2020 59th IEEE Conference on Decision and Control (CDC)
Jeju Island, Republic of Korea, December 14-18, 2020

Safe Learning-based Model Predictive Control under State- and Input-dependent Uncertainty using Scenario Trees

Angelo D. Bonzanini†, Joel A. Paulson†, and Ali Mesbah

Abstract— **The complex and uncertain dynamics of emerging systems pose several unique challenges that need to be overcome in order to design high-performance controllers. A key challenge is that safety is often achieved at the expense of closed-loop performance. This is particularly important when the uncertainty description is provided in the form of a bounded set that is estimated offline from limited data. Replacing this bounded set with a learned state- and input-dependent uncertainty enables representing the the variation of uncertainty in the model throughout the state space, thus improving closed-loop performance. Gaussian process (GP) models are a good candidate for learning such a representation; however, they produce a nonlinear and nonconvex description of the uncertainty set that is difficult to incorporate into currently available robust model predictive control (MPC) frameworks. In this work, we present a learning- and scenario-based MPC (L-sMPC) strategy that systematically accounts for feedback in the prediction using a state- and input-dependent scenario tree computed from a GP uncertainty model. To ensure that the closed-loop system evolution remains safe, we also propose a projection-based safety certification scheme that ensures the control inputs keep the system within an appropriately defined invariant set. The advantages of the proposed L-sMPC method in terms of improved performance and an enlarged feasible region are illustrated on a benchmark double integrator problem.**

I. Introduction

Learning-based control methods have been shown to be particularly effective for control of complex systems operating in uncertain and hard-to-model environments, especially when purely model-based control methods have limited effectiveness [1]. Learning-based model predictive control (L-MPC) is one of the most popular approaches when the underlying system is constrained and has multiple inputs and outputs [2], [3]. While learning the system uncertainty can improve control performance, the statistical nature of learning-based approaches introduces important challenges in guaranteeing robust constraint satisfaction. This is particularly important in L-MPC of safety-critical systems with hard-to-model dynamics, such as biomedical systems [4] and autonomous driving applications [5].

The notion of L-MPC with robustness guarantees was first introduced in [6], where two models of the system are used in order to decouple safety and performance: (i) an approximate model with bounded uncertainty, which is used offline to establish the stability and robustness properties via reachability analysis tools; and (ii) a model that is continuously updated online by using data-driven approaches and thus is employed for performance optimization. However, when the uncertainty is represented in the form of worst-case bounds, which are often conservatively estimated offline from limited data, the desired safety objective is typically achieved at the expense of closed-loop performance. Thus, several recent works have focused on replacing the bounded uncertainty with a state- and input-dependent uncertainty that can account for the fact that the degree of uncertainty in the model (or environment) may differ throughout the state space [4], [7], [8]. Although Gaussian process (GP) regression has proven to be an effective data-driven modeling approach for learning state- and input-dependent uncertainty representations [7], [9], [10], it naturally yields a nonlinear description of the uncertainty. This nonlinearity can produce highly nonconvex uncertainty sets that are difficult to incorporate into available L-MPC methods that can provide a theoretical guarantee of robust constraint satisfaction. This is because such L-MPC methods generally rely on offline tightening of the constraints in terms of a worst-case description of the uncertainty that can significantly reduce the feasible region of the controller.

In this work, we present a learning- and scenario-based model predictive control (L-sMPC) strategy that learns a GP representation of the unmodeled system dynamics while systematically accounting for feedback in the predictions by using a scenario tree structure. Scenario-based MPC enables the inclusion of recourse in the optimization, thus reducing the conservativeness that arises when optimizing over open-loop control actions as opposed to more general feedback policies [11], [12], [13]. Here, instead of bounding the uncertainty offline and constructing a scenario tree based on an unrealistically conservative worst-case scenario, we use GP regression to obtain a state- and input-dependent uncertainty model that allows the scenario tree to be adapted online. While online adaptation of the scenario tree can enhance closed-loop performance, the resulting controller does not guarantee constraint satisfaction by design as only a finite number of scenarios are considered. Thus, we leverage the notion of robust control invariant (RCI) sets [14], [15], [16], [17] to establish an online safety certificate for the L-sMPC strategy. To this end, we present an algorithm for constructing (maximal) RCI sets given the GP description of the state- and input-dependent uncertainty. The advantages

This work was supported by the National Science Foundation under Grant 1839527.

A. D. Bonzanini and A. Mesbah are with the Department of Chemical and Biomolecular Engineering at the University of California, Berkeley, CA 94720, USA. {adbonzanini,mesbah}@berkeley.edu

J. A. Paulson is with the Department of Chemical and Biomolecular Engineering at The Ohio State University, Columbus, OH 43210, USA. paulson.82@osu.edu

†A. D Bonzanini and J. A. Paulson contributed equally to this work.

of the proposed safe L-sMPC approach in terms of an enlarged safety region with guaranteed constraint satisfaction and improved performance are demonstrated on a benchmark double integrator problem.

Notation. The set of non-negative integers is denoted by $\mathbb{N}$. The ith element of a vector is denoted by $[x]_i$. Given a matrix M, the ij entry is denoted by $[M]_{i,j}$ and the ith column and row are denoted by $[M]_{\cdot,i}$ and $[M]_{i,\cdot}$, respectively. The weighted Euclidean norm is given by $\|x\|_M^2 := x^\top M^{-1} x$. $\mathbb{E}\{X\}$ denotes the expected value of a random vector X and $X \sim \mathcal{N}(\mu, \Sigma)$ denotes a normally distributed random vector with mean μ and covariance Σ. Given two sets $A \subset \mathbb{R}^n$ and $B \subset \mathbb{R}^n$, the set difference between A and B is $A \setminus B := \{a \in A \mid a \notin B\}$ and the Minkowski set addition of A and B is $A \oplus B := \{a+b \mid a \in A, b \in B\}$. Given a set $S \subset X \times Y$, the orthogonal projection of the set S onto X is defined as $\text{Proj}_X(S) := \{x \in X \mid \exists y \in Y \text{ s.t. } (x,y) \in S\}$.

