Learning-based Model Predictive Control for Safe Exploration

Torsten Koller, Felix Berkenkamp, Matteo Turchetta and Andreas Krause

***Abstract*— Learning-based methods have been successful in solving complex control tasks without significant prior knowledge about the system. However, these methods typically do not provide any safety guarantees, which prevents their use in safety-critical, real-world applications. In this paper, we present a learning-based model predictive control scheme that can provide provable high-probability safety guarantees. To this end, we exploit regularity assumptions on the dynamics in terms of a Gaussian process prior to construct provably accurate confidence intervals on predicted trajectories. Unlike previous approaches, we do not assume that model uncertainties are independent. Based on these predictions, we guarantee that trajectories satisfy safety constraints. Moreover, we use a terminal set constraint to recursively guarantee the existence of safe control actions at every iteration. In our experiments, we show that the resulting algorithm can be used to safely and efficiently explore and learn about dynamic systems.**

I. Introduction

In model-based reinforcement learning (RL, [1]), we aim to learn the dynamics of an unknown system from data, and based on the model, derive a policy that optimizes the long-term behavior of the system. Crucial to the success of such methods is the ability to efficiently explore the state space in order to quickly improve our knowledge about the system. While empirically successful, current approaches often use exploratory actions during learning, which lead to unpredictable and possibly unsafe behavior of the system, e.g., in exploration approaches based on the *optimism in the face of uncertainty* principle [2]. Such approaches are not applicable to real-world safety-critical systems.

In this paper we introduce SafeMPC, a safe model predictive control (MPC) scheme that guarantees the existence of feasible *return trajectories* to a safe region of the state space at every time step with high-probability. These return trajectories are identified through a novel uncertainty propagation method that, in combination with constrained MPC, allows for formal safety guarantees in learning control.

Related Work: One area that has considered safety guarantees is robust MPC. There, we iteratively optimize the performance along finite-length trajectories at each time step, based on a known model that incorporates uncertainties and disturbances acting on the system [3]. In a constrained robust MPC setting, we optimize these local trajectories under additional state and control constraints. Safety is typically defined in terms of recursive feasibility and robust constraint satisfaction. In [4], this definition is used to safely control urban traffic flow, while [5] guarantees safety by switching between a standard and a safety mode. However, these methods are conservative since they do not update the model.

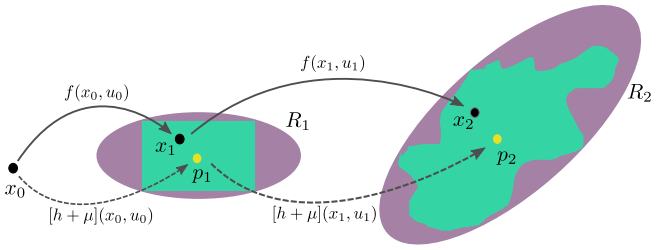

Fig. 1. Propagation of uncertainty over multiple time steps based on a well-calibrated statistical model of the unknown system. We iteratively compute ellipsoidal over-approximations (purple) of the intractable image (green) of the learned model for uncertain ellipsoidal inputs.

In contrast, learning-based control approaches adapt their models online based on observations of the system. This allows the controller to improve over time, given limited prior knowledge of the system. Theoretical safety guarantees in learning-based MPC (LBMPC) are established in [6]. A safety mechanism for general learning-based controllers using robust MPC is proposed in [7]. Both approaches require a known nominal linear model. The former approach requires deviations from the system dynamics to be bounded in an pre-specified polytope, the latter relies on sampling.

MPC based on Gaussian process (GP, [8]) models is proposed in a number of works, e.g. [9], [10]. The difficulty here is that trajectories have complex dependencies on states and unbounded stochastic uncertainties. Safety through probabilistic *chance constraints* is considered in [11]–[13] based on approximate uncertainty propagation. While often being empirically successful, these approaches do not theoretically guarantee safety of the underlying system.

Another area that has considered learning for control is model-based RL. There, we aim to learn global policies based on data-driven modeling techniques, e.g., by explicitly trading-off between finding locally optimal policies (exploitation) and learning the behavior of the system globally (exploration) [1]. This results in data-efficient learning of policies in unknown systems [14]. In contrast to MPC, where we optimize finite-length trajectories, in RL we typically aim to find an infinite horizon optimal policy. Hence, enforcing hard constraints in RL is challenging. Control-theoretic safety properties such as Lyapunov stability or robust constraint satisfaction are only considered in a few

This work was supported by SNSF grant 200020_159557, a fellowship within the FITweltweit program of the German Academic Exchange Service (DAAD), the Vector Institute, an Open Philantropy Project AI fellowship, and the Max Planck ETH Center for Learning Systems.

Torsten Koller is with the Department of Computer Science, University of Freiburg, Germany. Email: kollert@informatik.uni-freiburg.de

Felix Berkenkamp, Matteo Turchetta and Andreas Krause are with the Learning & Adaptive Systems Group, Department of Computer Science, ETH Zurich, Switzerland. Email: {befelix, matteotu, krausea}@inf.ethz.ch

Accepted at *the IEEE Conference on Decision and Control, 2018* .

works [15]. In [16], safety is guaranteed by optimizing parametric policies under stability constraints, while [17] guarantees safety in terms of constraint satisfaction through reachability analysis.

Our Contribution: We combine ideas from robust control and GP-based RL to design a MPC scheme that recursively guarantees the existence of a safety trajectory that satisfies the constraints of the system. In contrast to previous approaches, we use a novel uncertainty propagation technique that can reliably propagate the confidence intervals of a GP-model forward in time. We use results from statistical learning theory to guarantee that these trajectories contain the system with high probability jointly for all time steps. In combination with a constrained MPC approach and a terminal set constraint, we then prove the safety of the system. We apply the algorithm to safely explore the dynamics of an inverted pendulum simulation.

II. Problem Statement

We consider a nonlinear, discrete-time dynamical system

$$x_{t+1} = f(x_t, u_t) = \underbrace{h(x_t, u_t)}_{\text{prior model}} + \underbrace{g(x_t, u_t)}_{\text{unknown error}}, \tag{1}$$

where $x_t \in \mathbb{R}^{n_x}$ is the state and $u_t \in \mathbb{R}^{n_u}$ is the control input to the system at time step $t \in \mathbb{N}$. We assume that we have access to a twice continuously differentiable prior model $h(x_t, u_t)$, which could be based on a first principles physics model. The model error $g(x_t, u_t)$ is *a priori* unknown and we use a statistical model to learn it by collecting observations from the system during operation. In order to provide guarantees, we need reliable estimates of the model-error. In general, this is impossible for arbitrary functions g. We make the following additional regularity assumptions.

