Cautious Model Predictive Control Using Gaussian Process Regression

Journal Article

Author(s):
Hewing, Lukas; Kabzan, Juraj; Zeilinger, Melanie N.

Publication date:
2020-11

Permanent link:
https://doi.org/10.3929/ethz-b-000389524

Rights / license:
In Copyright - Non-Commercial Use Permitted

Originally published in:
IEEE Transactions on Control Systems Technology 28(6), https://doi.org/10.1109/tcst.2019.2949757

Funding acknowledgement:
157601 - Safety and Performance for Human in the Loop Control (SNF)

Cautious Model Predictive Control using Gaussian Process Regression

Lukas Hewing, Juraj Kabzan and Melanie N. Zeilinger

***Abstract*—Gaussian process (GP) regression has been widely used in supervised machine learning due to its flexibility and inherent ability to describe uncertainty in function estimation. In the context of control, it is seeing increasing use for modeling of nonlinear dynamical systems from data, as it allows the direct assessment of residual model uncertainty. We present a model predictive control (MPC) approach that integrates a nominal system with an additive nonlinear part of the dynamics modeled as a GP. We describe a principled way of formulating the chance constrained MPC problem, which takes into account residual uncertainties provided by the GP model to enable cautious control. Using additional approximations for efficient computation, we finally demonstrate the approach in a simulation example, as well as in a hardware implementation for autonomous racing of remote controlled race cars bluewith fast sampling times of 20 ms, highlighting improvements with regard to both performance and safety over a nominal controller.**

***Index Terms*—Model Predictive Control, Gaussian Processes, Learning-based Control, Model Learning, Autonomous Racing**

I. Introduction

Many modern control techniques depend on accurate model descriptions to enable safe and high performance control. Identifying these models, especially for nonlinear systems, is a time-consuming and complex endeavor. It is, however, often possible to derive an approximate system model, e.g. a linear model with adequate accuracy close to some operating point, or a simple model description from first principles. In addition, measurement data from previous experiments or during operation is often available, which can be exploited to enhance the system model and controller performance. In this paper, we present a model predictive control (MPC) approach, which improves such a nominal model description from data using Gaussian Processes (GPs) to safely enhance performance of the system.

The appeal of Gaussian Process regression for model learning stems from the fact that it requires little prior process knowledge and directly provides a measure of residual model uncertainty. In predictive control, GPs were successfully applied to improve control performance when learning periodic time-varying disturbances [1]. The task of learning the system dynamics as opposed to disturbances, has first been presented in [2]. In [3] a piecewise-linear approximate explicit solution for the GP-based MPC problem of a combustion plant was presented. Application of a one-step MPC with a GP model to a mechatronic system was demonstrated in [4] and the use for fault-tolerant MPC was presented in [5]. An application to high performance driving using a sparse spectrum approximation of the GP was shown in [6] and constrained tracking MPC for robotic applications in [7], [8], where uncertainty of the GP prediction is taken into account using confidence bounds. Data efficiency of these formulations for learning controllers was demonstrated in [9].

Unlike most previous approaches, we specifically consider a hybrid model structure by combining a nominal system description, which allows for rudimentary system operation in order to collect data, with an additive GP part that can be of different dimensionality and is used for learning specific dynamical effects. The paper makes the following contributions. We discuss approximate propagation of system uncertainty and the principled formulation of chance constraints in terms of probabilistic reachable sets [10], resulting in a deterministic approximation of the stochastic optimal control problem suitable for numerical optimization. The nominal system description allows for reducing the GP model learning to a subspace of states and inputs, reducing the dimensionality of the machine learning task. By introducing *dynamic sparse GPs* based on inducing points as an approximation technique tailored to MPC, we further reduce the computational burden of the approach. Together, this results in an MPC formulation which allows for the control of high performance systems at sampling rates of few milliseconds, which we demonstrate in two application examples. The first is a simulation of an autonomous underwater vehicle, illustrating key concepts and advantages in a simplified setting. Second, we present a hardware implementation for autonomous racing of remote controlled cars, showing the real-world feasibility of the approach for complex high performance control tasks. To the best of our knowledge this is the first hardware implementation of a Gaussian Process-based predictive control scheme to a system of this complexity at sampling times of 20 ms.