II. Problem Statement

Consider an uncertain discrete-time system of the form

$$x^+ = f(x,u) + B_d(g(x,u) + w), \tag{1}$$

where $x \in X := \mathbb{R}^{n_x}$ is the current state, x^+ is the state at the next time instant, $u \in U := \mathbb{R}^{n_u}$ is the control input, and $w \in W := \mathbb{R}^{n_d}$ is a process noise that is assumed to be normally distributed, i.e., $w \sim \mathcal{N}(0, \Sigma^w)$. The model is composed of a known nominal part $f : X \times U \to X$ and an additive term $g : X \times U \to W$ that describes initially unknown system dynamics that are to be learned from data and lie in the subspace spanned by the matrix $B_d \in \mathbb{R}^{n_x \times n_d}$. The state and input are required to satisfy the following set of mixed constraints

$$(x,u) \in \mathcal{Y} \subset X \times U. \tag{2}$$

In this work, we choose to model the noisy vector-valued function g (referred to as a state- and input-dependent uncertainty) using GP regression [18]. We use $M \in \mathbb{N}_+$ training points in the form of previously collected measurements of the states and inputs

$$y_j = B_d^\dagger (x_{j+1} - f(x_j, u_j)) = g(x_j, u_j) + w_j, \tag{3}$$

for all $j = 1, \ldots, M$, where $B_d^\dagger$ denotes the Moore-Penrose pseudoinverse of B_d. Letting $z_j = (x_j, u_j)$, the training data set can be defined as

$$\mathcal{D} = \{\mathbf{y} = [y_1, \ldots, y_M]^\top,\ \mathbf{z} = [z_1, \ldots, z_M]^\top\}. \tag{4}$$

To simplify presentation, we assume that each dimension of g is learned separately. By specifying a GP prior on each element $a \in \{1, \ldots, n_d\}$ of g with mean function $m^a(\cdot)$ and kernel $k^a(\cdot,\cdot)$ and conditioning on the training data $\mathcal{D}$, we obtain a Gaussian posterior distribution at any given test point $z = (x,u)$ with mean and covariance

$$\mu_a^d(z) = m^a(z) + K_{z\mathbf{z}}^a (K_{\mathbf{zz}}^a + \sigma_a^2 I)^{-1} ([\mathbf{y}]_{\cdot,a} - m_{\mathbf{z}}^a), \tag{5a}$$
$$\Sigma_a^d(z) = K_{zz}^a - K_{z\mathbf{z}}^a (K_{\mathbf{zz}}^a + \sigma_a^2 I)^{-1} K_{\mathbf{z}z}^a, \tag{5b}$$

where K is the Gram matrix that is composed of the following terms $[K_{\mathbf{zz}}^a]_{i,j} = k^a(z_i, z_j)$, $[K_{\mathbf{z}z}^a]_j = k^a(z_j, z)$, $K_{z\mathbf{z}}^a = (K_{\mathbf{z}z}^a)^\top$, $K_{zz}^a = k^a(z,z)$, and $[m_{\mathbf{z}}^a]_j = m^a(z_j)$. There are many possible choices for the kernel function, one of the most popular being the squared exponential kernel

$$k^a(z_i, z_j) = \sigma_{f,a}^2 \exp\left(-\frac{1}{2}(z_i - z_j)^\top L_a^{-1} (z_i - z_j)\right), \tag{6}$$

where $\sigma_{f,a}^2$ is the signal variance and L_a is a positive diagonal length-scale matrix [18]. The resulting state- and input-dependent GP approximation of the unknown function $g(x,u)$ is then given by

$$d(x,u) \sim \mathcal{N}(\mu^d(x,u), \Sigma^d(x,u)), \tag{7}$$

where the mean $\mu^d(\cdot) = [\mu_1^d(\cdot), \ldots, \mu_{n_d}^d(\cdot)]^\top$ and covariance $\Sigma^d(\cdot) = \text{diag}([\Sigma_1^d(\cdot), \ldots, \Sigma_{n_d}^d(\cdot)]^\top)$ are concatenated from the individual output predictions in (5).

In this paper, we aim to solve a closed-loop MPC problem for system (1) under the GP representation d of the unknown function g. At each time instant, the GP model evaluates to a stochastic distribution that is added to the process noise and then propagated forward through the prediction model. The resulting closed-loop MPC problem is formulated as

$$\min_{\Pi} \quad J(x, \Pi) := \mathbb{E}\left\{\sum_{i=0}^{N-1} \ell(x_i, u_i) + \ell_f(x_N)\right\}, \tag{8a}$$
$$\text{s.t.} \quad x_{i+1} = f(x_i, u_i) + B_d(d(x_i, u_i) + w_i), \tag{8b}$$
$$u_i = \pi(x_i), \tag{8c}$$
$$(x_i, u_i) \in \mathcal{Y}, \tag{8d}$$
$$x_0 = x, \quad \forall i = 0, \ldots, N-1, \tag{8e}$$

where the decision variables are defined by a control policy $\Pi = \{\pi_0(\cdot), \ldots, \pi_{N-1}(\cdot)\}$ that is a sequence of control laws $\pi_i(x)$ over the prediction horizon N. The objective function is composed of a scalar stage cost $\ell(x,u)$ and a terminal cost function $\ell_f(x)$. The expected value operator $\mathbb{E}\{\cdot\}$ is defined with respect to the random vector sequence $\tilde{\mathbf{w}} = \{\tilde{w}_0, \ldots, \tilde{w}_{N-1}\}$, where $\tilde{w}_i := d(x_i, u_i) + w_i$ is the overall source of uncertainty at the predicted time step i.

