Data-Driven Computation of Robust Control Invariant Sets With Concurrent Model Selection

Yuxiao Chen, *Member, IEEE*, and Necmiye Ozay, *Senior Member, IEEE*

***Abstract*— Set invariance in the presence of uncertainty and disturbance is of central importance for the safety of control systems. This article proposes a data-driven method to compute an approximation of a minimal robust control invariant set (mRCI) from experimental data. For a given dynamical model with additive and multiplicative uncertainty, the proposed method is able to compute a polytopic mRCI with fixed complexity via linear programming (LP). Moreover, the method can be combined with model selection to enable mRCI computation directly from experiment data when the system dynamics are unknown. Specifically, given a model structure, our algorithm begins by identifying the set of admissible models with constraints extracted from the experimental data. Each model in the set of admissible models contains information about the nominal model and the characterization of the model uncertainties. Then, two iterative algorithms based on robust optimization are proposed to compute an mRCI while simultaneously searching for a model "optimal" with regard to the mRCI computation and the corresponding invariance-inducing controller. Finally, the method is demonstrated in an experiment with an autonomous vehicle lane-keeping control example.**

***Index Terms*— Automotive control, learning, robust control invariant (RCI) set, safety-critical control, system identification.**

I. Introduction

SAFETY is critical in many industrial applications of autonomy, such as autonomous/semiautonomous vehicles, robotics, and manufacturing. To achieve guaranteed safety, correct-by-construction control synthesis has attracted increasing interest over the past decade with the promise that, through rigorous reasoning of system behavior, the closed-loop system can be guaranteed to satisfy the design specifications. A fundamental concept related to safety specifications is robust control invariant (RCI) sets. By definition, if an initial condition lies within an RCI, then there exist control inputs that keep the state trajectory inside the set indefinitely, despite all possible disturbances and uncertainties. In addition to providing a safety certificate, set invariance can be used in a supervisory control structure on top of a legacy controller [1], thus guaranteeing safety with minimal intervention.

Our motivation comes from autonomous driving. In particular, we consider the vehicle lane-keeping problem where the goal is to ensure that the vehicle remains within the lane boundary and as close to the lane center as possible unless it intends to leave the lane, e.g., in the case of a lane change. One challenge in providing safety guarantees is that such guarantees rely on the existence of a precise model, whereas this article presents a data-driven algorithm that computes invariant sets starting from experimental data applicable in case a precise model does not exist. For instance, in the context of driving, loading of the vehicle, surface/road conditions, or weather can affect the ability to obtain a precise model. While our algorithm can be useful in a broad range of applications, we present experimental validation of our approach on the vehicle lane-keeping example on a rainy day and demonstrate its efficacy.

A. Background and Literature Review

Set invariance has been a central topic in control and dynamical systems since early days both in continuous time (e.g., Nagumo's theorem [2] and viability theory [3]) and discrete time [4]. Existing methods for computing invariant sets include, on the one hand, LMI-based Lyapunov type analysis [5]–[7] and sum of squares programming [8]–[10], which results in invariant sets with smooth boundaries, and on the other hand, Minkowski type methods [11]–[14], polytopic projection [1], [15], and linear programming (LP) [16], which result in polytopic invariant sets. Although polytopes are commonly used to represent invariant sets for discrete-time dynamical systems, the complexity of the polytopic representation grows quickly within many iterative algorithms. To overcome this problem, several techniques are proposed to compute low-complexity robust invariant sets (see [16]–[18]). In this article, inspired by the one-shot approach proposed in [16] for low-complexity invariant set computation for autonomous systems (i.e., systems without a control input), we propose an iterative algorithm that computes an RCI with constant representation complexity, where one can leverage the available control authority for enforcing invariance.

Manuscript received July 12, 2020; revised January 10, 2021; accepted March 8, 2021. Manuscript received in final form March 27, 2021. This work was supported by the NSF under Grant CNS-1239037. The work of Necmiye Ozay was also supported in part by the NSF under Grant ECCS-1553873 and ONR under Grant N00014-18-1-2501. Recommended by Associate Editor L. Fagiano. *(Corresponding author: Yuxiao Chen.)*

Yuxiao Chen is with the Department of Mechanical and Civil Engineering, California Institute of Technology (Caltech), Pasadena, CA 91106 USA (e-mail: chenyx@caltech.edu).

Necmiye Ozay is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: necmiye@umich.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCST.2021.3069759.

Digital Object Identifier 10.1109/TCST.2021.3069759

Two types of control invariant sets may be useful for control synthesis—the maximal control invariant set and a minimal control invariant set. Typically, the maximal control invariant set is related to the region of attraction with a limited control authority. The formal definition of the maximal control invariant set can be traced back to [4] under the name "infinite-time reachable set." The definition of the maximum is in the set inclusion sense, that is, every control invariant set within a compact subset of the state space is a subset of the maximal control invariant set. The existence and uniqueness of the maximal control invariant set are guaranteed under some mild assumptions. For linear discrete-time systems, the maximal control invariant set can be computed via polytopic projection [1], [19]. On the other hand, an mRCI describes how small an RCI set can be under disturbance and uncertainty. It can be used to provide safety and other performance guarantees when the objective is to closely follow a desired trajectory while staying inside the desired set or away from an unsafe set. In particular, when assume-guarantee reasoning [20], [21] is adopted, an mRCI is useful since it minimizes the bound on the states of one part of the system, thus reducing the disturbance due to dynamic coupling for other parts of the system. In general, however, a unique mRCI that is a subset of every RCI does not exist.

The computation of invariant sets depends on a model of the system, and uncertainty characterization is critical for RCI computation. Among the abovementioned methods, while some can handle modeling uncertainties and exogenous disturbances, they all assume that the model is given, both the nominal model and the uncertainty characterization. The problem, however, is that an *a priori* uncertainty characterization might be too loose or too tight, leading to control designs that may fail to satisfy the specifications due to underestimating the uncertainty or designs that are too conservative due to overestimating the uncertainty. In addition, if the environment or the dynamical system itself is changing, the assumption about the bound of uncertainty must be large enough to cover all possible changes, thus possibly rendering the synthesized controllers unnecessarily conservative or even infeasible.

Although, for some control applications, the model is built based on the underlying physical laws, the majority of engineering systems depend on system identification to obtain a practical model of the system dynamics. Even physics-based models require the analysis of experimental data for characterizing the uncertainty. The most classic system identification method is the least-squares regression, including many extensions that incorporate various filtering structures [22]. Control relevant identification, studied since the 1980s, is dedicated to model identification for control design. This type of methods includes $\mathcal{H}_\infty$ identification [23], generalized predictive control [24], and stochastic embedding [25]. However, the $\mathcal{H}_\infty$ identification and stochastic embedding approaches are for model identification in the frequency domain, and the generalized predictive control focuses on optimality rather than robustness. In terms of the identification of a model for an uncertain system that suits the need for correct-by-construction control synthesis, there is a gap to be filled. Another important line of work consists of set membership methods, which identifies the set of admissible model parameters via set intersection [26]. The majority of this type of method assumes a fixed bound on the uncertainty, which might be hard to obtain or even be nonexistent. System identification with unknown bounds for uncertainty is studied in [27]–[29] and in [30]. The latter article proposes an identification method that uses mixed-integer programming to identify a piecewise linear model with a bound on the disturbance for formal synthesis. The works in [31] and [32] discuss data-driven stability analysis centered around Lyapunov functions but without any consideration for control synthesis. What is clear, in all of the abovementioned articles, however, is that *system identification and control synthesis are undertaken separately*, and the identified model is not necessarily "optimal" for control synthesis.

B. Article Contributions and Organization

In this article, we present a framework based on robust LP for approximating a minimal RCI set while simultaneously selecting the optimal admissible uncertain model. An admissible model, which is not unique, is defined as a model that explains a finite measurement history. The novelty of the proposed method lies in the following. First, we extend the LP-based method in [16] to controlled systems. Second, the mRCI algorithm directly identifies the set of admissible models from data, thus not relying on a known model. Third, an optimal model for computing an mRCI is selected simultaneously by the proposed algorithm while computing an mRCI.

A conference version of this article appeared in [33]. The current version extends the conference version in the following respects: 1) the algorithm is applied to a practical lane-keeping problem and validated with real vehicle experiments; 2) details on implementation issues, such as the computation complexity and the hyperplane orientation selection process, are discussed; and 3) some of the proofs missing in the conference version are fully presented.

The remainder of this article is organized as follows. We first present the parameterization and construction of the set of admissible models in Sections II and III. Section IV presents the main robust optimization method for computing an mRCI. Section V discusses the practical issues regarding the implementation and presents experimental results. Finally, we conclude this article in Section VI.