II. Preliminaries

A. Notation

The i-th element of a vector x is denoted by $[x]_i$. Similarly, $[M]_{ij}$ denotes element ij of a matrix M, and $[M]_{\cdot i}$, $[M]_{i\cdot}$ its i-th column or row, respectively. We use $\operatorname{diag}(x)$ for a diagonal matrix with entries given by the vector x. The quadratic form $x^\mathsf{T} M x$ is denoted by $\|x\|_M^2$ and $\nabla f(z)|_{\bar{z}}$ is the gradient of f evaluated at $\bar{z}$. The Pontryagin set difference is denoted $\mathcal{A} \ominus \mathcal{B} = \{a \,|\, a + b \in \mathcal{A} \; \forall b \in \mathcal{B}\}$. A normally

Manuscript received November 8, 2018; revised August 20, 2019. Manuscript received in final form October #, 2019. Recommended by Associate Editor #. This work was supported by the Swiss National Science Foundation under grant no. PP00P2 157601 / 1.

All authors are members of the Institute for Dynamic Systems and Control, ETH Zürich, Zürich CH-8092, Switzerland (e-mail: lhewing|kabzanj|mzeilinger@ethz.ch)

distributed vector x with mean μ and variance Σ is given by $x \sim \mathcal{N}(\mu, \Sigma)$ and the expected value of x is $\mathbb{E}(x)$. We use $\Pr(E)$, $(\Pr(E \,|\, A))$ to refer the probability of an event E (given A). Realized quantities during closed loop control are time indexed using parenthesis, i.e. $x(k)$, while quantities in prediction use subscripts, e.g. x_i is the predicted state i-steps ahead.

B. Problem formulation

We consider the control of dynamical systems that can be represented by a discrete-time model of the form

$$x(k+1) = f(x(k), u(k)) + B_d(g(x(k), u(k)) + w(k)) \,, \quad (1)$$

where $x(k) \in \mathbb{R}^{n_x}$ is the system state and $u(k) \in \mathbb{R}^{n_u}$ the control inputs at time k. The model is composed of a known nominal part f and additive term g that describes initially unknown dynamics of the system, which are to be learned from data and are assumed to lie in the subspace spanned by B_d. We consider i.i.d. process noise $w(k) \sim \mathcal{N}(0, \Sigma^w)$, which is spatially uncorrelated, i.e. has diagonal variance matrix $\Sigma^w = \text{diag}([\sigma_1^2, \ldots, \sigma_{n_d}^2])$. We assume that both f and g are differentiable functions. The system is subject to state and input constraints $\mathcal{X}_k \subseteq \mathbb{R}^{n_x}$, $\mathcal{U}_k \subseteq \mathbb{R}^{n_u}$, respectively, which are formulated as chance constraints

$$\Pr(x(k) \in \mathcal{X}_k) \geq p_x \,,$$
$$\Pr(u(k) \in \mathcal{U}_k) \geq p_u \,,$$

where p_x, p_u are the associated satisfaction probabilities.

C. Gaussian Process Regression

We use Gaussian processes regression to infer the noisy vector-valued function $g(x, u)$ in (1) using M data points generated from previously collected measurements of states and inputs x_j, u_j as

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

where $B_d^\dagger$ is the Moore-Penrose pseudo-inverse. Note that the measurement noise on the data points w_j corresponds to the process noise in (1). With $z_j := [x_j^\mathsf{T}, u_j^\mathsf{T}]^\mathsf{T}$, the data set is

$$\mathcal{D} = \{\mathbf{y} = [y_0, \ldots, y_M]^\mathsf{T} \in \mathbb{R}^{M \times n_d},$$
$$\mathbf{z} = [z_0, \ldots, z_M]^\mathsf{T} \in \mathbb{R}^{M \times n_z}\} \,.$$

Each output dimension is learned individually, meaning that we assume the components of each y_j to be independent, given the input data z_j. Specifying a GP prior on g in each output dimension $a \in \{1, \ldots, n_d\}$ with kernel $k^a(\cdot, \cdot)$, prior mean function $m^a(\cdot)$ and conditioning on the data D results in a predictive posterior distribution in each dimension a at a test point z which is Gaussian with mean and variance