The closed-loop MPC problem (8) is not directly solvable since one cannot readily optimize over the generic functions defining the feedback policy Π. This is often addressed by restricting the elements of Π to be from a class of parametrized feedback controllers such as $\pi_i(x_i) = K_i x_i + v_i$. Although this can improve tractability of the problem, it is a suboptimal choice that may lead to a large growth in the predicted uncertainty over time [19], [20]. Scenario trees, on the other hand, represent the evolution of the uncertainty in terms of a tree of discrete uncertainty realizations [12], [13]. Since the feedback structure is not restricted to be affine in the scenario tree, this can lead to improved performance relative to other robust control approaches such as tube-based MPC. The standard formulations of scenario-based MPC consider only exogenous sources of uncertainty that are independent of predicted state and input sequences, implying that the scenario tree can be straightforwardly generated offline.

The first contribution of this work is to present a scenario-based MPC strategy that explicitly incorporates the state- and input-dependent uncertainty in the form of the learned GP model (7), such that the scenario tree can be adapted online in a computationally efficient manner. The proposed learning- and scenario-based MPC (L-sMPC) strategy is presented in Section III. Since scenario-based MPC only enforces constraints at a discrete number of uncertainty realizations, the resulting closed-loop system may not satisfy (2) under the full uncertainty description. Therefore, our second contribution is to introduce a framework for providing a real-time safety certificate by projecting the optimal inputs into a "safe" set that can be derived offline using invariant set theory, as discussed in Section IV.

III. Scenario-based Model Predictive Control under Gaussian Process Uncertainty Models

The design of a suitable scenario tree is a tradeoff between coverage of the uncertainty space and computational cost. One systematic approach to scenario generation is to use quadrature rules [21], with the main goal of deriving an accurate approximation of the expectation integral in terms of a relatively small number of samples of the uncertainty. We look to extend this idea to the case of state- and input-dependent uncertainty by rewriting the cost function (8a) in terms of conditional expectations

$$\begin{aligned} J(x,\Pi) = \; & \ell(x_0,u_0) + \mathbb{E}_0\Big\{\ell(x_1,u_1) + \mathbb{E}_1\Big\{\ell(x_2,u_2) \\ & + \cdots + \mathbb{E}_{N-2}\big\{\ell(x_{N-1},u_{N-1}) \\ & + \mathbb{E}_{N-1}\{\ell_f(x_N)\}\big\}\cdots\Big\}\Big\}, \end{aligned} \tag{9}$$

where $\mathbb{E}_i\{\cdot\} = \mathbb{E}_{\tilde{w}_i|\tilde{w}_{i-1},\ldots,\tilde{w}_0}\{\cdot\}$ is the conditional expectation with respect to the uncertainty at time step i given all previous uncertainty values. Since the current state can be inferred from knowledge of the previous uncertainties, we know that $\tilde{w}_i|\tilde{w}_{i-1},\ldots,\tilde{w}_0 = \tilde{w}_i|x_i,u_i$ is Gaussian

$$\tilde{w}_i|x_i,u_i \sim \mathcal{N}(\mu^{\tilde{w}}(x_i,u_i),\Sigma^{\tilde{w}}(x_i,u_i)), \tag{10}$$

where $\mu^{\tilde{w}}(x,u) = \mu^d(x,u)$ and $\Sigma^{\tilde{w}}(x,u) = \Sigma^d(x,u) + \Sigma^w$. Applying quadrature rules with respect to (10) would require the points and their associated weights to be generated online (at every step of the optimization) since they are dependent on the evolution of the state and input throughout the prediction horizon. This procedure can quickly become intractable and thus we instead look to leverage the fact that $\tilde{w}_i|x_i,u_i$ is Gaussian in order to move much of this cost offline. This is done by transforming $\tilde{w}_i|x_i,u_i$ into a standard normal random variable $\xi_i \sim \mathcal{N}(0,I)$, i.e.,

$$\tilde{w}_i|x_i,u_i = \mu^{\tilde{w}}(x_i,u_i) + \left(\Sigma^{\tilde{w}}(x_i,u_i)\right)^{1/2}\xi_i. \tag{11}$$

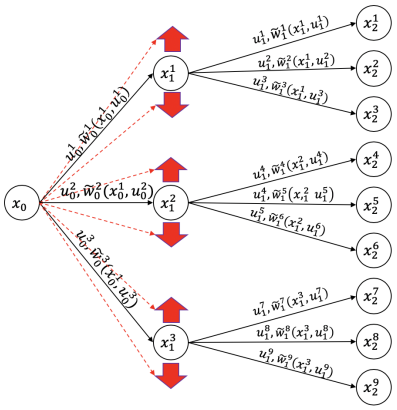

Fig. 1: Scenario tree representation of the state- and input-dependent uncertainty evolution in L-sMPC for an example case in which $s = 3$ and $N = 2$. The scenario tree is adapted at each iteration due to the state- and input-dependent description of uncertainty, which can be visualized as the nodes moving up and down.

We can now approximate the conditional expectation for some arbitrary function $h(x_i,u_i,\tilde{w}_i)$ as follows

$$\begin{aligned} \mathbb{E}_i\{h(x_i,u_i,\tilde{w}_i)\} &= \int h(x_i,u_i,\tilde{w}_i)p(\tilde{w}_i|x_i,u_i)d\tilde{w}_i, \\ &= \int h(x_i,u_i,\tilde{w}_i(\xi_i))p(\xi_i)d\xi_i \approx \sum_{n=1}^{s} p_i^n h(x_i,u_i,\tilde{w}_i(\xi_i^n)), \end{aligned} \tag{12}$$

where s is the number of sample points, $\{\xi_i^n\}_{i=1}^s$ are sample point locations, $\{p_i^n\}_{i=1}^s$ are the weights associated with each sample point, and $\tilde{w}_i(\xi_i^n) = \mu^{\tilde{w}}(x_i,u_i) + (\Sigma^{\tilde{w}}(x_i,u_i))^{1/2}\xi_i^n$. The computational cost of the multivariate integral approximation in (12) is largely dependent on the number of sample points s, which, therefore, should be as small as possible. A wide variety of integration rules over multidimensional spaces have been proposed, the most common being tensor and sparse grid [22], randomized [23], and optimization-based methods such as moment matching [24].

