kernel with a box constraint of 10. The computation of the reference must occur in real-time and for these simulations it is 0.89s on average and 1.93s in the worst case. The PRCAS update for these simulations takes 71.32ms on average and 0.92s in the worst case. However, the PRCAS update does not need to happen in real-time, since a current PRCAS always guarantees probabilistic constraint satisfaction. In fact it may be more effective to update the PRCAS after a few more data points are obtained, in an anytime fashion. For the current application less than 12MB of total memory is required to store the data for PRCAS. All such data are based on a non-optimized implementation of the method in `MATLAB`; note that an optimized `C` implementation can easily be 10×–40× faster and reduce storage usage by one order of magnitude or more: for example, by data compression and active data selection. Additionally, the use of SVM is exemplar, and in practice one could use deep neural networks or other machine learning algorithms that can be trained faster online, and do not require matrix inversion for computing the training loss.

6.2.4. Simulation Results

To test the PARG on an aggressive reference input, we use the Fishhook maneuver as the command reference r, 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. The only reference input is the wheel steering angle δ controlled by the PARG; there is no other control algorithm operating on the vehicle. These results were generated using the simulation model (21)–(24).

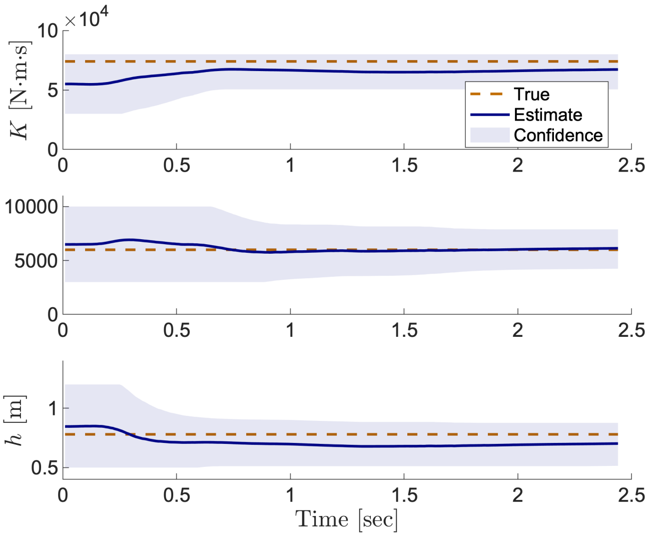

Figure 4: Parameter estimation results for the Fishhook maneuver. True values in red dashed, estimates in blue, and 2σ interval is shaded light blue.

Figure 4 shows the estimation results during closed-loop simulation with the PARG determining the steering angle. After the initial transients, the spring stiffness K_ϕ, damping coefficient D_ϕ, and CoG distance h all converge close to their respective true values. The estimates are contained within the confidence intervals, which reduce with time until they reach a steady-state confidence interval, limited by the noise level that we have selected for the MAPF.

Herein, we compare three reference selection algorithms to showcase the effectiveness of the PARG: (1) the PARG algorithm, as described in this section; (2) a nominal controller that uses the Fishhook reference command directly without adaptation; and (3) a prescient 'oracle' RG that operates with complete knowledge of the system, including the true parameters. The oracle-RG is implemented online by simulating the vehicle model with true parameters and selecting as v_t the value that does not result in a constraint violation and is closest to the reference. The oracle-RG is the ideal case where all parameters are known, but if these are assumed known yet are incorrectly set, constraint violations often occur in the maneuvers simulated next. The other comparison could be with a robust RG obtained by linearizing the nonlinear dynamics at multiple operating points, each for the extreme values of the a-priori parameter bounds, and using the resulting polytopic difference inclusion to compute a robust constraint admissible set [44]. However, due to the linearization at multiple operating points and the a-priori parameter ranges, such robust RG is extremely conservative, and the simulated maneuvers are poorly executed.