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

Here K^a is the Gram matrix, i.e. $[K_{\mathbf{z}\mathbf{z}}^a]_{ij} = k^a(z_i, z_j)$, $[K_{\mathbf{z}z}^a]_j = k^a(z_j, z)$, $K_{z\mathbf{z}}^a = (K_{\mathbf{z}z}^a)^\mathsf{T}$, $K_{zz}^a = k^a(z, z)$ and $m^a(\mathbf{z}) = [m^a(z_0), \ldots, m^a(z_M)]^\mathsf{T}$. The choice of kernel function k^a and its parameterization is the determining factor for the inferred distribution of g and is typically specified using prior process knowledge and optimization [11], e.g. by optimizing the likelihood of the observed data points. Throughout this paper we consider the squared exponential kernel function

$$k^a(z_i, z_j) = \sigma_{f,a}^2 \exp\left(-\frac{1}{2}(z_i - z_j)^\mathsf{T} L_a^{-1} (z_i - z_j)\right) \,, \quad (3)$$

where L_a is a positive diagonal length-scale matrix and $\sigma_{f,a}^2$ the signal variance. While it is straightforward to use any other (differentiable) kernel function, we have observed good results in practice with this commonly employed kernel and maximum likelihood hyperparameter tuning.

The resulting multivariate GP approximation of the unknown function $g(z)$ is then given by stacking the individual output dimensions, i.e.

$$d(z) \sim \mathcal{N}\big(\mu^d(z), \Sigma^d(z)\big) \quad (4)$$

with $\mu^d = [\mu_1^d, \ldots, \mu_{n_d}^d]^\mathsf{T}$ and $\Sigma^d = \text{diag}([\Sigma_1^d, \ldots, \Sigma_{n_d}^d]^\mathsf{T})$.

Evaluating mean and variance in (2) has cost $\mathcal{O}(n_d n_z M)$ and $\mathcal{O}(n_d n_z M^2)$, respectively and thus scales with the number of data points. For large amounts of data or fast real-time applications this can limit the use of GP models.

D. Sparse Gaussian Processes

Various approximation techniques have been proposed to reduce the computational complexity of GPs, for instance *sparse spectrum* approximation [12], which was shown in the context of MPC in [6]. In this paper, we make use of sparse GPs using inducing inputs [13], for which we present an extension tailored to MPC in Section III-B. These approximations make use of *inducing* targets $\mathbf{y}_{\text{ind}}$, inputs $\mathbf{z}_{\text{ind}}$ and conditional distributions to approximate the resulting prediction. Of the many variants available, we apply the frequently used Fully Independent Training Conditional (FITC) [14]. Given a selection of inducing inputs $\mathbf{z}_{\text{ind}}$ and using the shorthand notation $Q_{\zeta\tilde{\zeta}}^a := K_{\zeta\mathbf{z}_{\text{ind}}}^a (K_{\mathbf{z}_{\text{ind}}\mathbf{z}_{\text{ind}}}^a)^{-1} K_{\mathbf{z}_{\text{ind}}\tilde{\zeta}}^a$ the approximate posterior distribution is given by

$$\tilde{\mu}_a^d(z) = Q_{z\mathbf{z}}^a (Q_{\mathbf{z}\mathbf{z}}^a + \Lambda)^{-1} [\mathbf{y}]_{\cdot a} \,, \quad (5a)$$
$$\tilde{\Sigma}_a^d(z) = K_{zz}^a - Q_{z\mathbf{z}}^a (Q_{\mathbf{z}\mathbf{z}}^a + \Lambda)^{-1} Q_{\mathbf{z}z} \quad (5b)$$

with $\Lambda = \text{diag}(K_{\mathbf{z}\mathbf{z}}^a - Q_{\mathbf{z}\mathbf{z}}^a + I\sigma_a^2)$. Several of the matrices used in (5) can be precomputed, making the evaluation complexity independent of the number of original data points. With $\tilde{M}$ being the number of inducing points, this results in $\mathcal{O}(n_d n_z \tilde{M})$ and $\mathcal{O}(n_d n_z \tilde{M}^2)$ for the predictive mean and variance, respectively.

There are numerous options for selecting the inducing inputs, e.g. heuristically as a subset of the original data points, by treating them as hyperparameters and optimizing over their location [14], or letting them coincide with test points [15], which is often referred to as *transductive* learning. In Section III-B we make use of such transductive ideas and propose a dynamic selection of inducing points, with a resulting local approximation tailored to the predictive control task.