We assume that the model-error g is of the form $g(z) = \sum_{l=0}^{\infty} \alpha_i k(z, z_l)$, $\alpha_l \in \mathbb{R}, z = (x, u) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$, a weighted sum of distances between inputs z and representer points $z_l = (x_l, u_l) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$ as defined through a symmetric, positive definite *kernel* k. This class of functions is well-behaved in the sense that they form a reproducing kernel Hilbert space (RKHS, [18]) $\mathcal{H}_k$ equipped with an inner-product $\langle \cdot, \cdot \rangle_k$. The induced norm $||g||_k^2 = \langle g, g \rangle_k$ is a measure of the *complexity* of a function $g \in \mathcal{H}_k$. Consequently, the following assumption can be interpreted as a requirement on the smoothness of the model-error g w.r.t. the kernel k.

Assumption 1 *The unknown function g has bounded norm in the RKHS $\mathcal{H}_k$, induced by the continuously differentiable kernel k, i.e. $||g||_k \leq B_g$.*

In the case of a multi-dimensional output $n_x > 1$, we follow [19] and redefine g as a single-output function $\tilde{g}$ such that $\tilde{g}(\cdot, j) = g_j(\cdot)$ and assume that $||\tilde{g}||_k \leq B_g$.

We further assume that the system is subject to polytopic state and control constraints

$$\mathcal{X} = \{x \in \mathbb{R}^{n_x} | H_x x \leq h_x,\, h_x \in \mathbb{R}^{m_x}\}, \tag{2}$$

$$\mathcal{U} = \{u \in \mathbb{R}^{n_u} | H_u u \leq h_u,\, h_u \in \mathbb{R}^{m_u}\}, \tag{3}$$

which are bounded. For example, in an autonomous driving scenario, the state region could correspond to a highway lane and the control constraints could represent the physical limits on acceleration and steering angle of the car.

Lastly, we assume access to a backup controller that guarantees that we remain inside a given safe subset of the state space once we enter it. In the autonomous driving example, this could be a simple linear controller that stabilizes the car in a small region in the center of the lane at slow speeds.

Assumption 2 *We are given a controller $\pi_{\text{safe}}(\cdot)$ and a polytopic safe region*

$$\mathcal{X}_{\text{safe}} := \{x \in \mathbb{R}^{n_x} | H_s x \leq h_s\} \subseteq \mathcal{X}, \tag{4}$$

which is (robust) control positive invariant (RCPI) under $\pi_{\text{safe}}(\cdot)$. Moreover, the controller satisfies the control constraints inside $\mathcal{X}_{\text{safe}}$, i.e. $\pi_{\text{safe}}(x) \in \mathcal{U}\, \forall x \in \mathcal{X}_{\text{safe}}$.

This assumption allows us to gather initial data from the system inside the safe region even in the presence of significant model errors, since the system remains safe under the controller π_{safe}. Moreover, we can still guarantee constraint satisfaction asymptotically outside of $\mathcal{X}_{\text{safe}}$, if we can show that a finite sequence of control inputs eventually steers the system back to the safe set $\mathcal{X}_{\text{safe}}$. This idea and a similar definition of a safe set was introduced concurrently in [7]. A set and corresponding controller which fulfill Assumption 2 for general dynamical systems is difficult to find. However, there has been recent progress in finding stability regions for systems of the form (1), which are RCPI by design, that could, under additional considerations (e.g. through polytopic inner-approximations [20]), satisfy the assumptions.

Given a controller π, ideally we want to enforce the state- and control constraints at every time step,

$$\forall t \in \mathbb{N} : f_\pi(x_t) \in \mathcal{X},\, \pi(x_t) \in \mathcal{U}, \tag{5}$$

where $x_{t+1} = f_\pi(x_t) = f(x_t, \pi(x_t))$ denotes the closed-loop system under π. Apart from π_{safe}, which trivially and conservatively fulfills this, it is in general impossible to design a controller that enforces (5) without additional assumptions. Instead, we slightly relax this requirement to *safety with high probability* throughout its operation time.

Definition 1 *Let $\pi : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ be a controller for (1) with the corresponding closed-loop system f_π. Let $x_0 \in \mathcal{X}$ and $\delta \in (0, 1]$. A system is $\delta-$safe under the controller π iff:*

$$\Pr\left[\forall t \in \mathbb{N} : f_\pi(x_t) \in \mathcal{X},\, \pi(x_t) \in \mathcal{U}\right] \geq 1 - \delta. \tag{6}$$

Based on Definition 1, the goal is to design a control scheme that guarantees δ-safety of the system (1). At the same time, we want to improve our model by learning from observations collected outside of the initial safe set $\mathcal{X}_{\text{safe}}$ during operation, which increase the performance of the controller over time.

III. Background

In this section, we introduce the necessary background on GPs and set-theoretic properties of ellipsoids that we use to model our system and perform multi-step ahead predictions.

A. Gaussian Processes (GPs)

We want to learn the unknown model-error g from data using a GP model. A $\mathcal{GP}(m,k)$ is a distribution over functions, which is fully specified through a mean function $m : \mathbb{R}^d \to \mathbb{R}$ and a *covariance function* $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, where $d = n_x + n_u$. Given a set of n noisy observations $y_i = f(z_i) + w_i$, $w_i \sim \mathcal{N}(0, \lambda^2), i = 1, \ldots, n$, $\lambda \in \mathbb{R}$, we choose a zero-mean prior on g as $m \equiv 0$ and regard the differences $\tilde{y}_n = [y_1 - h(z_1), \ldots, y_n - h(z_n)]^T$ between prior model h and observed system response at input locations $Z = [z_1, .., z_n]^T$. The posterior distribution at z is then given as a Gaussian $\mathcal{N}(\mu_n(z), \sigma_n^2(z))$ with mean and variance

$$\mu_n(z) = k_n(z)^{\mathrm{T}}[K_n + \lambda^2 I_n]^{-1}\tilde{y}_n \tag{7}$$

$$\sigma_n^2(z) = k(z,z) - k_n(z)^{\mathrm{T}}[K_n + \lambda^2 I_n]^{-1}k_n(z), \tag{8}$$

where $[K_n]_{ij} = k(z_i, z_j), [k_n(z)]_j = k(z, z_j)$, and I_n is the $n-$dimensional identity matrix. In the case of multiple outputs $n_x > 1$, we model each output dimension with an independent GP, $\mathcal{GP}(m_j, k_j), j = 1, .., n_x$. We then redefine (7) and (8) as $\mu_n(\cdot) = (\mu_{n,1}(\cdot), .., \mu_{n,n_x}(\cdot))$ and $\sigma_n(\cdot) = (\sigma_{n,1}(\cdot), .., \sigma_{n,n_x}(\cdot))$ corresponding to the predictive mean and variance functions of the individual models.