Once the conditional probability distributions are discretized, the state- and input-dependent scenario tree can be constructed, as illustrated in Fig. 1. The root node corresponds to the known initial state $x_0 = x$. The uncertainty at each subsequent layer is approximated with s quadrature points, which grows over time. Each node represents a possible uncertainty and input trajectory, and is assigned a set of state and input variables $\{x_i^n, u_i^n\}$ with $i = 0,\ldots,N$ and $n \in \mathcal{N}(i)$ where $\mathcal{N}(i)$ is the set of nodes at level i in the tree. Then, ξ_i^n is the realization that defines the node state vector from some common parent node

$$\begin{aligned} x_{i+1}^n &= f(x_i^{n'},u_i^{n'}) + B_d\tilde{w}_i^n, \\ \tilde{w}_i^n &= \mu^{\tilde{w}}(x_i^{n'},u_i^{n'}) + \left(\Sigma^{\tilde{w}}(x_i^{n'},u_i^{n'})\right)^{1/2}\xi_i^n, \\ \text{Node}(i,n') &= \text{Parent}(\text{Node}(i+1,n)). \end{aligned} \tag{13}$$

The probability of any given node being visited is equal to

the product of the conditional probabilities (the quadrature weights p_i^n in this case) along the path to that node, which is denoted by P_i^n and should satisfy $\sum_{n\in\mathscr{N}(i)} P_i^n = 1$. The fact that all control inputs that branch from the same parent node are equal in (13) describes the real-time decision problem (i.e., current decisions cannot anticipate the future), which is a direct enforcement of the *non-anticipativity* constraints.

It is important to note that a key difference in (13) compared to traditional scenario-based MPC [12], [13] is that the location of the scenarios can move in response to the predicted state and input values based on the GP model, which is illustrated in Fig. 1. This can significantly reduce conservatism of the controller since, by definition, the magnitude of the state- and input-dependent uncertainties must be less than or equal to the worst-case realization. Given the measured state x, the proposed L-sMPC problem can be formulated as a large-scale optimization problem

$$\begin{aligned}
\min_{U} \quad & J(x,U), && \text{(14a)}\\
\text{s.t.} \quad & x_{i+1}^n = f(x_i^{n'},u_i^{n'}) + B_d\tilde{w}_i^n, && \text{(14b)}\\
& \tilde{w}_i^n = \mu^{\tilde{w}}(x_i^{n'},u_i^{n'}) + \left(\Sigma^{\tilde{w}}(x_i^{n'},u_i^{n'})\right)^{1/2}\xi_i^n, && \text{(14c)}\\
& \text{Node}(i,n') = \text{Parent}(\text{Node}(i+1,n)), && \text{(14d)}\\
& (x_i^n,u_i^n)\in\mathcal{Y}, && \text{(14e)}\\
& x_0^1 = x, \quad \forall i=1,\dots,N,\ \forall n\in\mathscr{N}(i), && \text{(14f)}
\end{aligned}$$

where $U=\{u_i^n \mid i=1,\dots,N,\ n\in\mathscr{N}(i)\}$ is the vector of all decision variables and the objective function is given by

$$\begin{aligned}
J(x,U) = &\sum_{i=0}^{N-1}\sum_{n\in\mathscr{N}(i)} P_i^n\ell(x_i^n,u_i^n) \\
&+ \sum_{n\in\mathscr{N}(N)} P_N^n\ell_f(x_N^n).
\end{aligned} \tag{15}$$

The solution to (14) is denoted by $U^\star(x)$ and the resulting L-sMPC law is defined as the first element of this vector

$$\kappa_{\text{smpc}}(x) := u_0^{1,\star}(x). \tag{16}$$

Remark 1: To avoid the exponential growth of the scenario tree size with respect to prediction horizon N, a usual additional simplifying assumption is to consider branching in the tree only up until a certain stage, often referred to as the robust horizon N_r. The rationale behind this is that branching in the far future is less important than the imminent layers, since the corresponding state trajectories and control variables will be heavily altered by the time that branching stage is reached.

Remark 2: In principle, adaptation of the scenario tree is achieved whether the GP model (7) is trained offline and deployed online, or re-trained at each sampling instant. Re-training the GP model (7) online can enhance controller performance even further due to the additional data collected during operation, at the expense of increased computational cost due to the re-training step.

IV. Non-conservative Real-time Safety Certification in the Presence of State- and Input-dependent Uncertainty

Given that the L-sMPC controller κ_{smpc} does not guarantee satisfaction of the constraints (2), we aim to derive a set of states $\mathcal{S}$ for which a feasible backup control strategy $u_{\mathcal{B}}$ exists such that (2) can be satisfied at all future times. As such, κ_{smpc} can be applied as long as it does not result in the system leaving $\mathcal{S}$; otherwise, the backup controller must be applied to keep the system safe. The notion of a safety controller has been formally defined in a variety of works, e.g., [1], [25], which we look to extend to the case of a GP model (7) of the structural plant-model mismatch. Since both the GP and process noise models have infinite support, constraints cannot be enforced for all possible values of the uncertainty. Instead, we use confidence intervals to characterize the actual expected range of the uncertainties, which can be straightforwardly computed from the GP model

$$\tilde{\mathcal{W}}(x,u) = \left\{ w \mid \|w-\mu^{\tilde{w}}(x,u)\|^2_{\Sigma^{\tilde{w}}(x,u)} \le \chi^2_{n_d}(\alpha)\right\}, \tag{17}$$

where $\chi^2_n(\cdot)$ is the quantile function for the chi-squared distribution with n degrees of freedom and $\alpha\in(0,1)$ is the desired probability level. The choice of α is a tuning parameter, with larger values representing less confidence in the knowledge of function g. We make the following assumption in order to provide theoretical guarantees.