C. Nomenclature

$\mathbb{R}$ is the set of real numbers, $\mathbb{R}^n$ is the n-dimensional Euclidean space, and $\mathbb{R}^n_{>0}$ and $\mathbb{R}^n_{\geq 0}$ are the open (and closed) positive orthants of $\mathbb{R}^n$. For two vectors $x, y \in \mathbb{R}^n$, the inequality $x \leq y$ is defined elementwise: $x \leq y \Leftrightarrow y - x \in \mathbb{R}^n_{\geq 0}$. $\mathbb{Z}$ and $\mathbb{Z}_{\geq 0}$ represent the sets of integers and nonnegative integers, respectively. $\mathbb{Z}_{1:T}$ denotes the sequence $1, 2, \ldots, T$ of natural numbers. For $x \in \mathbb{R}^n$, $\|x\|$ by default denotes the 2-norm of x, and $|x| = [|x_1|, \ldots, |x_n|]^\intercal$ is the entrywise absolute value of x. For a matrix A, A_i denotes its ith row, A^j denotes its jth column, A_{ij} denotes the entry on the ith row, jth column, and $|A|$ denotes its entrywise absolute value. $x(t_1 : t_2)$ denotes a sequence of vectors, indexed by time, starting from $t_1 \in \mathbb{Z}$

and ending at $t_2 \in \mathbb{Z}$. For simplicity, we use $\mathcal{P}(P, q)$ to denote the polyhedron $\{x \mid Px \leq q\}$.

II. Model Parameterization

In this article, we consider discrete-time linear models with uncertainty

$$x^+ = \hat{A}x + \hat{B}u + \hat{E}d + \tilde{A}x + \tilde{E}d + e \tag{1}$$

where $x \in \mathbb{R}^n$ is the state of the system, with x^+ being the state at the next sampling time, $u \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input, $d \in \mathcal{D} \subseteq \mathbb{R}^l$ is the exogenous measured disturbance, $\hat{A}$, $\hat{B}$, and $\hat{E}$ are the nominal model matrices, $\tilde{A}$ and $\tilde{E}$ are the matrices for the multiplicative uncertainty with respect to x and d, and $e \in \mathbb{R}^n$ is the additive uncertainty. Here, we assume that $\hat{B}$ is fixed, no multiplicative uncertainty is allowed for the input matrix B, and we shall explain the reason in Section IV. This assumption is not too restrictive since one can first use other existing system identification methods to obtain a $\hat{B}$ before applying the algorithm proposed in this article, and the uncertainty associated with the input dynamics can always be lumped into other uncertainty terms.

Remark 1: The model structure in (1) can be used to represent nonlinear systems as well with the proper choice of the uncertainty terms to cover the nonlinearity. However, if the required uncertainty terms are too large, the proposed algorithm might fail to find an RCI set.

Note that (1) can be represented by the standard linear parameterization form by letting

$$\begin{aligned}\varphi_i &= \left[x^\intercal, d^\intercal\right]^\intercal \in \mathbb{R}^{n+l}\\ \hat{\theta}_i &= \left[\hat{A}_i, \hat{E}_i\right]^\intercal \in \mathbb{R}^{n+l}\\ \tilde{\theta}_i &= \left[\tilde{A}_i, \tilde{E}_i\right]^\intercal \in \mathbb{R}^{n+l}\end{aligned}$$

where the subscript i denotes the ith row of the matrices. Then, it can be seen that (1) is simply n uncertain linear parameterizations, one per state dimension, stacked together. Taking the ith dimension as an example and defining $z_i = x_i^+ - \hat{B}_i u$, we have

$$z_i = \varphi_i^\intercal \hat{\theta}_i + \varphi_i^\intercal \tilde{\theta}_i + e_i. \tag{2}$$

For the uncertainty characterization, we use (unknown) hyperboxes to bound the uncertainties

$$|\tilde{\theta}_i| \leq \mathbf{\Omega}_M^i, \quad |e_i| \leq \mathbf{\Omega}_A^i. \tag{3}$$

That is, the bounds $\mathbf{\Omega}_M^i \in \mathbb{R}_{\geq 0}^{n+m+l}$ and $\mathbf{\Omega}_A^i \in \mathbb{R}_{\geq 0}$ are part of the variables to be identified from data, unlike the majority of the set membership methods, which assumes that the bounds for uncertainties are known. An uncertain linear model is determined by the value of $[\hat{\theta}, \mathbf{\Omega}_M, \mathbf{\Omega}_A]$, which contains the information of both the nominal model and the uncertainty characterization.

In what follows, we assume that u and d belong to bounded polytopes, known *a priori*

$$\begin{aligned}d &\in \mathcal{D} \doteq \mathcal{P}(G, g)\\ u &\in \mathcal{U} \doteq \mathcal{P}(R, r).\end{aligned} \tag{4}$$

This assumption is satisfied in many practical problems since u and d are often determined by system specifications or physics; for example, in the vehicle lane-keeping case, the bound on steering angle is determined by the tire friction limit, and the bound on road curvature is given as a specification of the environment. The proposed mRCI algorithm can be extended to the case where the range of u and d depends on state x

$$\begin{aligned}[G_1\ G_2]\left[x^\intercal, d^\intercal\right]^\intercal &\leq \bar{g}\\ [R_1\ R_2]\left[x^\intercal, u^\intercal\right]^\intercal &\leq \bar{r}.\end{aligned} \tag{5}$$

However, for simplicity, we stick to the case shown in (4). The extension to the case with state-dependent bounds is straightforward.

While determining the values for $[\hat{\theta}, \mathbf{\Omega}_M, \mathbf{\Omega}_A]$ would solve the identification problem, in many cases, the model to be identified has additional structures. For example, due to the underlying physics, some of the model parameters may be known to be a constant, with some entries of the system matrices possibly being linearly dependent. To impose additional structure, we assume that the model parameters are affinely parameterized by a hyperparameter $\pi \in \mathbb{R}^s$ (π contains the independent parameters of the model), and we have

$$\begin{aligned}\hat{\theta} &= \hat{\mathbf{\Theta}}(\pi)\\ \mathbf{\Omega}_M &= \mathbf{\Omega_M}(\pi)\\ \mathbf{\Omega}_A &= \mathbf{\Omega_A}(\pi)\end{aligned} \tag{6}$$

where we use **bold font** to denote the predefined affine mapping from π to the model parameters, e.g., $\hat{A} = \hat{\mathbf{A}}(\pi)$. When no additional structure is imposed, these mappings can be chosen to be the trivial ones.

III. Identification of Admissible Models

In this section, we present a procedure for generating the set of admissible models from the measurement data.

Given a sequence of measurements $x(1:T+1)$, the output and regressor for time step t are defined as

$$z_i(t) = x_i^+(t) - \hat{B}_i u, \quad \varphi_i(t) = \left[x(t)^\intercal, d(t)^\intercal\right]^\intercal. \tag{7}$$

A model $[\hat{\theta}, \mathbf{\Omega}_M, \mathbf{\Omega}_A]$ is called admissible if for $t = 1, 2, \ldots, T$

$$\begin{aligned}&\exists e(t), \tilde{\theta}(t), \quad \text{s.t. } |e(t)| \leq \mathbf{\Omega}_A,\ |\tilde{\theta}(t)| \leq \mathbf{\Omega}_M\\ &z(t) = \left(\hat{\theta} + \tilde{\theta}(t)\right)^\intercal \varphi(t) + e(t).\end{aligned} \tag{8}$$

Note that since, in (3), the uncertainties are assumed to be bounded by hyperboxes, condition (8) is equivalent to the following constraint in the parameter space:

$$|z(t) - \varphi(t)^\intercal \hat{\mathbf{\Theta}}(\pi)| \leq |\varphi(t)|^\intercal \mathbf{\Omega_M}(\pi) + \mathbf{\Omega_A}(\pi). \tag{9}$$

Therefore, the set of admissible models can be represented as a polyhedral set of admissible parameter values π:

$$\Sigma = \left\{ \pi \mid \forall t \in \mathbb{Z}_{1:T}, |z(t) - \varphi(t)^\intercal \hat{\mathbf{\Theta}}(\pi)| \leq |\varphi(t)|^\intercal \mathbf{\Omega_M}(\pi) + \mathbf{\Omega_A}(\pi) \right\}. \tag{10}$$

In Fig. 1, we conceptually demonstrate the relation between an admissible uncertain model as described above and a simple linear regression model. Given a data set, depicted as the black dots, the center red line represents the nominal model. On top of that, the uncertain model on the right introduces the bound on additive uncertainty (the parallel blue dotted lines) and

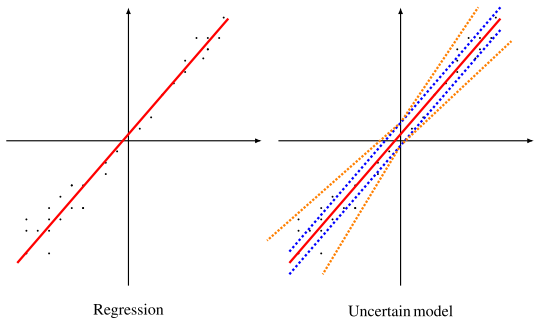

Fig. 1. Comparison of regression and the uncertainty model used in this work.

the bound on multiplicative uncertainty (the orange radiating dotted lines). With additive and multiplicative uncertainties, the model on the right covers all data points, and therefore, it is an admissible model.