III. MPC Controller Design

We consider the design of an MPC controller for system (1) using the GP approximation d of the unknown function g:

$$x_{i+1} = f(x_i, u_i) + B_d \left(d(x_i, u_i) + w_i \right) . \tag{6}$$

At each time step, the GP approximation evaluates to a stochastic distribution according to the residual model uncertainty and process noise, which is then propagated forward in time. A stochastic MPC formulation allows the principled treatment of chance constraints by imposing a prescribed maximum probability of constraint violation. The resulting stochastic optimal control problem is defined as

$$\begin{aligned}
\min_{\Pi} \quad & \mathbb{E}\left(l_f(x_N) + \sum_{i=0}^{N-1} l_i(x_i, u_i) \right) && \text{(7a)}\\
\text{s.t.} \quad & x_{i+1} = f(x_i, u_i) + B_d(d(x_i, u_i) + w_i) && \text{(7b)}\\
& u_i = \pi_i(x_i) && \text{(7c)}\\
& \Pr(x_{i+1} \in \mathcal{X}_{i+1}) \geq p_x && \text{(7d)}\\
& \Pr(u_i \in \mathcal{U}_i) \geq p_u && \text{(7e)}\\
& x_0 = x(k) && \text{(7f)}
\end{aligned}$$

for all $i = 0, \ldots, N-1$ with terminal cost $l_f(x_N)$ and stage cost $l_i(x_i, u_i)$, where the optimization is carried out over a sequence of input policies $\Pi = \{\pi_0(x), \ldots, \pi_{N-1}(x)\}$. Optimization over feedback policies $\Pi(x)$ instead of an input sequence can effectively reduce the predicted uncertainty over the prediction horizon, but is in general an infinite dimensional optimization problem and not computationally tractable. We restrict the policy class to linear state feedback controllers

$$\pi_i(x_i) = K_i(\mu_i^x - x_i) + \mu_i^u ,$$

where μ_i^x is the predicted mean of the state distribution. Online optimization of the feedback gains K_i is typically computationally prohibitive, such that we pre-select K_i and optimize over the mean of the applied input μ_i^u. An equivalent choice is given by linear disturbance feedback [16], which is popular in particular for linear systems, where it allows for convex optimization over the feedback gains [17]. The adequate choice of linear feedback gains K_i is generally a hard problem, especially if the system dynamics are highly nonlinear. A useful heuristic is to consider a linearization of the system dynamics around an approximate prediction trajectory, which in MPC applications is typically available using the solution trajectory of the previous time step. Feedback gains for the linearized system can be derived e.g. by solving a finite horizon LQR problem [18]. For mild or stabilizing nonlinearities, a fixed controller gain $K_i = K$ can be chosen to reduce computational burden, or the linear feedback can be omitted at the potential cost of added conservatism.

Due to the stochastic prediction model, optimization problem (7) is computationally intractable. In the following sections, we present techniques for deriving an efficiently solvable approximation. Specifically, we show techniques for simple approximate propagation of the system uncertainties in terms of mean and variance and present a framework for reformulating chance constraints deterministically, enabling the use of standard nonlinear programming solvers.

A. Uncertainty Propagation

Because of stochastic process noise and the representation by a GP model, future predicted states result in stochastic distributions. Evaluating the posterior of a GP from an input distribution is generally intractable and the resulting distribution is not Gaussian [19]. While under certain assumptions on g, some strict over-approximations of resulting confidence sets exist [20], they are typically conservative and computationally demanding. We focus instead on computationally cheap and practical approximations at the cost of losing strict guarantees.

Following a common approach, state, control input and nonlinear disturbance are approximated as jointly Gaussian distributed at each time step. Using a first-order Taylor approximation of the nominal dynamics f and posterior mean function (2), similar to extended Kalman filtering, this permits simple update equations for the state mean and variance based on affine transformations of the Gaussian distributions. This provides a good trade-off between approximation accuracy and computational complexity, resulting in

$$\mu_{i+1}^x = f(\mu_i^x, \mu_i^u) + B_d \mu^d(\mu_i^x, \mu_i^u) , \tag{8a}$$

$$\Sigma_{i+1}^x = \tilde{A}_i \Sigma_i^x \tilde{A}_i^\mathsf{T} + B_d(\Sigma^d(\mu_i^x, \mu_i^u) + \Sigma^w) B_d^\mathsf{T} \tag{8b}$$

with $\tilde{A}_i = \nabla(f(x, \mu_i^u + K_i x) + B_d \mu^d(x, \mu_i^u + K_i x))|_{\mu_i^x}$, while the input variance is given by $\Sigma_i^u = K_i \Sigma_i^x K_i^\mathsf{T}$.