Based on Assumption 1, we can use GPs to model the unknown part of the system (1), which provides us with reliable confidence intervals on the model-error g.

Lemma 1 *[16, Lemma 2]: Assume $||g||_k \leq B_g$ and that measurements are corrupted by λ-sub-Gaussian noise. Let $\beta_n = B_g + 4\lambda\sqrt{\gamma_n + 1 + \ln(1/\delta)}$, where γ_n is the information capacity associated with the kernel k. Then with probability at least $1-\delta$ we have for all $1 \leq j \leq n_x$, $z \in \mathcal{X} \times \mathcal{U}$ that $|\mu_{n-1,j}(z) - g_j(z)| \leq \beta_n \cdot \sigma_{n-1,j}(z)$.*

In combination with the prior model $h(z)$, this allows us to construct reliable confidence intervals around the true dynamics of the system (1). The scaling β_n depends on the number of data points n that we gather from the system through the information capacity, $\gamma_n = \max_{A \subset \tilde{Z}, |A| = \tilde{n}} I(\tilde{g}_A; g)$, $\tilde{Z} = \mathcal{X} \times \mathcal{U} \times \mathcal{I}$, $\tilde{n} = n \cdot n_x$, i.e. the maximum mutual information $I(\tilde{g}_A, g)$ between a finite set of samples A and the function g. Exact evaluation of γ_n is NP-hard in general, but it can be greedily approximated and has sublinear dependence on n for many commonly used kernels [21].

The regularity assumption Assumption 1 on our model-error and the smoothness assumption on the covariance function k additionally imply that the function g is Lipschitz.

B. Ellipsoids

We use ellipsoids to give an outer bound on the uncertainty of our system when making multi-step ahead predictions. Due to appealing geometric properties, ellipsoids are widely used in the robust control community to compute *reachable sets* [22], [23]. These sets intuitively provide an outer approximation on the next state of a system considering all possible realizations of uncertainties when applying a controller to the system at a given set-valued input. We briefly review some of these properties and refer to [24] for an exhaustive introduction to ellipsoids and to the derivations for the following properties.

We use the basic definition of an ellipsoid,

$$E(p, Q) := \{x \in \mathbb{R}^n | (x-p)^{\mathrm{T}} Q^{-1} (x-p) \leq 1\}, \tag{9}$$

with *center* $p \in \mathbb{R}^n$ and a symmetric positive definite (s.p.d) *shape matrix* $Q \in \mathbb{R}^{n \times n}$. Ellipsoids are invariant under *affine subspace transformations* such that for $A \in \mathbb{R}^{r \times n}, r \leq n$ with full row rank and $b \in \mathbb{R}^r$, we have that

$$A \cdot E(p, Q) + b = E(Ap + b, AQA^{\mathrm{T}}). \tag{10}$$

The *Minkowski sum* $E(p_1, Q_1) \oplus E(p_2, Q_2)$, i.e. the pointwise sum between two arbitrary ellipsoids, is in general not an ellipsoid anymore, but we have that

$$E(p_1, Q_1) \oplus E(p_2, Q_2) \subset E\big(p_1 + p_2, (1 + c^{-1})Q_1 + (1 + c)Q_2\big) \tag{11}$$

for all $c > 0$. Moreover, the minimizer of the trace of the resulting shape matrix is analytically given as $c = \sqrt{Tr(Q_1)/Tr(Q_2)}$. A particular problem that we encounter is finding the maximum distance r to the center of an ellipsoid $E := E(0, Q)$ under a special transformation, i.e.

$$r(Q, S) = \max_{x \in E(p,Q)} ||S(x-p)||_2 = \max_{s^{\mathrm{T}} Q^{-1} s \leq 1} s^{\mathrm{T}} S^{\mathrm{T}} S s, \tag{12}$$

where $S \in \mathbb{R}^{m \times n}$ with full column rank. This is a generalized eigenvalue problem of the pair $(Q, S^T S)$ and the optimizer is given as the square-root of the largest generalized eigenvalue.

IV. Safe Model Predictive Control

In this section, we use the assumptions in Sec. II to design a control scheme that fulfills our safety requirements in Definition 1. We construct reliable, multi-step ahead predictions based on our GP model and use MPC to actively optimize over these predicted trajectories under safety constraints. Using Assumption 2, we use a terminal set constraint to theoretically prove the safety of our method.

A. Multi-step Ahead Predictions

From Lemma 1 and our prior model $h(x_t, u_t)$, we directly obtain high-probability confidence intervals on $f(x_t, u_t)$ uniformly for all $t \in \mathbb{N}$. We extend this to over-approximate the system after a sequence of inputs $(u_t, u_{t+1}, ..)$. The result is a sequence of set-valued confidence regions that contain the true dynamics of the system with high probability.

a) One-step ahead predictions: We compute an ellipsoidal confidence region that contains the next state of the system with high probability when applying a control input, given that the current state is contained in an ellipsoid. In order to approximate the system, we linearize our prior model $h(x_t, u_t)$ and use the affine transformation property (10) to compute the ellipsoidal next state of the linearized model. Next, we approximate the unknown model-error $g(x_t, u_t)$ using the confidence intervals of our GP model. We finally apply Lipschitz arguments to outer-bound the approximation

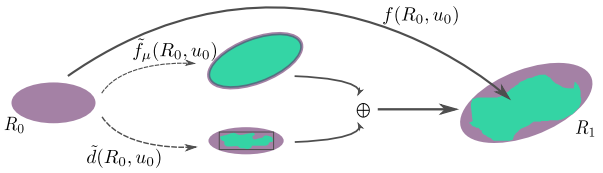

Fig. 2. Decomposition of the over-approximated image of the system (1) under an ellipsoidal input R_0. The exact, unknown image of f (right, green area) is approximated by the linearized model $\tilde{f}_\mu$ (center, top) and the remainder term $\tilde{d}$, which accounts for the confidence interval and the linearization errors of the approximation (center, bottom). The resulting ellipsoid R_1 is given by the Minkowski sum of the two individual approximations.

errors. We sum up these individual approximations, which result in an ellipsoidal approximation of the next state of the system. This is illustrated in Fig. 2. We formally derive the necessary equations in the following paragraphs. The reader may choose to skip the technical details of these approximations, which result in Lemma 2.

We first regard the system f in (1) for a single input vector $z = (x, u)$, $f(z) = h(z) + g(z)$. We linearly approximate f around $\bar{z} = (\bar{x}, \bar{u})$ via

$$f(z) \approx h(\bar{z}) + J_h(\bar{z})(z - \bar{z}) + g(\bar{z}) = \tilde{f}(z), \tag{13}$$

where $J_h(\bar{z}) = [A, B]$ is the Jacobian of h at $\bar{z}$.