If Σ is nonempty, then all models in Σ explain the measurement data. In fact, under mild assumptions, Σ is guaranteed to be nonempty by essentially making the additive uncertainty bound large enough. Naturally, there is a tradeoff between different types of uncertainty. When the additive uncertainty bound is large, the bound on multiplicative uncertainty can be smaller, and vice versa. This is the direct result of (9). The set of admissible models Σ gives the domain from which the model should be selected. Among the admissible models, the one deemed "the best" depends on how the model is to be used. If the goal is to find a model with the least-squared error, then the least-squares regression gives the best model, with corresponding uncertainty characterizations. However, since our goal is to compute an mRCI, the incorporation of the model selection process into the mRCI computation, as shown in Section IV, may result in a more desirable invariant set.

IV. Robust LP for mRCI

In this section, an iterative algorithm based on robust LP, which simultaneously selects an optimal model and approximates an mRCI, is proposed. Before presenting the algorithm, we briefly review the robust optimization approach, which is the main computation engine for the proposed mRCI algorithm.

A. Robust Optimization

Robust optimization is widely studied in the operation research field. We start by presenting a lemma that adapts the results in [34] by accounting for a bilinear term needed in our problem.

Lemma 1: Consider the following robust LP problem:

$$\begin{aligned} &\min_{\alpha} \ J(\alpha) \\ &\text{s.t. } \forall \beta \in \mathcal{P}(F, f) \\ &\quad H_1^i \beta + \alpha^\intercal H_2^i \beta + H_3^i \alpha \le h^i, \quad i = 1, \ldots, M \end{aligned} \tag{11}$$

where α is the decision variable, $J(\cdot)$ is a convex cost function, β is the uncertain variable, $\mathcal{P}(F, f)$ is the set describing uncertainty, and H_1^i, H_2^i, and H_3^i are constant matrices of appropriate dimensions. The robust optimization problem in (11) is equivalent to a standard convex optimization problem of the form

$$\begin{aligned} &\min_{\alpha, \lambda} \ J(\alpha) \\ &\text{s.t. } H_3^i \alpha + \left(\lambda^i\right)^\intercal f \le h^i \\ &\quad H_1^i + \alpha^\intercal H_2^i = \left(\lambda^i\right)^\intercal F \\ &\quad \lambda^i \ge 0, \quad i = 1, \ldots, M. \end{aligned} \tag{12}$$

See Appendix VI for the proof.

B. Background on Minimal RCI Sets

Next, we review some of the backgrounds on minimal RCI sets. First, we start with the definition of an RCI set.

Definition 1: A set $\mathcal{S} \subseteq \mathbb{R}^n$ is called RCI for the system described by (1), (3), and (4) if there exists a control strategy $\mu : \mathbb{R}^n \times \mathcal{D} \to \mathcal{U}$ such that, for all $d \in \mathcal{D}$ and for all $x \in \mathcal{S}$, we have $x^+ \in \mathcal{S}$ with $u = \mu(x, d)$ under all possible uncertainty given by (3).

As mentioned in Section I, the existence of a minimal control invariant set in the set inclusion sense is not guaranteed. One way to get around this difficulty is to define an mRCI with respect to the set inclusion partial order.

Definition 2: An RCI set $\mathcal{S}$ is a minimal RCI set (mRCI) if there does not exist an $\mathcal{S}' \subsetneq \mathcal{S}$, s.t. $\mathcal{S}'$ is an RCI set.

However, even with this definition, finding an mRCI is nontrivial. Typically, one tries to find an (approximate) mRCI by minimizing a certain measure of size, such as volume [14], [17]. We propose a method that computes a polytopic RCI that minimizes a linear objective function.

We draw inspiration from [16], where the author proposed a one-step LP approach to compute a robust invariant set for an autonomous system (i.e., system without control). The key idea is to fix the orientation of the candidate supporting hyperplanes that define a polytopic invariant set. Since the method in [16] can deal only with autonomous systems, for systems that are not open-loop stable, the problem becomes infeasible. One can design a feedback controller and search for the invariant set of the closed-loop, but there is no guarantee that the feedback input respects the input bounds for all states inside the RCI. Moreover, the prechosen feedback controller reduces the set of hyperplane orientations that would lead to an RCI. In addition, the uncertainty is assumed to be purely additive in [16], and no measured disturbance is included. We adopt the idea of fixing the hyperplane orientation and propose an iterative approach based on robust LP, where we search for the feedback control gains and mRCI at the same time, while respecting all the input constraints and providing robustness against the uncertainty model under consideration.

C. One-Step Propagation

Our method begins by choosing a set of L hyperplanes with fixed orientation P_i and varying offset q_i, $i = 1, \ldots, L$. The selection of the hyperplane orientations is application-dependent, and we shall present the heuristic that we use for the hyperplane orientation selection in Section V-A. Let $P = [P_1^\intercal, P_2^\intercal, \ldots, P_L^\intercal]^\intercal$ and $q = [q_1, \ldots, q_L]^\intercal$. Without loss of generality, assume that $\|P_i\| = 1$. If $\mathcal{S} = \mathcal{P}(P, q)$

has a nonempty interior, then P_i is the normalized normal vector pointing outward the corresponding supporting hyperplane.

Assumption 1: The hyperplanes are chosen such that $\{x \mid Px \leq \mathbf{1}_L\}$ is a compact set, where $\mathbf{1}_L \in \mathbb{R}^L$ denotes the column vector consisting of all ones.

One can show that $Px \leq \mathbf{1}_L$ is compact if and only if $Px \leq q$ is compact for all $q > 0$. Assumption 1 simply guarantees that the set $\mathcal{S} = \mathcal{P}(P, q)$ is compact for all $q > 0$.

The offset q is initialized with $q^0 \in \mathbb{R}^n_{>0}$ so that the origin is contained in $\mathcal{P}(P, q^0)$. Given a polytope $\mathcal{S} = \mathcal{P}(P, q)$, we consider the following one-step propagation that searches for $\mathcal{S}^+ = \mathcal{P}(P, q^+)$ that contains all possible x^+:

$$\begin{aligned} \min_{q^+} \ & c^\intercal q^+ \\ \text{s.t. } & \forall x \in \mathcal{P}(P, q) \quad \forall d \in \mathcal{D} \\ & \exists u \in \mathcal{U}, \quad \text{s.t. } \forall |e| \leq \Omega_A \quad \forall |\tilde{\theta}| \leq \Omega_M \\ & x^+ \in \mathcal{P}\big(P, q^+\big). \end{aligned} \tag{13}$$

The set $S^+ = \mathcal{P}(P, q^+)$ satisfies the following condition: for any $x \in \mathcal{S}$ and $d \in \mathcal{D}$, there exists $u \in \mathcal{U}$ such that all possible x^+'s under u are contained in $\mathcal{S}^+$. It is clear that, if $\mathcal{S}^+ \subseteq \mathcal{S}$, $\mathcal{S}$ is control invariant. The cost $c^\intercal q^+$ represents a measure of the size of $\mathcal{S}^+$, which we try to minimize.

Remark 2: The choice of the cost function $c^\intercal x$ is not unique. The simplest choice is to set $c = \mathbf{1}_L$. If c_i is set to be the area of the facet corresponding to P_i, then (13) is minimizing the linear approximation of the volume of mRCI although the computation of the area without knowing the polytope itself is difficult. One can also set c such that a particular dimension of the mRCI is penalized more than others.

Next, we discuss a few simplifications so that the one-step propagation is solvable by convex optimization. First, as mentioned at the beginning of this section, there is no minimum RCI that is a subset of every RCI. Therefore, the RCI obtained depends on a specific control strategy. For the linear discrete-time system discussed in this article, we impose the following control structure:

$$u = K_{\text{ff}}^\intercal d + K_{\text{fb}}^\intercal x \tag{14}$$

where K_{ff} and K_{fb} are constant matrices, representing the feedforward and feedback gain, respectively.

Remark 3: Since K_{ff} and K_{fb} are part of the optimization variables, we need to fix $\hat{B}$ without multiplicative uncertainty on B to get rid of the cross product terms between the controller gain and the input matrix. This simplification is possible since we can always lump the actuation uncertainty into other uncertainty terms with overapproximations.

Second, the one-step propagation should be robust against the model uncertainty, i.e., $\mathcal{S}^+$ should contain all possible x^+ under the uncertain model. This is enforced by considering the worst case uncertainty, captured by the "for all" quantifiers for e and $\tilde{\theta}$ in (13). Since the uncertainty bounds are assumed to be hyperboxes, these quantifiers can be eliminated by observing that

$$\begin{aligned} \max_{|\tilde{A}| \leq \Omega_{\tilde{A}}} P_i \tilde{A} x &= \max_{|\tilde{A}| \leq \Omega_{\tilde{A}}} \text{Tr}\big(|x P_i||\tilde{A}|\big) = |P_i|\Omega_{\tilde{A}}|x| \\ \max_{|\tilde{E}| \leq \Omega_{\tilde{E}}} P_i \tilde{E} d &= \max_{|\tilde{E}| \leq \Omega_{\tilde{E}}} \text{Tr}\big(|d P_i||\tilde{E}|\big) = |P_i|\Omega_{\tilde{E}}|d| \\ \max_{|e| \leq \Omega_A} P_i e &= |P_i|\Omega_A \end{aligned} \tag{15}$$

where $\Omega_{\tilde{A}} \in \mathbb{R}^{n \times n}$ and $\Omega_{\tilde{E}} \in \mathbb{R}^{n \times l}$ are the bounds on $|\tilde{A}|$ and $|\tilde{E}|$ induced from Ω_M, respectively. Note that the expressions in (15) are not yet linear in x and d due to the constraints with absolute values, which can be converted to linear constraints using standard LP techniques of enumerating the orthants. However, for the sake of keeping the notations simple, we will keep the absolute value form for the remainder of this article.