Due to the GP dynamics, the computational complexity when evaluating these predictions scale directly with the input and output dimensions, as well as the number of data points and thus becomes expensive to evaluate in high dimensional spaces. This presents a challenge for predictive control approaches with GP models, which in the past have mainly focused on relatively small and slow systems [2], [5]. Using a nominal model, however, it is possible to consider GPs that depend on only a subset of states and inputs, such that the computational burden can be significantly reduced. This is due to a reduction in the effective input dimension n_z, and more importantly due to a reduction in necessary training points M for learning in a lower dimensional space.

Remark 1: With slight notational changes the presented approximation similarly applies to prior mean and kernel functions that are functions of a subset of states and inputs.

A further significant reduction in computational complexity can be achieved by employing sparse GP approaches as outlined in Section II-D, for which we present a variant tailored to predictive control in the following.

B. Dynamic Sparse GPs for MPC

Inducing points in sparse GP approximations are typically chosen to optimize predictive quality *globally*, i.e. on the entire domain of possible test points. Given knowledge about future test points, however, one can instead aim to enhance the prediction quality *locally*, resulting in good approximations in a specific regions of interest. In MPC, some knowledge of these test points is typically available in terms of an approximate trajectory $\tilde{X}$, $\tilde{U}$ in the state-action space. This trajectory can, e.g., be given by the reference signal or a previous solution trajectory of the MPC problem, since solution trajectories in MPC typically do not change drastically between time steps, given a reasonably long prediction horizon and fast sampling times. Ideally, the inducing inputs would therefore be optimized to maximize predictive quality on the approximate trajectory $\tilde{X}$, $\tilde{U}$, which, however, is not computationally feasible in the targeted millisecond sampling times. Instead, we place inducing inputs heuristically along the approximate trajectory, resulting in a computationally cheap high fidelity local approximation around $\tilde{X}$, $\tilde{U}$.

To illustrate the procedure, we consider a simple double integrator system controlled by an MPC. Fig. 1 shows the variance $\Sigma^d(x)$ of a GP trained on the system states. Additionally, two successive trajectories from an MPC algorithm are displayed. The solid red line is the current prediction, while the dashed line is the prediction trajectory from the previous iteration, which we use as $\tilde{X}$, $\tilde{U}$. The plot on the left displays the original GP, with data points marked as crosses, whereas on the right we have the sparse approximation resulting from placing inducing points along $\tilde{X}$, $\tilde{U}$, indicated by circles. The figure illustrates how the full GP and the sparse approximation match closely along the predicted trajectory of the system, while approximation quality far away from the trajectory deteriorates. Since current and previous trajectory are similar, however, local information is sufficient for computation of the MPC controller.

C. *Chance Constraint Formulation*

The tractable Gaussian approximation of the state and input distribution over the prediction horizon in (8) can be used to approximate the chance constraints in (7d) and (7e). We reformulate the constraints on state and input w.r.t. their means μ^x, μ^u using constraint tightening based on the respective errors $e_i^x = \mu_i^x - x_i$ and $e_i^u = K_i(\mu_i^x - x_i)$. For this purpose, we make use of probabilistic reachable sets [10], an extension of the concept of reachable sets to stochastic systems, which is related to probabilistic set invariance [22], [23].

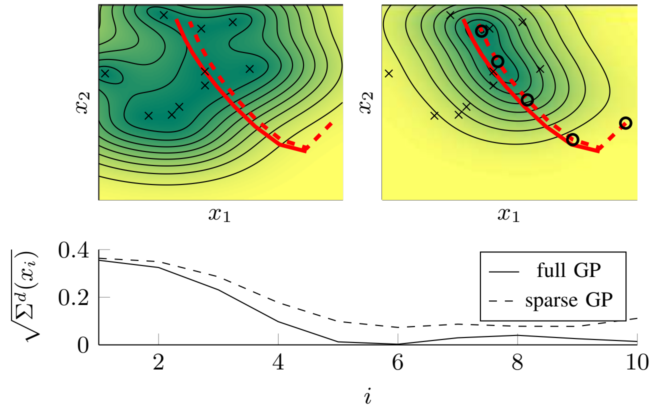


Fig. 1: Illustration of dynamic sparse approximation [21]. Contour plot of the posterior variance of the full GP (top left) and dynamic sparse approximation (top right) with data points as black crosses. Red lines show trajectories planned by an MPC, while the dashed lines show the prediction at the previous time step used in the approximation, with inducing points indicated by black circles. The bottom plot shows the respective variances along the planned trajectory.