Next, we use the Lagrangian remainder theorem [25] on the linearization of h and apply a continuity argument on our locally constant approximation of g. This results in an upper-bound on the approximation error,

$$|f_j(z) - \tilde{f}_j(z)| \leq \frac{L_{\nabla h,j}}{2}||z - \bar{z}||_2^2 + L_g||z - \bar{z}||_2, \tag{14}$$

where $f_j(z)$ is the ith component of f, $1 \leq j \leq n_x$, $L_{\nabla h,j}$ is the Lipschitz constant of the gradient ∇h_j, and L_g is the Lipschitz constant of g, which exists by **??**.

The function $\tilde{f}$ depends on the unknown model error g. We approximate g with the statistical GP model, $\mu_n(\bar{z}) \approx g(\bar{z})$. From Lemma 1 we have

$$|g_j(\bar{z}) - \mu_{n,j}(\bar{z})| \leq \beta_n \sigma_{n,j}(\bar{z}),\ 1 \leq j \leq n_x, \tag{15}$$

with high probability. We combine (14) and (15) to obtain

$$|f_j(z) - \tilde{f}_{\mu,j}(z)| \leq \beta_n \sigma_j(\bar{z}) + \frac{L_{\nabla h,j}}{2}||z - \bar{z}||_2^2 + L_g||z - \bar{z}||_2, \tag{16}$$

where $1 \leq j \leq n_x$ and $\tilde{f}_\mu(z) = h(\bar{z}) + J_h(\bar{z})(z - \bar{z}) + \mu_n(\bar{z})$. We can interpret (16) as the edges of the confidence hyper-rectangle

$$\tilde{m}(z) = \tilde{f}_\mu(z) \pm [\beta_n \sigma_{n-1}(\bar{z}) + \frac{L_{\nabla h}}{2}||z - \bar{z}||_2^2 + L_g||z - \bar{z}||_2], \tag{17}$$

where $L_{\nabla h} = [L_{\nabla h,1}, .., L_{\nabla h,n_x}]$ and we use the shorthand notation $a \pm b := [a_1 \pm b_1] \times [a_{n_x} \pm b_{n_x}]$, $a, b \in \mathbb{R}^{n_x}$.

We are now ready to compute a confidence region based on an ellipsoidal state $R = E(p, Q) \subset \mathbb{R}^{n_x}$ and a fixed input $u \in \mathbb{R}^{n_u}$, by over-approximating the output of the system $f(R, u) = \{f(x, u) | x \in R\}$ for ellipsoidal inputs R. Here, we choose p as the linearization center of the state and choose $\bar{u} = u$, i.e. $\bar{z} = (p, u)$. Since the function $\tilde{f}_\mu$ is affine, we can make use of (10) to compute

$$\tilde{f}_\mu(R, u) = E(h(\bar{z}) + \mu(\bar{z}), AQA^{\mathrm{T}}), \tag{18}$$

resulting again in an ellipsoid. This is visualized in Fig. 2 by the upper ellipsoid in the center. To upper-bound the confidence hyper-rectangle on the right hand side of (17), we upper-bound the term $\|z - \bar{z}\|_2$ by

$$l(R, u) := \max_{\substack{z(x)=(x,u),\\ x \in R}} ||z(x) - \bar{z}||_2, \tag{19}$$

which leads to

$$\tilde{d}(R, u) := \beta_n \sigma_{n-1}(\bar{z}) + L_{\nabla h} l^2(R, u)/2 + L_g l(R, u). \tag{20}$$

Due to our choice of $z, \bar{z}$, we have that $||z(x) - \bar{z}||_2 = ||x - p||_2$ and we can use (12) to get $l(R, u) = r(Q, I_{n_x})$, which corresponds to the largest eigenvalue of Q^{-1}. Using (19), we can now over-approximate the right side of (17) for inputs R by an ellipsoid

$$0 \pm \tilde{d}(R, u) \subset E(0, Q_{\tilde{d}}(R, u)), \tag{21}$$

where we obtain $Q_{\tilde{d}}(R, u)$ by over-approximating the hyper-rectangle $\tilde{d}(R, u)$ with the ellipsoid $E(0, Q_{\tilde{d}}(R, u))$ through $a \pm b \subset E(a, \sqrt{n_x} \cdot \mathrm{diag}([b_1, .., b_{n_x}]))$, $\forall a, b \in R^{n_x}$. This is illustrated in Fig. 2 by the lower ellipsoid in the center. Combining the previous results, we can compute the final over-approximation using (11),

$$R_+ = \tilde{m}(R, u) = \tilde{f}_\mu(R, u) \oplus E(0, Q_{\tilde{d}}(R, u)). \tag{22}$$

Since we carefully incorporated all approximation errors and extended the confidence intervals around our model predictions to set-valued inputs, we get the following generalization of Lemma 1.

Lemma 2 *Let $\delta \in (0, 1]$ and choose β_n as in Lemma 1. Then, with probability greater than $1 - \delta$, we have that:*

$$\forall x \in R : f(x, u) \in \tilde{m}(R, u), \tag{23}$$

uniformly for all $R = E(p, Q) \subset \mathcal{X}$, $u \in \mathcal{U}$.

Proof: Define $m(x, u) = h(x, u) + \mu_n(x, u) \pm \beta_n \sigma_{n-1}(x, u)$. From Lemma 1 we have $\forall R \subset \mathcal{X}$, $u \in \mathcal{U}$ that, with high probability, $\bigcup_{x \in R} f(x, u) \subset \bigcup_{x \in R} m(x, u)$. Due to the over-approximations, we have $\bigcup_{x \in R} m(x, u) \subset \tilde{m}(R, u)$. ∎

Lemma 2 allows us to compute confidence ellipsoid around the next state of the system, given that the current state of the system is given through an ellipsoidal *belief*.

b) Multi-step ahead predictions: We now use the previous results to compute a sequence of ellipsoids that contain a trajectory of the system with high-probability, by iteratively applying the one-step ahead predictions (22).

Given an initial ellipsoid $R_0 \subset \mathbb{R}^{n_x}$ and control input $u_t \in \mathcal{U}$, we iteratively compute confidence ellipsoids as

$$R_{t+1} = \tilde{m}(R_t, u_t). \tag{24}$$

We can directly apply Lemma 2 to get the following result.