Figure 5 shows the closed-loop constraint satisfaction and resulting steering angle command of PARG, oracle-RG, and nominal Fishhook maneuver, where the reference is directly applied as command, for the same initial conditions of Figure 4. Following the nominal Fishhook maneuver results in repeated constraint violation (black dots) indicating wheel lift-offs, while PARG and oracle-RG satisfy the constraints. PARG is slightly more conservative than oracle-RG, since the latter perfectly knows the vehicle parameters, which results in slightly smaller LTR values. While oracle-RG reaches the reference, the PARG converges to a value close to the reference due to the uncertainty not disappearing even at steady state, as discussed above. On the other hand, during the transient, from 0.2s to 0.5s, the PARG behaves very similarly to the oracle-RG showing that is not excessively conservative. Figure 6 shows recursive feasibility according to Theorem 1.

Figure 6 shows the closed-loop constraint satisfaction and resulting steering angle command for a slightly less aggressive version of the Fishhook maneuver. In this scenario, PARG also reaches the reference according to Theorem 2, albeit slower than the oracle-RG due to accounting for the uncertainty. The interested reader can find the improvement of performance of PARG over a robust RG in this case study in [25], which is not repeated here.

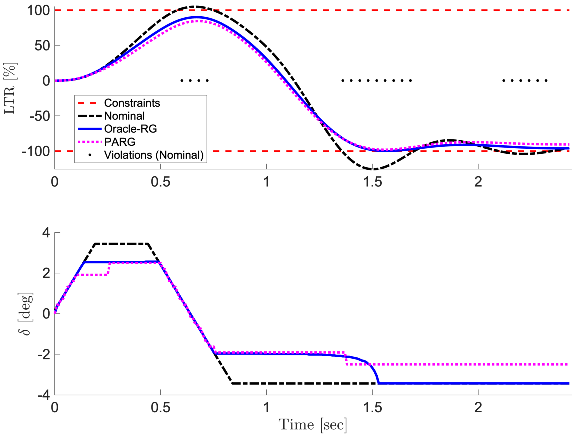

Figure 5: LTR and steering angle for closed-loop control in the Fishhook maneuver. The PARG using the estimator with results in Figure 4 satisfies the LTR constraint, whereas the nominal steering reference violates multiple times the constraints.

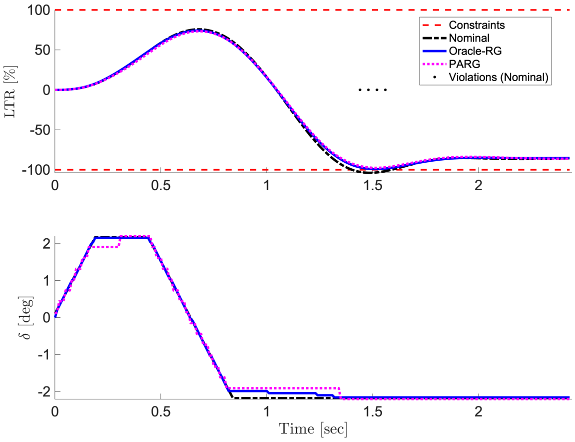

Figure 6: LTR and steering angle for closed-loop control in a less aggressive Fishhook maneuver (compare with Figure 5). The PARG using the estimator with results in Figure 4 satisfies the LTR constraint and converges to the reference as the oracle RG.

A detailed comparison of PARG with other robust or adaptive RG methods is beyond the scope of this paper, and in general may not be appropriate due to the structural differences of PARG with existing algorithms. Specifically, the method in [45] considers additive disturbance, the method in [21] freely allows constraint violations during learning, which is impractical for rollover avoidance, and the method in [23] is completely black-box and arguably requires larger amount of data and exploration of several trajectories, as opposed to our grey-box approach.