Assumption 1: The state- and input-dependent uncertainty set (17) is found through offline learning and $g(x(k),u(k))+w(k)\in\tilde{\mathcal{W}}(x(k),u(k))$ for all $k\ge 0$.

Under the GP model (7), Assumption 1 does not formally hold; however, it is useful in practice since Gaussian uncertainty models are already an approximation of reality wherein unbounded disturbances do not naturally occur. In other words, this assumption will almost surely hold for α values sufficiently close to one. Combining system constraints (2) with the disturbance constraint (17), we let

$$\Upsilon := \{(x,u,\tilde{w}) \mid (x,u)\in\mathcal{Y} \text{ and } \tilde{w}\in\tilde{\mathcal{W}}(x,u)\} \tag{18}$$

denote the subset of the graph $\tilde{\mathcal{W}}(\cdot)$ where the state and input constraints are satisfied such that $\mathcal{Y}=\text{Proj}_{X\times U}(\Upsilon)$. The state-dependent set of admissible control inputs is then

$$\mathcal{U}(x) := \{u \mid (x,u)\in\mathcal{Y}\}, \tag{19}$$

such that the admissible state set becomes

$$\mathcal{X} := \{x \mid \exists u \text{ s.t. } (x,u)\in\mathcal{Y}\} = \text{Proj}_X(\mathcal{Y}). \tag{20}$$

Given these definitions, we now recall the notion of robust control invariant sets [14], which are known to fulfill the requirements of a safety set [1], [25].

Definition 1: A given set $\mathcal{S}\subseteq\mathcal{X}$ is *robust control invariant* (RCI) for system $x^+ = f(x,u)+B_d\tilde{w}$ and constraints Υ if, for any $x\in\mathcal{X}$, there exists a $u\in\mathcal{U}(x)$ such that $f(x,u)+B_d\tilde{w}\in\mathcal{S}$ for all $\tilde{w}\in\tilde{\mathcal{W}}(x,u)$.

An RCI set $\mathcal{C}_\infty$ is said to be *maximal* in $\mathcal{X}$ if all other RCI sets in $\mathcal{X}$ are contained in $\mathcal{C}_\infty$. By definition, the

maximal RCI set is the largest possible region of the state space in which an admissible control law exists that ensures state constraint satisfaction for all times and all possible disturbance sequences. This implies $\mathcal{S} = \mathcal{C}_\infty$ is ideal, though any RCI set is a valid safety set. For any RCI set $\mathcal{S}$, we define the following set-valued map for each element $x \in \mathcal{S}$

$$\mathcal{S}_u(x) := \{u \in \mathcal{U}(x) \mid f(x,u) \oplus B_d\tilde{\mathcal{W}}(x,u) \subseteq \mathcal{S}\}. \quad (21)$$

The backup controller $u_{\mathcal{B}}$ can make any selection from $\mathcal{S}_u(x)$, i.e., $u_{\mathcal{B}}(x) \in \mathcal{S}_u(x)$ for all $x \in \mathcal{S}$. In particular, we propose to project the L-sMPC solution into the set (21), so that the inputs supplied to the system are as close as possible to L-sMPC while keeping the system safe. This can be formulated in terms of the following optimization

$$u_{\mathcal{S}}(x) = \underset{u \in \mathcal{S}_u(x)}{\operatorname{argmin}} \ \|u - \kappa_{\text{smpc}}(x)\|. \quad (22)$$

Implementing $u_{\mathcal{S}}$ in place of κ_{smpc} provides a direct certificate of safety, which can be formalized into our main result.

Theorem 1: Let Assumption 1 hold. Then, the closed-loop system $x^+ = f(x, u_{\mathcal{S}}(x)) + B_d(g(x, u_{\mathcal{S}}(x)) + w)$ satisfies constraints (2) for all times and all possible disturbances.

Proof: The result follows from [26, Theorem 1] since $\mathcal{S}_u(x) \neq \emptyset$ for any $x \in \mathcal{S}$ and the effective disturbance satisfies $g(x,u) + w \in \tilde{\mathcal{W}}(x,u)$ from Assumption 1. ∎

RCI sets can be calculated recursively from some appropriately specified target set $X_f \subseteq \mathcal{X}$

$$X_{i+1} = \text{Pre}(X_i), \quad X_0 = X_f, \quad (23)$$

where

$$\text{Pre}(\Omega) = \{x \mid \exists u \in \mathcal{U}(x) \text{ such that } f(x,u) + B_d\tilde{w} \in \Omega, \ \forall \tilde{w} \in \mathcal{W}(x,u)\} \quad (24)$$

denotes the *predecessor set* of a given set Ω. Two choices of X_f are particularly relevant in this work. For $X_f = \mathcal{X}$, it can be shown that $X_{i+1} \subseteq X_i$ for all $i \in \mathbb{N}$ and $\mathcal{C}_\infty \subseteq \bigcap_{i\in\mathbb{N}} X_i$. Furthermore, $C_\infty = X_i$ for some $i \in \mathbb{N}$ if and only if $X_{i+1} = X_i$. An alternative is to select X_f to be any (potentially small) RCI set, as it is known that X_i will be RCI for all $i \in \mathbb{N}$ (interested readers are referred to, e.g., [14], [27] for further details). Although the latter method may lead to a smaller safety set $\mathcal{S}$, it has the advantage that the algorithm can be stopped at any iteration.