Corollary 1 *Let $\delta \in (0,1]$ and choose β_n as in Lemma 1. Choose $x_0 \in R_0 \subset \mathcal{X}$. Then the following holds jointly for all $t \geq 0$ with probability at least $1-\delta$: $x_t \in R_t$, where $z_t = (x_t, u_t) \in \mathcal{X} \times \mathcal{U}$, $R_0, R_1, ..$ is computed as in* (24) *and x_t is the state of the system* (1) *at time step t.*

Proof: Since Lemma 2 holds uniformly for all ellipsoids $R \subset \mathcal{X}$ and $u \in \mathcal{U}$, this is a special case that holds uniformly for all control inputs u_t, $t \in \mathbb{N}$ and for all ellipsoids R_t, $t \in \mathbb{N}$ obtained through (24). ■

Corollary 1 guarantees that, with high probability, the system is always contained in the propagated ellipsoids (24). Thus, if we provide safety guarantees for these sequences of ellipsoids, we obtain high-probability safety guarantees for the system (1).

c) Predictions under state-feedback control laws: When applying multi-step ahead predictions under a sequence of feed-forward inputs $u_t \in \mathcal{X}$, the individual sets of the corresponding reachability sequence can quickly grow unreasonably large. This is because these *open loop* input sequences do not account for future control inputs that could correct deviations from the model predictions. Hence, we extend (22) to *affine state-feedback control laws* of the form

$$\pi_t(x_t) := K_t(x_t - p_t) + u_t, \tag{25}$$

where $K_t \in \mathbb{R}^{n_u \times n_x}$ is a feedback matrix and $u_t \in \mathbb{R}^{n_u}$ is the open-loop input. The parameter p_t is determined through the center of the current ellipsoid $R_t = E(p_t, Q_t)$. Given an appropriate choice of K_t, the control law actively contracts the ellipsoids towards their center. Similar to the derivations (13)-(22), we can compute the function $\tilde{m}$ for affine feedback controllers (25) π_t and ellipsoids $R_t = E(p_t, Q_t)$. The resulting ellipsoid is

$$\tilde{m}(R_t, \pi_t) = E(h(\bar{z}_t) + \mu(\bar{z}_t), H_t Q_t H_t^{\mathrm{T}}) \oplus E(0, Q_{\tilde{d}}(R_t, \pi_t)), \tag{26}$$

where $\bar{z}_t = (p_t, u_t)$ and $H_t = A_t + B_t K_t$. The set $E(0, Q_{\tilde{d}}(R_t, \pi_t))$ is obtained similarly to (19) as the ellipsoidal over-approximation of

$$0 \pm [\beta_n \sigma(\bar{z}) + L_{\nabla h} \frac{l^2(R_t, S_t)}{2} + L_g l(R_t, S_t)], \tag{27}$$

with $S_t = [I_{n_x}, K_t^{\mathrm{T}}]$ and $l(R_t, S_t) = \max_{x \in R_t} ||S_t(z(x) - \bar{z}_t)||_2$. The theoretical results of Lemma 2 and Corollary 1 directly apply to the case of the uncertainty propagation technique (26).

B. Safety constraints

The derived multi-step ahead prediction technique provides a sequence of ellipsoidal confidence regions around trajectories of the true system f through Corollary 1. We can guarantee that the system is safe by verifying that the computed confidence ellipsoids are contained inside the polytopic constraints (2) and (3). That is, given a sequence of feedback controllers π_t, $t = 0, .., T-1$ we need to verify

$$R_{t+1} \subset \mathcal{X},\ \pi_t(R_t) \subset \mathcal{U},\ t = 0, .., T-1, \tag{28}$$

where $(R_0, .., R_T)$ is given through (24).

Since our constraints are polytopes, we have that $\mathcal{X} = \bigcap_{i=1}^{m_x} \mathcal{X}_i$, $\mathcal{X}_i = \{x \in \mathbb{R}^{n_x} | [H_x]_{i,\cdot} x - h_i^x \leq 0\}$, where $[H_x]_{i,\cdot}$ is the ith row of H^x. We can now formulate the state constraints through the condition $R_t = E(p_t, Q_t) \subset \mathcal{X}$ as m_x individual constraints $R_t \subset \mathcal{X}_i$, $i = 1, .., m_x$, for which an analytical formulation exists [26],

$$[H_x]_{i,\cdot} p_t + \sqrt{[H_x]_{i,\cdot} Q_t [H_x]_{i,\cdot}^T} \leq h_i^x,\ \forall i \in \{1, .., m_x\}. \tag{29}$$

Moreover, we can use the fact that π_t is affine in x to obtain $\pi_t(R_t) = E(k_t, K_t Q_t, K_t^T)$, using (10). The corresponding control constraint $\pi_t(R_t) \subset \mathcal{U}$ is then equivalently given by

$$[H_u]_{i,\cdot} u_t + \sqrt{[H_u]_{i,\cdot} K_t Q_t K_t^{\mathrm{T}} [H_u]_{i,\cdot}^{\mathrm{T}}} \leq h_i^u,\ \forall i \in \{1, .., m_u\}. \tag{30}$$

C. The SafeMPC algorithm

Based on the previous results, we formulate a MPC scheme that optimizes the long-term performance of our system, while satisfying the safety condition in Definition 1:

$$\begin{aligned}
\underset{\pi_0, .., \pi_{T-1}}{\text{minimize}} \quad & J_t(R_0, .., R_T) && \text{(31a)}\\
\text{subject to} \quad & R_{t+1} = \tilde{m}(R_t, \pi_t),\ t = 0, .., T-1 && \text{(31b)}\\
& R_t \subset \mathcal{X},\ t = 1, .., T-1 && \text{(31c)}\\
& \pi_t(R_t) \subset \mathcal{U}, t = 0, .., T-1 && \text{(31d)}\\
& R_T \subset \mathcal{X}_{\mathrm{safe}}, && \text{(31e)}
\end{aligned}$$

where $R_0 := \{x_t\}$ is the current state of the system and the intermediate state and control constraints are defined in (29), (30). The terminal set constraint $R_T \subset \mathcal{X}_{\mathrm{safe}}$ has the same form as (29) and can be formulated accordingly. The objective J_t can be chosen to suit the given control task.

Due to the terminal constraint $R_T \subset \mathcal{X}_{\mathrm{safe}}$, a solution to (31) provides a sequence of feedback controllers $\pi_0, .., \pi_T$ that steer the system back to the safe set $\mathcal{X}_{\mathrm{safe}}$. We cannot directly show that a solution to MPC problem (31) exists at every time step (this property is known as recursive feasibility) without imposing additional assumption, e.g. on the safety controller π_{safe}. However, employing a control scheme similar to standard robust MPC, we guarantee that such a sequence of feedback controllers exists at every time step as follows: Given a feasible solution $\Pi_t = (\pi_t^0, .., \pi_t^{T-1})$ to (31) at time t, we apply the first feed-back control π_t^0. In case we do not find a feasible solution to (31) at the next time step, we shift the previous solution in a receding horizon fashion and append π_{safe} to the sequence to obtain $\Pi_{t+1} = (\pi_t^1, .., \pi_t^{T-1}, \pi_{\mathrm{safe}})$. We repeat this process until a new feasible solution exists that replaces the previous input sequence. This procedure is summarized in Algorithm 1. We now state the main result of the paper that guarantees the safety of our system under the proposed algorithm.