7. Conclusions

In this paper, we developed an adaptation mechanism for reference governors that can handle constraint satisfaction in systems with parametric uncertainties. Offline, we use machine learning to construct a parameter-robust constraint admissible set for different ranges of the uncertainty and online we use the confidence interval obtained from the estimator to instantiate the invariant set used for determining the command. The dynamics of the estimator confidence intervals and the properties of the algorithm used for machine learning ensure recursive feasibility in probability, as well as finite time convergence in probability to robustly admissible references. We have demonstrated the method in two case studies of different difficulty. Future work will investigate how to reduce the sample-complexity to further reduce the online computational load of the PARG and investigate using PARG with non-statistical estimators, such as interval estimators and set membership estimators.

References

[1] E. Gilbert and I. Kolmanovsky, "Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor," *Automatica*, vol. 38, no. 12, pp. 2063–2073, 2002.

[2] E. Gilbert and K. Tan, "Linear systems with state and control constraints: the theory and applications of maximal output admissible sets," *IEEE Trans Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.

[3] E. Garone, S. Di Cairano, and I. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 75, pp. 306–328, 2017.

[4] F. Tedesco, C. Ocampo-Martinez, A. Casavola, and V. Puig, "Centralized and distributed command governor approaches for water supply systems management," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 4, pp. 586–595, 2018.

[5] G. Tartaglione, M. Ariola, G. De Tommasi, A. Petrillo, S. Santini, and F. Amato, "Constrained reference tracking via structured input-output finite-time stability," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11, 2022.

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

[7] K. Liu, N. Li, I. Kolmanovsky, D. Rizzo, and A. Girard, "Model-free learning for safety-critical control systems: A reference governor approach," ser. Proc. American Control Conference (ACC). IEEE, 2020, pp. 943–949.

[8] K. Han, J. Feng, Y. Li, P. Jiang, and X. Wang, "Fault-tolerant tracking control optimization of constrained lpv systems based on embedded preview regulation and reference governance," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–13, 2022.

[9] F. Blanchini and S. Miani, *Set-theoretic methods in control.* Birkhäuser Boston, 2007.

[10] J. Sun and I. Kolmanovsky, "Load governor for fuel cell oxygen starvation protection: A robust nonlinear reference governor approach," *Proc. of the American Control Conference*, vol. 1, no. 6, pp. 828–833, 2004.

[11] Z. Peng and J. Wang, "Output-feedback path-following control of autonomous underwater vehicles based on an extended state observer and projection neural networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 4, pp. 535–544, 2018.

[12] C. J. Ong, S. S. Keerthi, E. G. Gilbert, and Z. Zhang, "Stability regions for constrained nonlinear systems and their functional characterization via support-vector-machine learning," *Automatica*, vol. 40, no. 11, pp. 1955–1964, 2004.

[13] A. Chakrabarty, A. Raghunathan, S. Di Cairano, and C. Danielson, "Data-driven estimation of backward reachable and invariant sets for unmodeled systems via active learning," ser. Proc. of the IEEE Conf. on Decision and Control (CDC). IEEE, 2018, pp. 372–377.

[14] R. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 15, no. 2, pp. 175–184, 1970.

[15] S. Särkkä and A. Nummenmaa, "Recursive noise adaptive Kalman filtering by variational Bayesian approximations," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 596–600, 2009.

[16] Y. Huang, Y. Zhang, Z. Wu, N. Li, and J. Chambers, "A novel adaptive kalman filter with inaccurate process and measurement noise covariance matrices," *IEEE Transactions on Automatic Control*, vol. 63, no. 2, pp. 594–601, 2017.

[17] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," ser. Handbook of Nonlinear Filtering, D. Crisan and B. Rozovsky, Eds. Oxford University Press, 2009.

[18] E. Özkan, V. Šmídl, S. Saha, C. Lundquist, and F. Gustafsson, "Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters," *Automatica*, vol. 49, no. 6, pp. 1566–1575, 2013.

[19] K. Berntorp and S. Di Cairano, "Process-noise adaptive particle filtering with dependent process and measurement noise," ser. Int. Conf. Decision and Control. Las Vegas, NV: IEEE, Dec 2016.