It is clear that the key result needed to derive the safety set is the computation of the predecessor set (24). This can be done with set algebra as summarized next.

Theorem 2: The set of states that are one-step robustly controllable to Ω is given by

$$\text{Pre}(\Omega) = \text{Proj}_X(\Sigma), \quad (25)$$

where Σ is given by

$$\Sigma = \mathcal{Y} \setminus \text{Proj}_{X\times U}(\Upsilon \setminus \Phi), \quad (26)$$

and $\Phi := f_{\tilde{w}}^{-1}(\Omega) = \{(x,u,\tilde{w}) \mid f(x,u) + B_d\tilde{w} \in \Omega\}$.

Proof: See [28, Theorem 1] for a graphical proof. ∎

The conceptual steps in Theorem 2 are difficult to execute for general nonlinear systems, as the set objects can have arbitrarily complex shapes. In fact, the set Υ will usually be non-convex since $\tilde{\mathcal{W}}(x,u)$ is large in regions lacking training data and small near the training points. We would like to evaluate $\text{Pre}(\cdot)$ using computational geometry, as quantifier elimination methods with polynomial complexity are known to exist. Standard computational geometry tools rely on the efficient manipulation of polytopes, which can be extended to handle non-convex polygon set representations using polyhedral covers (i.e., union of polytopes). We are not aware of similar strategies for working with the non-convex union of ellipsoids, such as (17), so we must first construct a polyhedral cover outer approximation for Υ, i.e.,

$$\Upsilon \subseteq \tilde{\Upsilon} := \bigcup_{i=1}^{N_{\text{pc}}} \{v = (x,u,\tilde{w}) \mid H_i v \leq g_i\}, \quad (27)$$

where N_{pc} is the number of regions used to construct the polyhedral cover. A simple procedure for deriving this outer approximation is shown in Algorithm 1. Line 3 requires a global optimization problem to be solved using, e.g., branch and bound. Note that selecting $N_{\text{pc}} = 1$ results in a worst-case formulation, which corresponds to neglecting the state- and input-dependence of the disturbance.

Algorithm 1 Outer polyhedral constraint approximation

Require: Polyhedral partition for $\mathcal{Y} = \bigcup_{i=1}^{N_{\text{pc}}} \mathcal{Y}_i$
1: **for** $i = 1, 2, \ldots, N_{\text{pc}}$ **do**
2: **for** $j = 1, 2, \ldots, n_d$ **do**
3: Calculate minimum $[\tilde{w}_i^{\min}]_j$ and maximum $[\tilde{w}_i^{\max}]_j$ of the j^{th} element of the disturbance subject to the constraints $(x,u) \in \mathcal{Y}_i$ and $\tilde{w} \in \tilde{\mathcal{W}}(x,u)$
4: **end for**
5: Get Υ_i by combining $\mathcal{Y}_i$ and $\{\tilde{w} \mid \tilde{w}_i^{\min} \leq \tilde{w} \leq \tilde{w}_i^{\max}\}$
6: **end for**

Given polyhedral cover representations of the sets Ω and $\tilde{\Upsilon}$ and nominal dynamics $f(\cdot)$ that are linear or piecewise affine, the computation of inner approximations for $\text{Pre}(\Omega)$ and Σ can be achieved using standard polytope manipulations with, e.g., MPT3 [29]. The required steps can be summarized as

1) Compute the inverse map: $\Phi = f_{\tilde{w}}^{-1}(\Omega)$.
2) Compute the set difference: $\Delta = \tilde{\Upsilon} \setminus \Phi$.
3) Compute the projection: $\Psi = \text{Proj}_{X\times U}(\Delta)$.
4) Compute the set difference: $\Sigma = \mathcal{Y} \setminus \Psi$.
5) Compute the projection: $\text{Proj}_X(\Sigma) \subseteq \text{Pre}(\Omega)$.

An inner RCI approximation $\tilde{\mathcal{S}}$ to the RCI set $\mathcal{S}$, defined by the recursion (23), can be obtained by repeating the above steps with $\Omega \leftarrow \text{Proj}_X(\Sigma)$. The approximation $\tilde{\mathcal{S}}$ converges to $\mathcal{S}$ as $N_{\text{pc}} \to \infty$ since $\tilde{\Upsilon} \to \Upsilon$ according to Algorithm 1. It is important to note that set difference between two polygons is also a polygon, which implies the iterates toward $\tilde{\mathcal{S}}$ will also be polygons. The polyhedral cover representation can be derived from the set difference of all combinations of the sets defining the polyhedral cover of the two polygons (see, e.g., [28, Appendix II]). Computational requirements can be substantially reduced by removing all empty sets

in the polyhedral cover as well as removing any redundant inequalities describing the non-empty polyhedral sets.

Remark 3: Given a polyhedral cover representation of the RCI safety set $\tilde{\mathcal{S}}$, the projection problem (22) can be formulated as a mixed-integer quadratic program (MIQP) using the big-M reformulation and can be solved efficiently using state-of-the-art solvers (e.g., Gurobi or CPLEX).