Theorem 2 *Let π be the controller defined through algorithm Algorithm 1 and $x_0 \in X_{\mathrm{safe}}$. Then the system* (1) *is $\delta-$safe under the controller π.*

Algorithm 1 Safe Model Predictive Control (SAFEMPC)

Input: Safe policy π_{safe}, dynamics model h, statistical model $\mathcal{GP}(0, k)$.
$\Pi_0 \leftarrow \{\pi_{\text{safe}}, .., \pi_{\text{safe}}\}$ with $|\Pi_0| = T$
for $t = 0, 1, ..$ **do**
  $J_t \leftarrow$ objective from high-level planner
  feasible, $\Pi \leftarrow$ solve MPC problem (31)
  if feasible **then**: $\Pi_t \leftarrow \Pi$
  else: $\Pi_t \leftarrow (\Pi_{t-1,1:T-1}, \pi_{\text{safe}})$
  $x_{t+1} \leftarrow$ apply $u_t = \Pi_{t,0}(x_t)$ to the system (1)

Proof: From Corollary 1, the ellipsoidal outer approximations (and by design of the MPC problem, also the constraints (2)) hold uniformly with high probability for *all* closed-loop systems f_Π, where Π is a feasible solution to (31), over the corresponding time horizon T. Hence we can show uniform high probability safety by induction. Base case: If (31) is infeasible, we are δ-safe using the backup controller π_{safe} of Assumption 2, since $x_0 \in \mathcal{X}_{\text{safe}}$. Otherwise the controller returned from (31) is δ-safe as a consequence of Corollary 1 and the terminal set constraint that leads to $x_{t+T} \in \mathcal{X}_{\text{safe}}$. Induction step: let the previous controller π_t be δ-safe. At time step $t+1$, if (31) is infeasible then Π_t leads to a state $x_{t+T} \in \mathcal{X}_{\text{safe}}$, from which the backup-controller is δ-safe by Assumption 2. If (31) is feasible, then the return path is δ-safe by Corollary 1. ∎

D. Optimizing long-term behavior

While the proposed MPC problem (31) yields a safe return strategy, we are often interested in a controller that optimizes performance over a possibly much longer horizon. In the autonomous driving example, a safety trajectory that stabilizes the car towards the center of the lane can be much shorter than for planning a steering maneuver before entering a turn. We hence propose to simultaneously plan a *performance trajectory* $s_0, .., s_H$ under a sequence of inputs $\pi_0^{\text{perf}}, .., \pi_{H-1}^{\text{perf}}$ using a performance-model m_{perf} along with the return strategy that we obtain when solving (31). We do not make any assumptions on the performance model which could be given by one of the approximate uncertainty propagation methods proposed in the literature (see, e.g. [11] for an overview). In order to maintain the safety of our system, we enforce that the first $r \in \{1, ., \min\{T, H\}\}$ controls are the same for both trajectories, i.e. we have that $\pi_k = \pi_k^{\text{perf}}, k = 0, .., r-1$. This extended MPC problem is

$$\begin{aligned} \underset{\substack{\pi_t,..,\pi_{t+T-1} \\ \pi_t^{\text{perf}},..,\pi_{t+H-1}^{\text{perf}}}}{\text{minimize}} \quad & J_t(s_t, .., s_{t+H}) \\ \text{subject to} \quad & \text{(31b)} - \text{(31e)}, \; t = 0, .., T-1 \\ & s_{t+1} = m_{\text{perf}}(s_t, \pi_t^{\text{perf}}), t = 0, .., H-1 \\ & \pi_t = \pi_t^{\text{perf}}, \; t = 0, .., r-1, \end{aligned} \tag{32}$$

where we replace (31) with this problem in Algorithm 1. The safety guarantees of Theorem 2 directly translate to this setting, since we can always fall back to the return strategy.

E. Discussion

Algorithm Algorithm 1 theoretically guarantees that the system remains safe, while actively optimizing for performance via the MPC problem (32). This problem can be solved by commonly used, nonlinear programming (NLP) solvers, such as the *Interior Point OPTimizer (Ipopt, [27])*. Due to the solution of the eigenvalue problem (12) that is required to compute (22), our uncertainty propagation scheme is not analytic. However, we can still obtain exact function values and derivative information by means of algorithmic differentiation, which is at the core of many state-of-the-art optimization software libraries [28].

One way to further reduce the conservatism of the multi-step ahead predictions is to linearize the GP mean prediction $\mu_n(x_t, u_t)$, which we omitted for clarity.

V. EXPERIMENTS

In this section, we evaluate the proposed SAFEMPC algorithm to safely explore the dynamics of an inverted pendulum system.

The continuous-time dynamics of the pendulum are given by $ml^2\ddot{\theta} = gml\sin(\theta) - \eta\dot{\theta} + u$, where $m = 0.15\text{kg}$ and $l = 0.5\text{m}$ are the mass and length of the pendulum, respectively, $\eta = 0.1\text{Nms/rad}$ is a friction parameter, and $g = 9.81\text{m/s}^2$ is the gravitational constant. The state of the system $x = (\theta, \dot{\theta})$ consists of the angle θ and angular velocity $\dot{\theta}$ of the pendulum. The system is controlled by a torque u that is applied to the pendulum. The origin of the system corresponds to the pendulum standing upright.

The system is underactuated with control constraints $\mathcal{U} = \{u \in \mathbb{R} | -1 \leq u \leq 1\}$. Due to these limits, the pendulum becomes unstable and falls down beyond a certain angle. We do not impose state constraints, $\mathcal{X} = \mathbb{R}^2$. However the terminal set constraint (31e) of the MPC problem (31) acts as a stability constraint and prevents the pendulum from falling. Apart from being smooth, we do not make any assumptions on our prior model h and we choose it to be a linearized and discretized approximation to the true system with a lower mass and neglected friction as in [16]. The safety controller π_{safe} is a discrete-time, infinite horizon linear quadratic regulator (LQR, [29]) of the approximated system h with cost matrices $Q = \text{diag}([1, 2])$, $R = 20$. The corresponding safety region $\mathcal{X}_{\text{Safe}}$ is given by a conservative polytopic inner-approximation of the true region of attraction of π_{safe}. We use the same mixture of linear and Matérn kernel functions for both output dimensions, albeit with different hyperparameters. We initially train our model with a dataset $(Z_0, \tilde{y}_0)$ sampled inside the safe set using the backup controller π_{Safe}. That is, we gather $n_0 = 25$ initial samples $Z_0 = \{z_1^0, .., z_{n_0}^0\}$ with $z_i^0 = (x_i, \pi_{\text{safe}}(x_i)), x_i \in \mathcal{X}_{\text{safe}}$, $i = 1, .., n$ and observed next states $\tilde{y}_0 = \{y_0^0, .., y_{n_0}^0\} \subset \mathcal{X}_{\text{safe}}$. The theoretical choice of the scaling parameter β_n for the confidence intervals in Lemma 1 can be conservative and we choose a fixed value of $\beta_n = 2$ instead, following [16].