[20] 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 Transactions on Automatic Control*, vol. 62, no. 1, pp. 135–148, 2016.

[21] B. P. Maldonado, N. Li, I. Kolmanovsky, and A. G. Stefanopoulou, "Learning reference governor for cycle-to-cycle combustion control with misfire avoidance in spark-ignition engines at high exhaust gas recirculation–diluted conditions," *International Journal of Engine Research*, vol. 21, no. 10, pp. 1819–1834, 2020.

[22] P. Zhao, I. Kolmanovsky, and N. Hovakimyan, "Integrated adaptive control and reference governors for constrained systems with state-dependent uncertainties," *arXiv preprint arXiv:2208.02985*, 2022.

[23] K. Liu, N. Li, I. Kolmanovsky, D. Rizzo, and A. Girard, "Safe learning reference governor: Theory and application to fuel truck rollover avoidance," *Journal of Autonomous Vehicles and Systems*, pp. 1–19, 2021.

[24] A. Chakrabarty, K. Berntorp, and S. Di Cairano, "Learning-based parameter-adaptive reference governors," ser. Proc. American Control Conference (ACC). IEEE, 2020, pp. 956–961.

[25] K. Berntorp, A. Chakrabarty, and S. Di Cairano, "Vehicle rollover avoidance by parameter-adaptive reference governor," ser. Proc. of the 60th IEEE Conf. Decision and Control. IEEE, 2021.

[26] J. Chen, J. Huang, and Y. Sun, "Resistances and speed estimation in sensorless induction motor drives using a model with known regressors," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 2659–2667, 2018.

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

[28] A. Chakrabarty, E. Maddalena, H. Qiao, and C. Laughman, "Scalable bayesian optimization for model calibration: Case study on coupled building and hvac dynamics," *Energy and Buildings*, vol. 253, p. 111460, 2021.

[29] F. Gustafsson, *Statistical Sensor Fusion.* Lund, Sweden: Utbildningshuset/Studentlitteratur, 2010.

[30] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, vol. 23, no. 1, pp. 69–101, 1996.

[31] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1517–1531, 2011.

[32] A. Grønlund, L. Kamma, and K. G. Larsen, "Near-tight margin-based generalization bounds for support vector machines," ser. Proc. International Conference on Machine Learning. PMLR, 2020, pp. 3779–3788.

[33] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Contemporary Mathematics*, vol. 26, 1984.

[34] S. Di Cairano, U. V. Kalabić, and I. V. Kolmanovsky, "Reference governor for network control systems subject to variable time-delay," *Automatica*, vol. 62, pp. 77–86, 2015.

[35] R.-E. Fan, P.-H. Chen, C.-J. Lin, and T. Joachims, "Working set selection using second order information for training support vector machines." *Journal of machine learning research*, vol. 6, no. 12, 2005.

[36] 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.

[37] C. R. Carlson and J. C. Gerdes, "Optimal rollover prevention with steer by wire and differential braking," ser. ASME Int. Mechanical Engineering Congress and Exposition. ASME, 2003.

[38] S. Solmaz, M. Corless, and R. Shorten, "A methodology for the design of robust rollover prevention controllers for automotive vehicles: Part 2-active steering," ser. Proc. American Control Conference (ACC). AACC, 2007.

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

[40] K. Berntorp, A. Chakrabarty, and S. Di Cairano, "Vehicle center-of-gravity height and dynamics estimation with uncertainty quantification by marginalized particle filter," ser. Proc. American Control Conference (ACC). IEEE, 2021, pp. 160–165.

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

[42] 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.

[43] 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.

[44] S. Di Cairano and C. Danielson, "Indirect adaptive model predictive control and its application to uncertain linear systems," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 18, pp. 8678–8702, 2021.

[45] J.-H. Oh, H. S. Kim, and Y. M. Cho, "Adaptive reference governor for constrained linear systems," *Journal of Mechanical Science and Technology*, vol. 22, no. 1, pp. 61–69, 2008.