V. Numerical Example

The proposed L-sMPC strategy is demonstrated on a modified version of the benchmark double integrator problem from [30]. The system dynamics are of the form (1) with

$$f(x,u) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u,$$
$$g(x,u) = 2\left(1 - \cos\left(\tfrac{1.6}{\pi}[x]_1\right)\right),$$

$B_d = [1,0]^\top$ and $\Sigma^w = 0$ (i.e., no measurement error). The constraints (2) are given by

$$\mathcal{Y} = \mathcal{X} \times \mathcal{U} = \{x \mid x \in [-10,10]^2\} \times \{u \mid -5 \le u \le 5\}.$$

The stage cost is given by $\ell(x,u) = \|x - x_{\text{sp}}\|^2 + 0.01\|u - u_{\text{sp}}\|^2$, where x_{sp} and u_{sp} are the desired setpoints for the state and input, respectively, and the terminal cost is $\ell_f(x) = 0$. The quadrature rule (12) is used to approximate the conditional expectations using $s = 5$ scenarios. The GP model is trained using UQLab [31] in Matlab using $M = 5$ training samples that were randomly sampled from a uniform distribution defined over the support $\mathcal{Y}$. All necessary set operations were performed in the MPT3 toolbox [29] and the L-sMPC optimization problem (14) was solved using CasADi [32] with IPOPT [33].

Given the trained GP model, we specify the set $\tilde{\mathcal{W}}(x,u)$ according to (17) with $\alpha = 0.99$, implying a 99% confidence level. Since this set is a nonlinear function of (x,u), we can apply Algorithm 1 to derive a polyhedral cover outer approximation to the uncertainty. The results are shown in Fig. 2 for $N_{\text{pc}} = 20$, where each of the rectangles denotes $\{\tilde{w} \mid \tilde{w}_i^{\min} \le \tilde{w} \le \tilde{w}_i^{\max}\}$ for $i = 1,\dots,N_{\text{pc}}$ projected onto the $[x]_1$ space. We can clearly see that the quality of the outer approximation depends on the number of boxes, so that N_{pc} should be carefully chosen to tradeoff between conservativeness and computational cost. Given $\tilde{\Upsilon}$ from Algorithm 1, a close approximation to the maximal RCI set can be computed via the recursion (23) with $X_f = \mathcal{X}$, which converges in $i = 4$ steps in this problem. We aim to track two setpoints: $x_{\text{sp}} = [10,0]^\top$ for $k = 0,\dots,10$ and $x_{\text{sp}} = [0,0]^\top$ for $k = 11,\dots,20$. The initial state $x(0) = [-10,-4.5]^\top$ is chosen to be at the boundary of the maximal RCI set $\mathcal{S} = \mathcal{C}_\infty$.

The closed-loop state trajectories of the proposed L-sMPC strategy are shown in Fig. 3. For comparison purposes, results are also shown for a scenario-based MPC strategy that uses a fixed scenario tree derived from the worst-case uncertainty bounds. We first note that both control strategies are able to guarantee robust constraint satisfaction, even though the initial desired setpoint is unreachable. This is due to the fact that both controllers use the projection-based safety certification scheme in (22); however, it is clear that the proposed L-sMPC strategy outperforms the worst-case scenario-based MPC, as it is able to push the system closer to the constraint. This is a direct consequence of the fact that L-sMPC accounts for the state- and input-dependence of the uncertainty, such that it is aware that it can more safely operate the system near the constraint (even without exactly knowing the unmodeled dynamics). This, in turn, leads to a larger feasible region, as shown in Fig. 3. Moreover, when the setpoint is shifted to the origin at time $k = 11$, the L-sMPC strategy is able to steer the system very close to the origin, as the uncertainty is mostly concentrated near zero, as opposed to being spread out as in the worst-case scenario-based MPC.

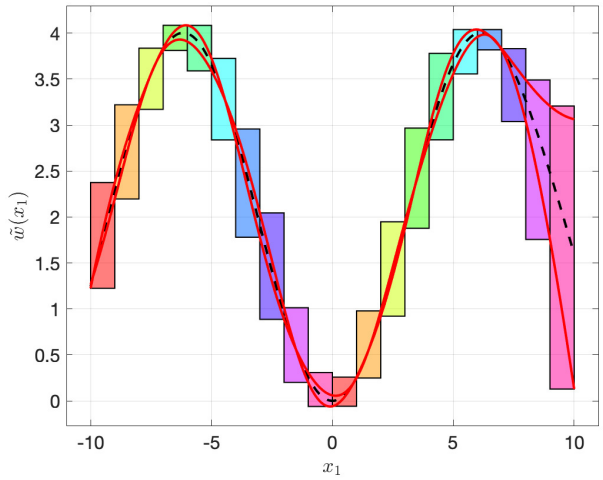

Fig. 2: The learned GP model of the uncertainty with 99% confidence regions shown by red lines. The polytope union is constructed according to Algorithm 1 and projected onto the $[x]_1$ space.

VI. Conclusions

This paper presents a learning- and scenario-based MPC strategy that uses Gaussian process regression to learn a state- and input-dependent system uncertainty, which enables the online adaptation of the scenario tree. While incorporation of feedback in the prediction yields improved control performance, robust constraint satisfaction cannot be guaranteed by design. Thus, we also present a projection-based safety certification scheme that ensures the control inputs keep the system within a robust invariant set designed to account for the state- and input-dependence of the uncertainty. Future work will focus on analyzing the stability properties of the proposed strategy, as well as exploring ways to further reduce the cost of online scenario tree construction via sparse forms of the Gaussian process model predictions. In addition, we will investigate embedded implementation of the proposed strategy for safety-critical systems with fast sampling times.

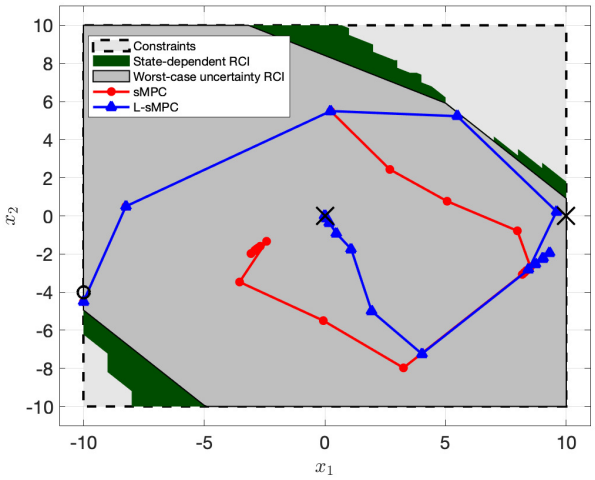

Fig. 3: Closed-loop state trajectories of the L-sMPC strategy based on state-dependent uncertainty (blue) and worst-case scenario-based MPC (sMPC) whereby the scenario tree is fixed a priori by considering the worst-case uncertainty bounds (red). The enlargement of the RCI set computed based on the state-dependent uncertainty description is superimposed. The initial state is marked by "$\circ$", whereas the setpoints are marked by "$\times$".