We aim to iteratively collect the most informative samples of the system, while preserving its safety. To evaluate the exploration performance, we use the mutual information

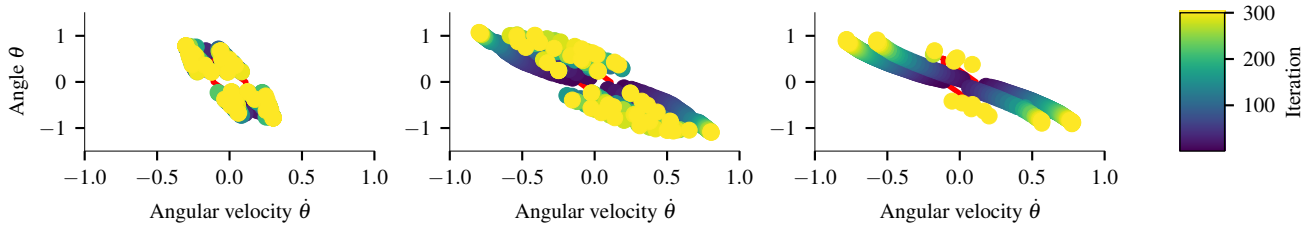

Fig. 3. Visualization of the samples acquired in the static exploration setting in Sec. V-A for $T \in \{1, 4, 5\}$. The algorithm plans informative paths to the safe set $\mathcal{X}_{\text{safe}}$ (red polytope in the center). The baseline sample set for $T = 1$ (left) is dense around origin of the system. For $T = 4$ (center) we get the optimal trade-off between cautiousness due to a long horizon and limited length of the return trajectory due to a short horizon. The exploration for $T = 5$ (right) is too cautious, since the propagated uncertainty at the final state is too large.

$I(g_{Z_n}, g)$ between the collected samples $Z_n = \{z_0, .., z_n\} \cup Z_0$ and the GP prior on the unknown model-error g, which can be computed in closed-form [21].

A. Static Exploration

For a first experiment, we assume that the system is *static*, so that we can reset the system to an arbitrary state $x_n \in \mathbb{R}^2$ in every iteration. In the static case and without terminal set constraints, a provably close-to-optimal exploration strategy is to, at each iteration n, select state-action pair z_{n+1} with the largest predictive standard deviation [21]

$$z_{n+1} = \underset{z \in \mathcal{X} \times \mathcal{U}}{\arg\max} \sum_{1 \le j \le n_x} \sigma_{n,j}(z), \tag{33}$$

where $\sigma^2_{n,j}(\cdot)$ is the predictive variance (8) of the jth $\mathcal{GP}(0, k_j)$ at the nth iteration. Inspired by this, at each iteration we collect samples by solving the MPC problem (31) with cost function $J_n = -\sum_{j=1}^{n_x} \sigma_{n,j}$, where we additionally optimize over the initial state $x_n \in \mathcal{X}$. Hence, we visit high-uncertainty states, but only allow for state-action pairs z_n that are part of a feasible return trajectory to the safe set $\mathcal{X}_{\text{Safe}}$.

Since optimizing the initial state is highly non-convex, we solve the problem iteratively with 25 random initializations to obtain a good approximation of the global minimizer. After every iteration, we update the sample set $Z_{n+1} = Z_n \cup \{z_n\}$, collect an observation (z_n, y_n) and update the GP models. We apply this procedure for varying horizon lengths.

The resulting sample sets are visualized for varying horizon lengths $T \in \{1, .., 5\}$ with 300 iterations in Fig. 3, while Fig. 4 shows how the mutual information of the sample sets Z_i, $i = 0, .., n$ for the different values of T. For short time horizons ($T = 1$), the algorithm can only slowly explore, since it can only move one step outside of the safe set. This is also reflected in the mutual information gained, which levels off quickly. For a horizon length of $T = 4$, the algorithm is able to explore a larger part of the state-space, which means that more information is gained. For larger horizons, the predictive uncertainty of the final state is too large to explore effectively, which slows down exploration initially, when we do not have much information about our system. The results suggest that our approach could further benefit from adaptively choosing the horizon during operation, e.g. by employing a variable horizon MPC approach [30], or by increasing the horizon when the mutual information saturates for the current horizon.

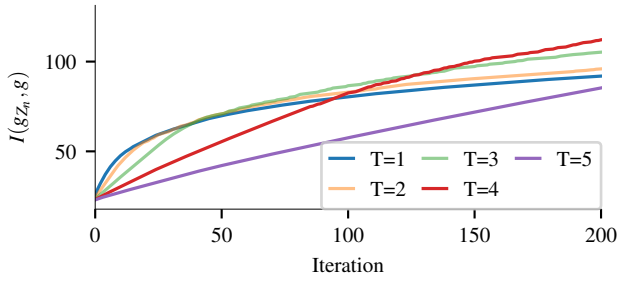

Fig. 4. Mutual information $I(g_{Z_n}, g)$, $n = 1, .., 200$ for horizon lengths $T \in \{1, .., 5\}$. Exploration settings with shorter horizon gather more informative samples at the beginning, but less informative samples in the long run. Longer horizon lengths result in less informative samples at the beginning, due to uncertainties being propagated over long horizons. However, after having gathered some knowledge they quickly outperform the smaller horizon settings. The best trade off is found for $T = 4$.