With these simplifications, the one-step propagation problem takes the following robust optimization form:

$$\begin{aligned} \min_{K_{\text{ff}}, K_{\text{fb}}, \pi, q^+} \ & c^\intercal q^+ \\ \text{s.t. } & \pi \in \Sigma \quad \forall x \in \mathcal{P}(P, q) \ \forall d \in \mathcal{D} \\ & P\Big(\hat{\mathbf{A}}(\pi)x + \hat{B}\big(K_{\text{ff}}^\intercal d + K_{\text{fb}}^\intercal x\big) + \hat{\mathbf{E}}(\pi)d\Big) \\ & + |P|\Omega_{\tilde{\mathbf{A}}}(\pi)|x| + |P|\Omega_{\tilde{\mathbf{E}}}(\pi)|d| + |P|\Omega_{\mathbf{A}}(\pi) \leq q^+ \\ & K_{\text{ff}}^\intercal d + K_{\text{fb}}^\intercal x \in \mathcal{U}. \end{aligned} \tag{16}$$

The optimization problem in (16) concurrently searches for: 1) a controller that satisfies the input bound for all $x \in \mathcal{S}$ and all $d \in \mathcal{D}$; 2) an admissible model from Σ; and 3) a set $\mathcal{S}^+$ that contains all possible x^+ under the controller and the model selected.

The problem in (16) is a robust linear program in the sense that the constraints have to be satisfied for all $x \in \mathcal{S}$ and all $d \in \mathcal{D}$. In particular, if we take $\alpha = [K_{\text{ff}}, K_{\text{fb}}, q^+, \pi]$ and $\beta = [x, d]$, it follows the robust optimization form of (11). Therefore, by Lemma 1, it can be transformed to a LP problem and solved efficiently.

D. Iterative Algorithm

With the one-step propagation efficiently solvable via LP, in this section, we devise algorithms to find an RCI set $\mathcal{S}$, i.e., a set $\mathcal{S}$ that satisfies $\mathcal{S}^+ \subseteq \mathcal{S}$. In the invariant set computation literature [1], [14], [19], two iterative algorithms can be found—the inside-out algorithm and the outside-in algorithm. With the robust LP-based one-step propagation, the inside-out algorithm is used to solve for an RCI, and the outside-in algorithm is used to shrink a known RCI to a smaller size.

1) Inside-Out Algorithm: The inside-out algorithm starts with a small initial $\mathcal{S}$, iteratively solves for $\mathcal{S}^+$ with the one-step propagation, and replaces $\mathcal{S}$ with $\mathcal{S}^+$, until $\mathcal{S}^+ \subseteq \mathcal{S}$ is satisfied. The overall procedure is given in **Algorithm** 1, where $0 < \epsilon \ll 1$ is a small constant.

Proposition 1: If **Algorithm** 1 terminates, $\mathcal{S} = \mathcal{P}(P, q)$ is an RCI set.

Proof: By construction, $\mathcal{S}^+ = \mathcal{P}(P, q^+ - \epsilon \mathbf{1}_L)$ contains all possible x^+ with $x \in \mathcal{S}$, $d \in \mathcal{D}$, and since $q^+ \leq q + \epsilon \mathbf{1}_L$, $\mathcal{S}^+ \subseteq \mathcal{S}$; therefore, $\mathcal{S}$ is an RCI. ■

With $\epsilon > 0$, the algorithm searches for an $\mathcal{S}^+$ slightly larger than that in (16) so that we can allow ϵ tolerance

Algorithm 1 Inside-Out Algorithm for mRCI

procedure RCI-IO(Σ, P, q^0, $\mathcal{D}$, $\mathcal{U}$, ϵ)
  $q^+ \leftarrow q^0$
  do
    $q \leftarrow q^+$
    Find $\left[q^+, \pi, K_{ff}, K_{fb}\right]$ s.t. $\pi \in \Sigma$,
    $\forall x \in \mathcal{P}(P,q), \forall d \in \mathcal{D}, K_{ff}d + K_{fb}x \in \mathcal{U}$,
    $x^+ \in \mathcal{P}(P, q^+ - \epsilon \mathbf{1}_L)$
  while $q^+ \leq q + \epsilon \mathbf{1}_L$
  return $\left[q, \pi, K_{ff}, K_{fb}\right]$
end procedure

Algorithm 2 Outside-In Algorithm for mRCI

procedure RCI-OI(Σ, P, q^0, $\mathcal{D}$, $\mathcal{U}$, ϵ)
  $q^+ \leftarrow q^0$
  do
    $q \leftarrow q^+$
    Find $\left[q^+, \pi, K_{ff}, K_{fb}\right]$ s.t. $\pi \in \Sigma, q^+ \leq q$
    $\forall x \in \mathcal{P}(P,q), \forall d \in \mathcal{D}, K_{ff}d + K_{fb}x \in \mathcal{U}$,
    $x^+ \in \mathcal{P}(P, q^+)$
  while $\left\|q^+ - q\right\| \geq \epsilon$
  return $\left[q^+, \pi, K_{ff}, K_{fb}\right]$
end procedure

for the termination condition $\mathcal{S}^+ \subseteq \mathcal{S}$, which accelerates the convergence of the algorithm.

The optimal choice of the hyperparameter π changes in every iteration, allowing the algorithm to choose a model based on $\mathcal{S}$ in each iteration. For a small $\mathcal{S}$, more uncertainty may be lumped into multiplicative terms since $|x|$ is relatively small; for a large $\mathcal{S}$, a smaller multiplicative uncertainty bound may be preferred, which is automatically adjusted by the optimization process.

2) Outside-In Algorithm: On the other hand, if an initial RCI is known for some admissible model, the outside-in algorithm can further shrink the initial RCI with a convergence guarantee. The algorithm iteratively solves for $\mathcal{S}^+ \subseteq \mathcal{S}$ and replaces $\mathcal{S}$ with $\mathcal{S}^+$ until $\mathcal{S}$ cannot be further shrunken.

Algorithm 2 shows the outside-in algorithm, which is similar to the inside-out algorithm but differs in two ways. First, the one-step propagation has an additional constraint $q^+ \leq q$, which ensures that $S^+ \subseteq S$. Second, the termination condition is on the norm of the difference between q and q^+.

Theorem 1: If, for a certain admissible model π^0, a polytopic RCI $\mathcal{P}(P, q^0)$ is known and satisfies the following: 1) it can be rendered invariant with a controller of the form shown in (14) and 2) $\forall x \in \mathcal{P}(P, q^0), d \in \mathcal{D}$, $K_{\text{ff}}^{\intercal} d + K_{\text{fb}}^{\intercal} x \in \mathcal{U}$, then the outside-in algorithm is guaranteed to terminate in finite time.

See Appendix VI for proof.

A comparison of the properties of the two algorithms is shown in Table I. In summary, the inside-out algorithm solves for an RCI from scratch, whereas the outside-in algorithm shrinks the size of a known RCI.

TABLE I
COMPARISON OF ITERATIVE ALGORITHMS

	Initialization	Termination
Inside-out	Arbitrary	Not guaranteed
Outside-in	Need an RCI to start with	Guaranteed

E. Computational Complexity

Due to the iterative nature of the overall approach, it is hard to analyze the computational complexity. In what follows, we briefly talk about the complexity of the linear programs solved in each iteration. The complexity largely depends on two factors: the dimension of multiplicative uncertainty $\tilde{\theta}$ and the size of the hyperplane pool L. The impact of the dimension of $\tilde{\theta}$ is due to (15), which requires the enumeration of all the orthants of the $\tilde{\theta}$-space. The number of constraints, thus, scales exponentially with the dimension of $\tilde{\theta}$. However, in practice, the dimension of $\tilde{\theta}$ is usually determined by the problem setup. Therefore, the hyperplane pool size L is the dominating parameter affecting the complexity of the robust linear program in the mRCI algorithm. The dual variable λ's dimension is roughly L^2 (ignoring the constraint enforcing $d \in \mathcal{D}$), and the number of equality and inequality constraints in the robust linear program also scales quadratically with L. To conclude, the variable dimension of the linear program solved in the mRCI algorithm scales quadratically with L; the number of constraints of the linear program is roughly $\sim \mathcal{O}(2^{[\tilde{\theta}]} \cdot L^2)$, where $[\tilde{\theta}]$ is the dimension of $\tilde{\theta}$. The complexity per iteration, then, follows standard complexity results for LP, which is polynomial in the number of variables and constraints.