REFERENCES

[1] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2018.

[2] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 10.1–10.28, 2019.

[3] A. Mesbah, "Stochastic model predictive control with active uncertainty learning: A survey on dual control," *Annual Reviews in Control*, vol. 45, pp. 107–117, 2018.

[4] A. D. Bonzanini, D. B. Graves, and A. Mesbah, "Learning-based stochastic model predictive control for reference tracking under state-dependent uncertainty: An application to cold atmospheric plasmas," *IEEE Transactions on Control Systems Technology*, Under Review, 2020.

[5] S. Lefevre, A. Carvalho, and F. Borrelli, "A learning-based framework for velocity control in autonomous driving," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 32–42, 2015.

[6] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.

[7] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using Gaussian process regression," *IEEE Transactions on Control Systems Technology*, In Press, 2019.

[8] R. Soloperto, M. A. Müller, S. Trimpe, and F. Allgöwer, "Learning-based robust model predictive control with state-dependent uncertainty," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 442–447, 2018.

[9] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Learning-based nonlinear model predictive control to improve vision-based mobile robot path-tracking in challenging outdoor environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014, pp. 4029–4036.

[10] B. Likar and J. Kocijan, "Predictive control of a gas–liquid separation plant based on a Gaussian process model," *Computers & Chemical Engineering*, vol. 31, no. 3, pp. 142–152, 2007.

[11] K. Høyland and S. W. Wallace, "Generating scenario trees for multistage decision problems," *Management Science*, vol. 47, no. 2, pp. 295–307, 2001.

[12] D. Bernardini and A. Bemporad, "Scenario-based model predictive control of stochastic constrained linear systems," in *Proceedings of the IEEE Conference on Decision and Contro*, Shanghai, China, 2009, pp. 6333–6338.

[13] S. Lucia, T. Finkler, and S. Engell, "Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty," *Journal of Process Control*, vol. 23, no. 9, pp. 1306–1319, 2013.

[14] E. C. Kerrigan, "Robust constraint satisfaction: Invariant sets and predictive control," Ph.D. dissertation, University of Cambridge, 2001.

[15] S. Rakovic, P. Grieder, M. Kvasnica, D. Mayne, and M. Morari, "Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances," in *Proceedings of the IEEE Conference on Decision and Control*, Atlantis, Bahamas, 2004, pp. 1418–1423.

[16] T. Alamo, M. Fiacchini, A. Cepeda, D. Limon, J. Bravo, and E. Camacho, "On the computation of robust control invariant sets for piecewise affine systems," in *Assessment and Future Directions of Nonlinear Model Predictive Control*, 2007, pp. 131–139.

[17] A. D. Bonzanini, J. A. Paulson, D. B. Graves, and A. Mesbah, "Toward safe dose delivery in plasma medicine using projected neural network-based fast approximate NMPC," *IFAC Proceedings Volumes*, 2020.

[18] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, MA: MIT Press, 2006.

[19] L. Chisci, J. A. Rossiter, and G. Zappa, "Systems with persistent disturbances: Predictive control with restricted constraints," *Automatica*, vol. 37, pp. 1019–1028, 2001.

[20] W. Langson, I. Chryssochoos, S. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, pp. 125–133, 2004.

[21] C. Leidereiter, A. Potschka, and H. G. Bock, "Quadrature-based scenario tree generation for nonlinear model predictive control," *IFAC Proceedings Volumes*, vol. 47, pp. 11 087–11 092, 2014.

[22] T. Gerstner and M. Griebel, "Dimension–adaptive tensor–product quadrature," *Computing*, vol. 71, no. 1, pp. 65–87, 2003.

[23] S. Hosder, R. Walters, and M. Balch, "Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables," in *Proceedings of AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, 2012, pp. 1939–1955.

[24] E. K. Ryu and S. P. Boyd, "Extensions of gauss quadrature via linear programming," *Foundations of Computational Mathematics*, vol. 15, no. 4, pp. 953–971, 2015.

[25] A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin, "Reachability-based safe learning with Gaussian processes," in *Proceedings of the IEEE Conference on Decision and Control*, Los Angeles, 2014, pp. 1424–1431.

[26] J. A. Paulson and A. Mesbah, "Approximate closed-loop robust model predictive control with guaranteed stability and constraint satisfaction," *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 719–724, 2020.

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

[28] S. V. Rakovic, E. C. Kerrigan, D. Q. Mayne, and J. Lygeros, "Reachability analysis of discrete-time systems with disturbances," *IEEE Transactions on Automatic Control*, vol. 51, no. 4, pp. 546–561, 2006.

[29] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proceedings of the European Control Conference*, Zürich, 2013, pp. 502–510.

[30] D. Q. Mayne, S. Raković, R. Findeisen, and F. Allgöwer, "Robust output feedback model predictive control of constrained linear systems," *Automatica*, vol. 42, no. 7, pp. 1217–1222, 2006.

[31] S. Marelli and B. Sudret, "UQLab: A framework for uncertainty quantification in MATLAB," in *Proceedings of the International Conference on Vulnerability, Risk Analysis and Management*, Liverpool, 2014.

[32] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[33] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.