B. Dynamic Exploration

As a second experiment, we collect informative samples during operation; without resetting the system at every iteration. Starting at $x_0 \in \mathcal{X}_{\text{safe}}$, we apply the SafeMPC, Algorithm 1, over 200 iterations. We consider two settings. In the first, we solve the MPC problem (31) with $-J_n$ given by (33), similar to the previous experiments. In the second setting, we additionally plan a performance trajectory as proposed in Sec. IV-D. We define the states of the performance trajectory as Gaussians $s_t = \mathcal{N}(m_t, S_t) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x \times n_x}$ and the next state is given by the predictive mean and variance of the current state m_t and applied action u_t. That is, $s_{t+1} = \mathcal{N}(m_{t+1}, S_{t+1})$ with

$$m_{t+1} = \mu_n(m_t, u_t),\ S_{t+1} = \Sigma_n(m_t, u_t), t = 0, .., H-1,$$

where $\Sigma_n(\cdot) = \text{diag}(\sigma^2_n(\cdot))$ and $m_0 = x_n$. This simple approximation technique is known as *mean-equivalent* uncertainty propagation. We define the cost-function $-J_t = \sum_{t=0}^{H} \text{trace}(S_t^{1/2}) - \sum_{t=1}^{T} (m_t - p_t)^T Q_{\text{perf}} (m_t - p_t)$, which maximizes the sum of predictive confidence intervals along the trajectory $s_1, .., s_H$, while penalizing deviation from the safety trajectory. We choose $r = 1$ in the problem (32),

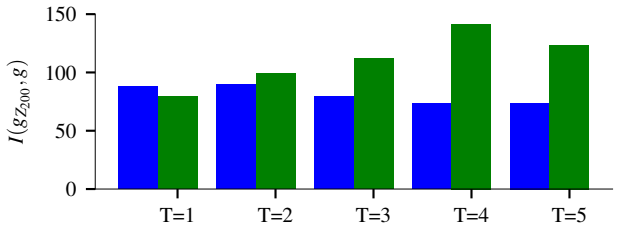

Fig. 5. Comparison of the information gathered from the system after 200 iterations for the standard setting (blue) and the setting where we plan an additional performance trajectory (green).

i.e. the first action of the safety trajectory and performance trajectory are the same. As in the static setting, we update our GP models after every iteration.

We evaluate both settings for varying $T \in \{1, .., 5\}$ and fixed $H = 5$ in terms of their mutual information in Fig. 5. We observe a similar behavior as in the static exploration experiments and get the best exploration performance for $T = 4$ with a slight degradation of performance for $T = 5$. We can see that, except for $T = 1$, the performance trajectory decomposition setting consistently outperforms the standard setting. Planning a performance trajectory (green) provides the algorithm with an additional degree of freedom, which leads to drastically improved exploration performance.

VI. Conclusion

We introduced SafeMPC, a learning-based MPC scheme that can safely explore partially unknown systems. The algorithm is based on a novel uncertainty propagation technique that uses a reliable statistical model of the system. As we gather more data from the system and update our statistical mode, the model becomes more accurate and control performance improves, all while maintaining safety guarantees throughout the learning process.

References

[1] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1054–1054, 1998.

[2] C. Xie, S. Patil, T. Moldovan, S. Levine, and P. Abbeel, "Model-based reinforcement learning with parametrized physical models and optimism-driven exploration," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 504–511.

[3] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Pub., 2009.

[4] S. Sadraddini and C. Belta, "A provably correct MPC approach to safety control of urban traffic networks," in *American Control Conference (ACC)*, 2016, pp. 1679–1684.

[5] J. M. Carson, B. Açıkmeşe, R. M. Murray, and D. G. MacMartin, "A robust model predictive control algorithm augmented with a reactive safety mode," *Automatica*, vol. 49, no. 5, pp. 1251–1260, 2013.

[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] K. P. Wabersich and M. N. Zeilinger, "Linear model predictive safety certification for learning-based control," in *Proc. of the Conference on Decision and Control (CDC)*, 2018.

[8] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. MIT Press, Cambridge MA, 2006.

[9] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard, "Gaussian process model based predictive control," in *Proc. of the American Control Conference (ACC)*, vol. 3, 2004, pp. 2214–2219.

[10] G. Cao, E. M.-K. Lai, and F. Alam, "Gaussian process model predictive control of an unmanned quadrotor," *Journal of Intelligent & Robotic Systems*, vol. 88, no. 1, pp. 147–162, 2017.

[11] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious NMPC with gaussian process dynamics for autonomous miniature race cars," in *In Proc. of the European Control Conference (ECC)*, 2018.

[12] A. Jain, T. X. Nghiem, M. Morari, and R. Mangharam, "Learning and control using Gaussian processes: Towards bridging machine learning and controls for physical systems," in *Proc. of the International Conference on Cyber-Physical Systems*, 2018, pp. 140–149.

[13] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Robust constrained learning-based NMPC enabling reliable mobile robot path tracking," *The International Journal of Robotics Research*, vol. 35, no. 13, pp. 1547–1563, 2016.

[14] M. P. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *Proc. of the International Conference on Machine Learning*, 2011, pp. 465–472.

[15] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel, "Reinforcement learning versus model predictive control: A comparison on a power system problem," *In IEEE Transactions on Systems, Man, and Cybernetics*, vol. 39, no. 2, pp. 517–529, 2009.

[16] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," *Proc. of Neural Information Processing Systems (NIPS)*, vol. 1705, 2017.

[17] 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 *In Proc. of the IEEE Conference on Decision and Control (CDC)*, 2014, pp. 1424–1431.

[18] G. Wahba, *Spline Models for Observational Data*. Siam, 1990, vol. 59.

[19] F. Berkenkamp, A. Krause, and A. P. Schoellig, "Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics," *arXiv:1602.04450 [cs]*, 2016.

[20] E. M. Bronstein, "Approximation of convex sets by polytopes," *Journal of Mathematical Sciences*, vol. 153, no. 6, pp. 727–762, 2008.

[21] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proc. of the International Conference on Machine Learning (ICML)*, 2010, pp. 1015–1022.

[22] T. F. Filippova, "Ellipsoidal estimates of reachable sets for control systems with nonlinear terms," *Proc. of the International Federation of Automatic Control (IFAC)*, vol. 50, no. 1, pp. 15 355–15 360, 2017.

[23] L. Asselborn, D. Gross, and O. Stursberg, "Control of uncertain nonlinear systems using ellipsoidal reachability calculus," *Proc. of the International Federation of Automatic Control (IFAC)*, vol. 46, no. 23, pp. 50–55, 2013.

[24] A. B. Kurzhanskii and I. Vályi, *Ellipsoidal Calculus for Estimation and Control*. Boston, MA : Birkhäuser, 1997.

[25] L. Breiman and A. Cutler, "A deterministic algorithm for global optimization," *Mathematical Programming*, vol. 58, no. 1-3, pp. 179–199, 1993.

[26] D. H. van Hessem and O. H. Bosgra, "Closed-loop stochastic dynamic process optimization under input and state constraints," in *Proc. of the American Control Conference (ACC)*, vol. 3, 2002, pp. 2023–2028.

[27] 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, no. 1, pp. 25–57, 2006.

[28] J. Andersson, "A general-purpose software framework for dynamic optimization," PhD Thesis, Arenberg Doctoral School, KU Leuven, Leuven, Belgium, 2013.

[29] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*. Wiley-interscience New York, 1972, vol. 1.

[30] Richards Arthur and How Jonathan P., "Robust variable horizon model predictive control for vehicle maneuvering," *International Journal of Robust and Nonlinear Control*, vol. 16, no. 7, pp. 333–351, 2006.