V. IMPLEMENTATION AND EXPERIMENTAL VALIDATION

The overall procedure from mRCI computation to vehicle deployment is depicted in Fig. 2. The algorithm starts with collecting data, possibly under some legacy controller if the system is open-loop unstable. The state and input traces are then used to identify the set of admissible models given the model structure by extracting linear constraints from the measurement data. The hyperplane orientation selection step selects the orientations of the hyperplanes of the polytopic invariant set, i.e., the P matrix, which is explained in detail in Section V-A. With the hyperplane orientations selected, the set of admissible models Σ is then used to run the iterative one-step propagation algorithm (either the inside-out algorithm in Algorithm 1 or the outside-in algorithm shown in Algorithm 2) to obtain the mRCI and the controller enforcing this mRCI, which is then implemented on the vehicle in our experiments.

In what follows, we first discuss a practical issue: hyperplane orientation selection. Then, we present implementation details on the vehicle lane-keeping problem and the experimental results.

A. Selection of Hyperplane Orientations

The selection of the hyperplane orientations for the RCI is critical to the success of the mRCI computation. A large number of hyperplanes may cause a heavy computation load, while a small number of hyperplanes may cause undue conservatism.

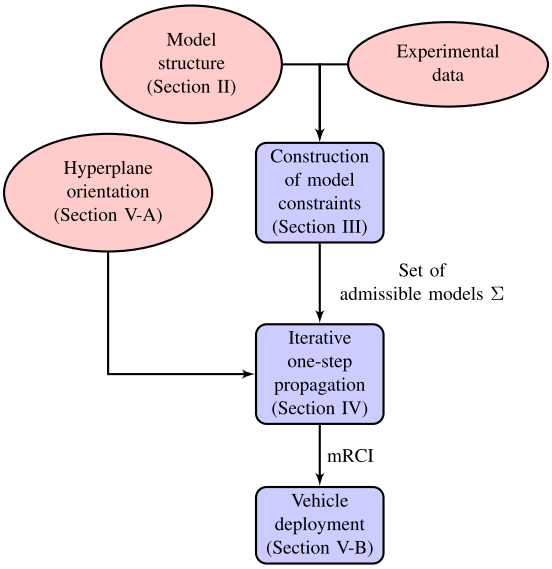

Fig. 2. Flowchart showing the overall procedure from mRCI computation to vehicle deployment.

Unfortunately, it is not yet clear how to select the hyperplane orientations systematically except for the low-dimensional state space. It should be noted that the mRCI algorithm automatically selects the hyperplanes that are useful for the mRCI via minimizing the cost function; here, we are simply selecting the pool of hyperplane orientations for the mRCI algorithm. For a 1-D state space, there is only one choice of hyperplane orientation, that is, $P = [1; -1]$. In [16], the author shows a 2-D example, where the L hyperplanes are selected so that $\mathcal{P}(P, \mathbf{1}_L)$ is an L-sided regular polygon. However, this approach is not applicable when the state space is of higher dimensions. There exist random sampling strategies for choosing hyperplane orientations in higher dimensional space; for example, using quaternions [35], [36], but uniform sampling is, in general, inefficient for computing RCI since the shape of RCI is strongly influenced by the dynamical system. Certain dimensions or orientations may require a denser patch, while others can be covered with a sparser set of hyperplane orientations. We develop a particle simulation approach to generate the hyperplane orientations for the RCI computation.

To be specific, an uncertain model is identified from the experimental data, with the nominal model being the least-squares regression result and having only additive uncertainty

$$x^+ = Ax + Bu + Ed + e, \quad |e| \leq \Omega_A. \tag{17}$$

Note that, when fixing the nominal model to be the least-squares regression result, the minimum bounds for additive uncertainty can be uniquely identified. A default controller is then used to keep the system stable. As an example, in the lane-keeping case to be discussed later, a saturated LQR controller is used as the default controller

$$u = \mathbf{Sat}_{\mathcal{U}}(Kx) \tag{18}$$

where $\mathbf{Sat}_{\mathcal{U}}$ is the saturation function under the input set $\mathcal{U}$.

Then, M particles with random initial state are simulated following this control law. The external disturbance and additive uncertainty are chosen in the following fashion: with probability λ, d, and e being chosen such that $\|x^+\|$ is maximized; with probability $1 - \lambda$, d, and e being uniformly sampled from their domains.

Fig. 3. Experimental platform at Mcity.

After T sampling times, the convex hull of the M particles is computed, denoted as $\mathcal{H}$

$$\mathcal{H} = \{x | Hx \leq \mathbf{1}\}. \tag{19}$$

$\mathcal{H}$ can be roughly seen as an invariant set of the dynamical system under the default controller, which captures the shape information determined by the dynamical system. Then, P is chosen to be $[H, -H]$.

Remark 4: If the specification requires that the RCI is inside a polytope given as $\mathcal{P}(P_0, q_0)$, it is straightforward to encode this constraint. First, the hyperplane orientation matrix P is augmented as $P = [P_0; P]$. Then, upper bounds are enforced on q^+ on the corresponding dimensions; that is, suppose that P_0 corresponds to the first N_0 hyperplanes, and then, $q^+_{1:N_0} \leq q_0$ is enforced in the optimization. This would ensure that $\mathcal{P}(P, q^+) \subseteq \mathcal{P}(P_0, q_0)$.

B. mRCI for Vehicle Lane-Keeping Control

An mRCI is relevant for lane keeping since the objective is to keep the vehicle inside the lane boundaries, and a tight tracking of the lane center is preferred. To evaluate the proposed mRCI algorithm in this context, experiments were conducted in the Mcity test facility of the University of Michigan on the OpenAV platform, as shown in Fig. 3, which is an autonomous vehicle test platform built based on a Lincoln MKZ sedan [37]. First, measurement data were collected when the vehicle followed the lane with a default lane-keeping algorithm that has no performance or safety guarantees. The data were then used to compute an mRCI with the proposed method. Finally, the computed mRCI was validated by experiments on the OpenAV platform.

1) Model Structure: Recent work has demonstrated the utility of linear models obtained via system identification for vehicle lateral dynamics [38]. While it is typical to approximate the vehicle lateral dynamics by a linear model, uncertainty exists in such models due to factors, such as nonlinear tire properties and parameter uncertainties. Moreover, the nominal

model and the model uncertainty vary with vehicle conditions and environmental factors, making them hard to model or predict.

The model to be identified for the lateral dynamics is called the lateral-yaw model, or bicycle model, which has four states

$$x = \left[y, v_y, \psi, r\right]^\mathsf{T} \tag{20}$$

where y is the lateral displacement from the lane center, v_y is the sideslip velocity, ψ is the heading angle with respect to the lane direction, and r is the yaw rate. The model is linear, yet the model parameters may change with longitudinal speed v_x, road conditions, and vehicle conditions, such as mass, inertia, and tire properties. A linear discrete-time model with uncertainty is to be learned from data to describe the dynamics. The input is the steering angle on the front axle δ_f; the measured disturbance is road curvature r_d.

To reduce the complexity of the uncertainty characterization, a certain structure is imposed on the model based on the properties of the dynamics. The dynamics for $\dot{y}$ and $\dot{\psi}$ are essentially integrators, described by the following differential equation:

$$\begin{bmatrix} \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & v_x \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ \psi \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r + \begin{bmatrix} 0 \\ -1 \end{bmatrix} r_d. \tag{21}$$

After time-discretizing the model in (21), the entries in the nominal model that corresponds to the parameters from the kinetic relationship are fixed. As a result, all entries of $\hat{E}$ are now fixed (the first and third entries of $\hat{E}$ are determined by the discretization of (21) and rest are zero). To prevent overfitting, when there is a correlation between the entries of the nominal model matrix, it is not necessary to parameterize all the entries independently. $\hat{A}$ is parameterized as follows:

$$\begin{aligned} \hat{A} &= \sum_{i=1}^{n_1} \pi_1^i \bar{A}_i + \hat{A}_0 \\ \hat{A}_0 &= \begin{bmatrix} 1 & 0 & v_x T_s & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \bar{A}_i = \begin{bmatrix} 0 & * & 0 & * \\ * & * & * & * \\ 0 & 0 & 0 & 0 \\ * & * & * & * \end{bmatrix} \end{aligned} \tag{22}$$

where $\{\bar{A}_i\}$ is a basis for $\hat{A} - \hat{A}_0$. This linear parameterization reduces the parameters for $\hat{A}$ from 10 (ten nonzero entries in $\bar{A}_i$) to n_1.

Remark 5: The basis was obtained with the following procedure. First, multiple simulations under different scenarios were conducted on the high fidelity simulator Carsim,[1] each generating an $\hat{A}$ via least-squares regression of the measurement following the structure shown in (22). Then, the basis vectors are selected with the principal component analysis (PCA) [39] on the multiple $\hat{A}$ obtained from each simulation run. The number n_1 of basis vectors is chosen to be 5.

Since the kinetic relationship in (21) is precise, the multiplicative uncertainty terms on the corresponding entries of the nominal model are assumed to be zero; additive uncertainty terms are used to represent the sensor error. Then, the overall

[1]CarSim is a vehicle simulation package that is widely used in industry. It is a registered trademark of Mechanical Simulation Corporation, Ann Arbor, MI, USA.

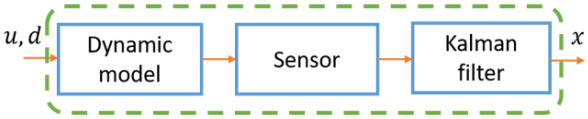

Fig. 4. Generalized dynamical system.

parameterization of the vehicle lateral dynamic model appears as

$$\begin{aligned} \begin{bmatrix} y^+ \\ v_y^+ \\ \psi^+ \\ r^+ \end{bmatrix} &= \hat{\mathbf{A}}(\pi) \begin{bmatrix} y \\ v_y \\ \psi \\ r \end{bmatrix} + \hat{B}\delta_f + \hat{E} r_d \\ &\quad + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \tilde{A}_{22} & 0 & \tilde{A}_{24} \\ 0 & 0 & 0 & 0 \\ 0 & \tilde{A}_{42} & 0 & \tilde{A}_{44} \end{bmatrix} \begin{bmatrix} y \\ v_y \\ \psi \\ r \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix}. \end{aligned} \tag{23}$$

The bounds on uncertainties are

$$\left[\left|\tilde{A}_{22}\right| \quad \left|\tilde{A}_{24}\right| \quad \left|\tilde{A}_{42}\right| \quad \left|\tilde{A}_{44}\right|\right]^\mathsf{T} \le \Omega_{\mathbf{M}}(\pi), \quad |e| \le \Omega_{\mathbf{A}}(\pi). \tag{24}$$

The overall π parameterization appears as

$$\begin{aligned} \hat{\mathbf{A}}(\pi) &= A_0 + \sum_{i=1}^{n_1} \pi_1^i \bar{A}_i, \quad \Omega_{\mathbf{M}}(\pi) = \left[\pi_2^1, \pi_2^2, \pi_2^3, \pi_2^4\right]^\mathsf{T} \\ \Omega_{\mathbf{A}}(\pi) &= \left[\pi_3^1, \pi_3^2, \pi_3^3, \pi_3^4\right]^\mathsf{T}, \quad \pi = \left[\pi_1^\mathsf{T}, \pi_2^\mathsf{T}, \pi_3^\mathsf{T}\right]^\mathsf{T}. \end{aligned} \tag{25}$$

The parameterization is indeed affine.

2) Experimental Setup: We conducted experiments on a hardware platform built based on a Lincoln MKZ sedan. Multiple sensors, such as Lidars, radars, cameras, and GPS units, are installed on the car, and the vehicle can be controlled by wire, using commands including its throttle, braking, transmission shift, and steering. We used a high-precision real-time kinematic (RTK) GPS as the main sensor in our experiments; the model onboard is OXTS RT3003. The GPS gives measurements of the position, heading angle, and velocity. Together with a map of the desired path, the states for lane keeping can be calculated.

Among the measurements, the position and heading angle measurements are extremely accurate; the measurements of v_y and r, however, are noisy. To get a better estimation of the states, a Kalman filter is used to filter out the noise. The design of the Kalman filter follows the standard procedure with a crude model of the vehicle dynamics (the Kalman filter is robust against inaccurate models) and is omitted here. To this end, the model identified from the measurement and used for mRCI computation actually describes the generalized dynamical system shown in Fig. 4. Since the RTK GPS uses the same kinetic relationship to obtain velocity measurements, the model structure in (22) remains unchanged for the generalized model. Note that, in our implementation of the controllers, the same Kalman filter is used to generate the state estimation, so, indeed, the controller is dealing with the generalized dynamical system shown in Fig. 4.

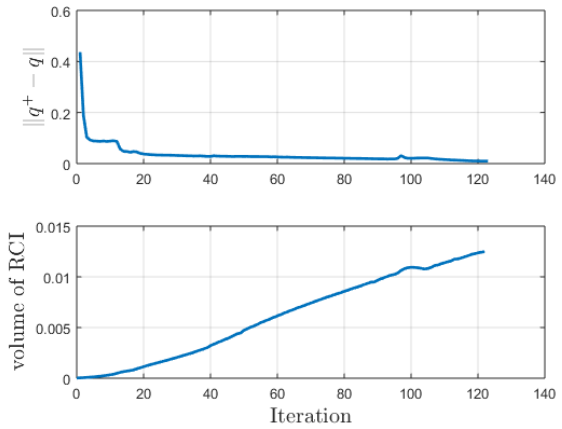

Fig. 5. Convergence of the inside-out algorithm.

3) Experimental Results: The first step is to collect data. We let the vehicle follow a sinusoidal-shaped path with the default lane-keeping controller equipped by the OpenAV platform, which is a preview lane-keeping controller, as introduced in [40]. The path was recorded in advance by a human driver, which was in the form of trajectory points consisting of X, Y coordinates, heading angle, and curvature. During the data collection process, the GPS signal is compared to the recorded path to obtain the lane-keeping states in (20), which are then filtered by the Kalman filter. Then, the output of the Kalman filter was used to formulate the set of admissible models Σ following the procedure in (9). With the set of admissible models ready, the iterative algorithm was used to compute an mRCI. Since no existing RCI was available, the inside-out algorithm was used. The specification include keeping the vehicle within the lane boundaries ($|y| \leq 0.9$ m) while driving at $v_x = 5$ m/s with maximum road curvature 0.025 rad/m. To make sure that the steer-by-wire actuator was able to keep up with the steering command, the following bound was enforced on the control gains in the mRCI computation:

$$\left|K_{\text{fb}}^1\right| \leq 0.3, \quad \left|K_{\text{fb}}^3\right| \leq 1.2. \tag{26}$$

The inside-out algorithm converged after 123 iterations, as shown in Fig. 5, returning an admissible model, an mRCI [denoted by $\mathcal{P}(P^{\text{LK}}, q^{\text{LK}})$], and a set of control gains depending on the states x and the measurable disturbance r_d

$$u = [-0.30, -0.061, -0.85, -0.05]x + 0.03r_d. \tag{27}$$

This controller was used onboard as the lane-keeping controller for validating the computed mRCI.

Then, the computed mRCI was validated with experiments. A sinusoidal-shaped path was used, with a 1.5-m amplitude, a wavelength at 49 m, and a maximum road curvature of 0.025 rad/m, which is assumed to be the maximum allowed road curvature. Fig. 6 shows the plots of $P_i^{\text{LK}} x / q_i^{\text{LK}}$ for different hyperplanes indexed by i. The initial condition was outside the invariant set as it was not possible to arbitrarily initialize the real vehicle and the vehicle was not aligned with the prerecorded path. Though, after the initial convergence, all trajectories are below 1, indicating that the state is, indeed, contained in the mRCI. A video of the experimental validation

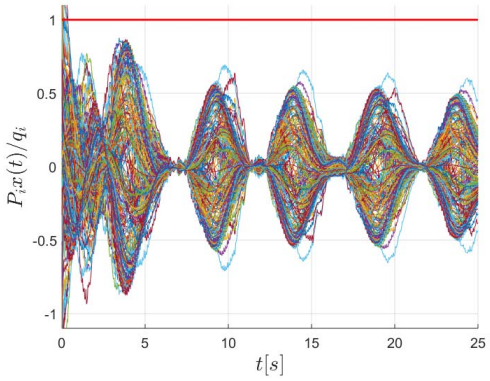

Fig. 6. Relative position with respect to the different hyperplanes i in the computed mRCI $\mathcal{P}(P^{\text{LK}}, q^{\text{LK}})$.

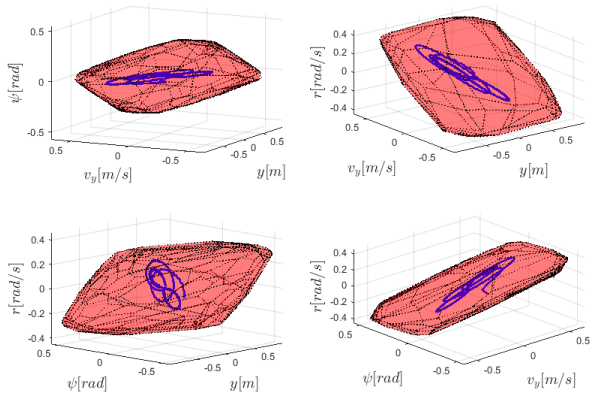

Fig. 7. Projection of the 4-D mRCI on 3-D space for visualization.

can be found at https://youtu.be/gl4rLJm3RcQ. We also note that the experiments were run on a rainy day as can be seen from the video.

A more intuitive plot is shown in Fig. 7. The pink polytopes are four projections of the mRCI to 3-D space, and the blue curves are the projections of the state during the experiment. Indeed, the state trajectory is contained in the mRCI.

Remark 6: As a benchmark, the mRCI algorithm was tested with the undetermined entries of (22) fixed to be the linear regression model identified from the experiment data. We did a similar comparison in [33], where the data came from the Carsim simulation, and the regression model resulted in a much larger RCI than the optimal model. When real experimental data were used, the uncertainties were stronger, and the mRCI algorithm failed to obtain an RCI with the regression model. This indicates that, although the linear regression model minimized model uncertainty, it is not the optimal model for mRCI computation. One of the strengths of the proposed algorithm is that the model and the mRCI are optimized simultaneously, resulting in a better result than a preselected nominal model.

Remark 7: Experiments were conducted on sinusoidal paths with sharper (40-m wavelength) and milder (80-m wavelength) turns, and the state was contained in the mRCI in both cases. However, when the longitudinal speed v_x of the vehicle was changed, the performance deteriorated significantly, indicating that the proposed method is quite sensitive

to v_x. Since v_x is an important parameter in the nominal model, the model may not be admissible once v_x is changed. One possible solution is to parameterize the model with v_x, but some modification will then be needed in the one-step propagation, as the model will no longer be linear.

VI. Conclusion

This article presents a data-driven algorithm to approximately compute a minimal robust invariant set by simultaneously selecting an admissible model and minimizing the size of the RCI. The algorithm has two steps: first, the set of all admissible models with uncertainty characterization is identified from the measurement data; then, a robust linear program is formulated to iteratively search for an mRCI. The robust LP-based algorithm is able to simultaneously select an appropriate model, finding a good tradeoff between the nominal model and different types of uncertainties, and minimize the size of an mRCI. The algorithm is used to construct an mRCI using data from a real vehicle, and the corresponding controller is deployed on the vehicle at Mcity for validation purposes. Our experiments show that the obtained controller successfully keeps the vehicle within a small distance from the lane center, where our algorithm also provides a guarantee on the worst case tracking error bounds in the form of an invariant set.

There are several interesting directions for future research in simultaneously learning the system models and enforcing safety. For instance, the proposed algorithm depends on a finite sequence of data collected from the system; therefore, in theory, it does not provide any invariance guarantees against unseen data if the additional data invalidate the admissible model set. Analyzing the reliability of the computed mRCI in a stochastic setting will be considered in our future work.

Appendix A
Proof of Lemma 1

Proof: The robust optimization problem in (11) is equivalent to the following:

$$\begin{aligned}&\min\ J(\alpha)\\ &\text{s.t.}\ \left\{\max_{\beta\in\mathcal{P}(F,f)} H_1^i\beta+\alpha^\intercal H_2^i\beta+H_3^i\alpha\right\}\le h^i,\quad i=1,\dots,M\end{aligned}\tag{28}$$

where the maximum is taken over the uncertainty set. Consider now one of the inner primal problems

$$\mathbf{p}^i=\max_{\beta\in\mathcal{P}(F,f)} H_1^i\beta+\alpha^\intercal H_2^i\beta+H_3^i\alpha$$

and its dual

$$\mathbf{d}^i=\min_{\lambda^i\ge 0}\max_{\beta} H_1^i\beta+\alpha^\intercal H_2^i\beta+H_3^i\alpha+\left(\lambda^i\right)^\intercal(f-F\beta).$$

If both problems are feasible, then, by strong duality, we have $\mathbf{d}^i=\mathbf{p}^i$. Note that

$$\max_{\beta} H_1^i\beta+\alpha^\intercal H_2^i\beta+H_3^i\alpha+\left(\lambda^i\right)^\intercal(f-F\beta)=\begin{cases}H_3^i\alpha+\left(\lambda^i\right)^\intercal f, & \text{if } H_1^i+\alpha^\intercal H_2^i=\left(\lambda^i\right)^\intercal F\\ \infty, & \text{otherwise.}\end{cases}\tag{29}$$

Thus, the dual is written as

$$\mathbf{d}^i=\min_{\lambda^i\ge 0,\,H_1^i+\alpha^\intercal H_2^i=\left(\lambda^i\right)^\intercal F} H_3^i\alpha+\left(\lambda^i\right)^\intercal f.\tag{30}$$

Plugging this expression into (28), the original robust optimization problem (11) is transformed to the following:

$$\begin{aligned}&\min\ J(\alpha)\\ &\text{s.t. } H_3^i\alpha+\left(\lambda^i\right)^\intercal f\le h^i\\ &\quad H_1^i+\alpha^\intercal H_2^i=\left(\lambda^i\right)^\intercal F\\ &\quad \lambda^i\ge 0,\quad i=1,\dots,M.\end{aligned}\tag{31}$$

Note that, due to the duality, $\mathbf{d}^i$ is always an upper bound of the right-hand side of the constraint in (11); therefore, a solution to (31) is always a feasible solution to (11), and when strong duality holds, the two formulations are equivalent. ■

Appendix B
Proof of Theorem 1

Before proving Theorem 1, we present two lemmas related to convex compact polytopes.

Lemma 2: For a nonempty compact polytope $\mathcal{P}(P,q)$, suppose that one moves the ith hyperplane from q_i to $\bar{q}_i$ and leaves the rest unchanged, resulting in the following polytope $\mathcal{P}(P,q')$, where $q'=[q_1,\dots,\bar{q}_i,q_{i+1},\dots,q_L]^\intercal$. Then, there exists a constant c_i such that, if $\bar{q}_i<c_i$, $\mathcal{P}(P,q')=\emptyset$.

Proof: Since $\mathcal{P}(P,q)$ is compact, $f(x)=P_ix$ is a continuous function, and it always achieves its minimum value on a compact set. Let $c_i=\min_{x\in\mathcal{P}(P,q)} P_ix$. It is obvious that $c_i\le q_i$. Note that, if $\bar{q}_i\le q_i$, $\mathcal{P}(P,q')=\mathcal{P}(P,q)\cap\{x\mid P_ix\le\bar{q}_i\}$. Therefore, when $\bar{q}_i<c_i$, $\mathcal{P}(P,q')=\emptyset$. ■

Lemma 3: Let $\mathcal{P}(P,q)$ be a nonempty compact polytope. Define $c_i=\min_{x\in\mathcal{P}(P,q)} P_ix$. For any $q'\le q$ and $c=[c_1,c_2,\dots,c_L]^\intercal$, if the set $\mathcal{P}(P,q')$ is nonempty, then $q'\ge c$.

Proof: The proof follows from **Lemma** 2 and the fact that $c_i'=\min_{x\in\mathcal{P}(P,q')} P_ix\ge c_i$. ■

Now, we can prove **Theorem** 1.

Proof of Theorem 1: First, we show that the one-step propagation is always feasible, and every q^+ during the iteration leads to an RCI. For clarity, denote the offset q found in the ith iteration as q^i. We show this by induction. For the first iteration, by assumption, q^0 is a feasible solution to the one-step propagation. Assuming the nth iteration is feasible, and the optimal solution for q^+ is q^n, then

$$\begin{aligned}&\exists K_{\mathrm{ff}},K_{\mathrm{fb}},\pi,q^+\quad\text{s.t. }\pi\in\Sigma\\ &\quad\forall x\in\mathcal{P}\left(P,q^{n-1}\right)\quad\forall d\in\mathcal{D}\ \forall|\tilde{A}|\le\Omega_{\tilde{\mathbf{A}}}(\pi)\\ &\quad\forall|\tilde{E}|\le\Omega_{\tilde{\mathbf{E}}}(\pi)\quad\forall|e|\le\Omega_{\mathbf{A}}(\pi)\\ &\quad P\begin{pmatrix}\hat{\mathbf{A}}(\pi)x+\hat{B}\left(K_{\mathrm{ff}}^\intercal d+K_{\mathrm{fb}}^\intercal x\right)\\ +\hat{\mathbf{E}}(\pi)d+\tilde{E}d+\tilde{A}x+e\end{pmatrix}\le q^+\\ &\quad K_{\mathrm{ff}}^\intercal d+K_{\mathrm{fb}}^\intercal x\in\mathcal{U},\quad q^+\le q^n.\end{aligned}\tag{32}$$

Note that we replace the last constraint $q^+\le q^{n-1}$ with $q^+\le q^n$ since q^n is a feasible solution. Consider the one-step propagation in the $n+1$th iteration. The only difference between the robust LP solved in the $n+1$th iteration and the constraints in (32) is that the uncertainty set of x changes

from $\mathcal{P}(P, q^{n-1})$ to $\mathcal{P}(P, q^n)$. Since $\mathcal{P}(P, q^n) \subseteq \mathcal{P}(P, q^{n-1})$, the uncertainty set for the $n+1$th iteration is a subset of that in the nth iteration; therefore, q^n is still a solution to the one-step propagation in the $n+1$th iteration, and the one-step propagation is still feasible. By induction, the one-step propagation in the outside-in algorithm is always feasible, and for all q^n, $\mathcal{P}(P, q^n)$ is always an RCI.

Next, let $c_i = \min_{x \in \mathcal{P}(P, q^0)} P_i x$. Since, for all $n \in \mathbb{Z}_{\geq 0}$, $q^n \leq q^0$, by **Assumption** 1, $\mathcal{P}(P, q^0)$ is compact. Then, by **Lemma** 3, for all $n \in \mathbb{Z}_{\geq 0}$, $q^n \geq c$. Note that, in every iteration, $\mathbf{0} \leq q^{n+1} \leq q^n$, $\mathbf{1}^\intercal q^{n+1} < \mathbf{1}^\intercal q^n - \epsilon$. Then, we have $\mathbf{1}^\intercal q^n < \mathbf{1}^\intercal q^0 - n\epsilon$, and $\mathbf{1}^\intercal q^n$ is lower bounded by $\mathbb{1}^\intercal c$; therefore, the iteration will terminate in finite time.

ACKNOWLEDGMENT

The authors gratefully acknowledge Prof. Huei Peng and Prof. Jessy Grizzle for their help during the early phases of this project. They also thank Dr. Shaobing Xu who helped them conduct the experiments at Mcity.

REFERENCES

[1] P. Nilsson *et al.*, "Correct-by-construction adaptive cruise control: Two approaches," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 4, pp. 1294–1307, Jul. 2016.

[2] M. Nagumo, "Über die lage der integralkurven gewöhnlicher differentialgleichungen," *Proc. Physico-Math. Soc. Jpn. 3rd Ser.*, vol. 24, pp. 551–559, 1942.

[3] J. P. Aubin, "A survey of viability theory," *SIAM J. Control Optim.*, vol. 28, no. 4, pp. 749–788, May 1990.

[4] D. Bertsekas, "Infinite time reachability of state-space regions by using feedback control," *IEEE Trans. Autom. Control*, vol. 17, no. 5, pp. 604–613, Oct. 1972.

[5] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA, USA: SIAM, 1994.

[6] M. V. Khlebnikov, B. T. Polyak, and V. M. Kuntsevich, "Optimization of linear systems subject to bounded exogenous disturbances: The invariant ellipsoid technique," *Autom. Remote Control*, vol. 72, no. 11, pp. 2227–2275, Nov. 2011.

[7] S. A. Nazin, B. T. Polyak, and M. V. Topunov, "Rejection of bounded exogenous disturbances by the method of invariant ellipsoids," *Autom. Remote Control*, vol. 68, no. 3, pp. 467–486, Mar. 2007.

[8] A. Papachristodoulou and S. Prajna, "A tutorial on sum of squares techniques for systems analysis," in *Proc. Amer. Control Conf.*, Jun. 2005, pp. 2686–2700.

[9] W. Tan and A. Packard, "Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming," *IEEE Trans. Autom. Control*, vol. 53, no. 2, pp. 565–571, Mar. 2008.

[10] Y. Chen, H. Peng, and J. W. Grizzle, "Validating noncooperative control designs through a Lyapunov approach," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 2, pp. 527–539, Mar. 2019.

[11] I. Kolmanovsky and E. G. Gilbert, "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Math. Problems Eng.*, vol. 4, no. 4, pp. 317–367, 1998.

[12] D. Q. Mayne, M. M. Seron, and S. V. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, Feb. 2005.

[13] S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne, "Invariant approximations of the minimal robust positively invariant set," *IEEE Trans. Autom. Control*, vol. 50, no. 3, pp. 406–410, Mar. 2005.

[14] S. V. Rakovic and M. Baric, "Parameterized robust control invariant sets for linear systems: Theoretical advances and computational remarks," *IEEE Trans. Autom. Control*, vol. 55, no. 7, pp. 1599–1614, Jul. 2010.

[15] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, Nov. 1999.

[16] P. Trodden, "A one-step approach to computing a polytopic robust positively invariant set," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 4100–4105, Dec. 2016.

[17] T. B. Blanco, M. Cannon, and B. D. Moor, "On efficient computation of low-complexity controlled invariant sets for uncertain linear systems," *Int. J. Control*, vol. 83, no. 7, pp. 1339–1346, Jul. 2010.

[18] F. Tahir and I. M. Jaimoukha, "Low-complexity polytopic invariant sets for linear systems subject to norm-bounded uncertainty," *IEEE Trans. Autom. Control*, vol. 60, no. 5, pp. 1416–1421, May 2015.

[19] M. Rungger and P. Tabuada, "Computing robust controlled invariant sets of linear systems," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3665–3670, Jul. 2017.

[20] S. W. Smith, P. Nilsson, and N. Ozay, "Interdependence quantification for compositional control synthesis with an application in vehicle safety systems," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 5700–5707.

[21] Y. Chen, J. Anderson, K. Kalsi, S. H. Low, and A. D. Ames, "Compositional set invariance in network systems with assume-guarantee contracts," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 1027–1034.

[22] L. Ljung, "System identification," in *Wiley Encyclopedia of Electrical and Electronics Engineering*. 1999, pp. 1–19.

[23] J. Chen and G. Gu, *Control-Oriented System Identification: An H [Infinity] Approach*, vol. 19. Hoboken, NJ, USA: Wiley-Interscience, 2000.

[24] D. S. Shook, C. Mohtadi, and S. L. Shah, "A control-relevant identification strategy for GPC," *IEEE Trans. Autom. Control*, vol. 37, no. 7, pp. 975–980, Jul. 1992.

[25] G. C. Goodwin and M. E. Salgado, "A stochastic embedding approach for quantifying uncertainty in the estimation of restricted complexity models," *Int. J. Adapt. Control Signal Process.*, vol. 3, no. 4, pp. 333–356, Dec. 1989.

[26] R. L. Kosut, M. K. Lau, and S. P. Boyd, "Set-membership identification of systems with parametric and nonparametric uncertainty," *IEEE Trans. Autom. Control*, vol. 37, no. 7, pp. 929–941, Jul. 1992.

[27] V. F. Sokolov, "Closed-loop identification for the best asymptotic performance of adaptive robust control," *Automatica*, vol. 32, no. 8, pp. 1163–1176, Aug. 1996.

[28] V. F. Sokolov, "Model evaluation for robust tracking under unknown upper bounds on perturbations and measurement noise," *IEEE Trans. Autom. Control*, vol. 59, no. 2, pp. 483–488, Feb. 2014.

[29] V. F. Sokolov, "Modeling the system of suboptimal robust tracking under unknown upper bounds on the uncertainties and external disturbances," *Autom. Remote Control*, vol. 75, no. 5, pp. 900–916, May 2014.

[30] S. Sadraddini and C. Belta, "Formal guarantees in data-driven model identification and control synthesis," in *Proc. 21st Int. Conf. Hybrid Syst., Comput. Control (Part CPS Week)*, Apr. 2018, pp. 147–156.

[31] J. Kenanian, A. Balkan, R. M. Jungers, and P. Tabuada, "Data driven stability analysis of black-box switched linear systems," *Automatica*, vol. 109, Nov. 2019, Art. no. 108533.

[32] H. Ravanbakhsh and S. Sankaranarayanan, "Learning Lyapunov (potential) functions from counterexamples and demonstrations," 2017, *arXiv:1705.09619*. [Online]. Available: http://arxiv.org/abs/1705.09619

[33] Y. Chen, H. Peng, J. Grizzle, and N. Ozay, "Data-driven computation of minimal robust control invariant set," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 4052–4058.

[34] A. Ben-Tal, D. D. Hertog, and J.-P. Vial, "Deriving robust counterparts of nonlinear uncertain inequalities," *Math. Program.*, vol. 149, nos. 1–2, pp. 265–299, Feb. 2015.

[35] S. T. S. Wong and M. S. Roos, "A strategy for sampling on a sphere applied to 3D selective RF pulse design," *Magn. Reson. Med.*, vol. 32, no. 6, pp. 778–784, Dec. 1994.

[36] J. J. Kuffner, "Effective sampling and distance metrics for 3D rigid body path planning," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Apr./May 2004, pp. 3993–3998.

[37] U. of Michigan. *U-M Offers Open-Access Automated Cars to Advance Driverless Research*. [Online]. Available: https://news.umich.edu/u-m-offers-open-access-automated-cars-to-advance-driverless-research/

[38] B. A. H. Vicente, S. S. James, and S. R. Anderson, "Linear system identification versus physical modeling of lateral-longitudinal vehicle dynamics," *IEEE Trans. Control Syst. Technol.*, early access, May 22, 2020, doi: 10.1109/TCST.2020.2994120.

[39] S. Maitra and J. Yan, "Principle component analysis and partial least squares: Two dimension reduction techniques for regression," *Applying Multivariate Stat. Models*, vol. 79, pp. 79–90, 2008.

[40] H. Peng and M. Tomizuka, "Lateral control of front-wheel-steering rubber-tire vehicles," Program Adv. Technol. Highway, Univ. California Berkeley, Berkeley, CA, USA, Tech. Rep. UCB-ITS-PRR-90-5, 1990.

Yuxiao Chen (Member, IEEE) received the B.S. degree in mechanical engineering from Tsinghua University, Beijing, China, in 2013 and the Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2018.

He is currently a Postdoc Researcher with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA, USA. His research interests include control and robotics, especially on safety-critical control synthesis and multiagent systems.

Necmiye Ozay (Senior Member, IEEE) received the B.S. degree from Bogazici University, Istanbul, Turkey, in 2004, the M.S. degree from the Pennsylvania State University, University Park, State College, PA, USA, in 2006, and the Ph.D. degree from Northeastern University, Boston, MA, USA, in 2010, all in electrical engineering.

She was a Post-Doctoral Scholar with the California Institute of Technology, Pasadena, CA, USA, between 2010 and 2013. She joined the University of Michigan, Ann Arbor, MI, USA, in 2013, where she is currently an Associate Professor of electrical engineering and computer science. She is also a core member of Michigan Robotics. Her research interests include hybrid dynamical systems, control, optimization and formal methods with applications in cyber–physical systems, system identification, verification and validation, autonomy, and dynamic data analysis.

Dr. Ozay has received the 1938E Award and a Henry Russel Award from the University of Michigan for her contributions to teaching and research, and five young investigator awards, including NSF CAREER.