Definition 1 (Probabilistic i-step Reachable Set): A set $\mathcal{R}_i$ is said to be a probabilistic i-step reachable set (i-step PRS) of probability level p if

$$\Pr(e_i \in \mathcal{R}_i \,|\, e_0 = 0) \geq p \,.$$

Given an i-step PRS $\mathcal{R}_i^x$ of probability level p_x for the state error e_i^x and similarly $\mathcal{R}_i^u$ for the input e_i^u, we can define tightened constraints on μ_i^x and μ_i^u as

$$\mu_i^x \in \mathcal{Z}_i = \mathcal{X}_i \ominus \mathcal{R}_i^x \,, \tag{9a}$$

$$\mu_i^u \in \mathcal{V}_i = \mathcal{U}_i \ominus \mathcal{R}_i^u \,, \tag{9b}$$

where $\ominus$ denotes the Pontryagin set difference. Satisfaction of the tightened constraints (9) for the mean thereby implies satisfaction of the original constraints (7d) and (7e), i.e. when $\mu_i^x \in \mathcal{Z}$ we have $\Pr(x_i = \mu_i^x + e_i^x \in \mathcal{X}) \geq \Pr(e_i^x \in \mathcal{R}^x) \geq p_x$.

Under the approximation of a normal distribution of x_i, the uncertainty in each time step is fully specified by the variance matrices Σ_i^x and Σ_i^u. The sets can then be computed as functions of these variances, i.e. $\mathcal{R}^x(\Sigma_i^x)$ and $\mathcal{R}^u(\Sigma_i^x)$. We present two specific formulations for half-space and 2-norm ball constraints in the following, concentrating on state constraints, as input constraints can be treated analogously.

1) Half-space constraints: Consider the constraint set $\mathcal{X}_i$ given by a single half-space constraint $\mathcal{X}_i^{hs} := \left\{x \,\middle|\, h_i^\mathsf{T} x \leq b_i\right\}$, $h_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}_+$. The marginal distribution of the error in the direction of the half-space $h_i^\mathsf{T} e_i^x \sim \mathcal{N}(0, h_i^\mathsf{T} \Sigma_i^x h_i)$ allows the use of the quantile function of a standard Gaussian random variable $\phi^{-1}(p_x)$ at the needed probability of constraint satisfaction p_x, such that

$$\mathcal{R}^x(\Sigma_i^x) := \left\{ e \,\middle|\, h_i^\mathsf{T} e \leq \phi^{-1}(p_x) \sqrt{h_i^\mathsf{T} \Sigma_i^x h_i} \right\}$$

is an i-step PRS of probability level p_x. In this case, evaluating the Pontryagin difference in (9) is straightforward and we can directly define the tightened constraint on the state mean as

$$\mathcal{Z}_i^{hs}(\Sigma_i^x) := \left\{ z \,\middle|\, h_i^\mathsf{T} z \leq b_i - \phi^{-1}(p_x) \sqrt{h_i^\mathsf{T} \Sigma_i^x h_i} \right\} .$$

Remark 2: A tightening for slab constraints $\mathcal{X}^{sl} = \left\{x \,\middle|\, |h_i^\mathsf{T} x| \leq b_i\right\}$ can be similarly derived as

$$\mathcal{Z}_i^{sl}(\Sigma_i^x) := \left\{ z \,\middle|\, |h_i^\mathsf{T} z| \leq b_i - \phi^{-1}\left(\frac{p_x + 1}{2}\right) \sqrt{h_i^\mathsf{T} \Sigma_i^x h_i} \right\} .$$

2) 2-Norm Ball Constraints: Consider the constraint set $\mathcal{X}_i$ given as a norm ball $\mathcal{X}_i^b := \{x \,|\, \|x - x_i^c\|_2 \leq r_i\}$, $r \in \mathbb{R}_+$. Based on the approximate Gaussian distributions we define

$$\mathcal{R}^{ell}(\Sigma_i^x) = \left\{ e \,\middle|\, (e^\mathsf{T} (\Sigma_i^x)^{-1} e \leq \chi_n^2(p_x) \right\} ,$$

where $\chi_n^2(p_x)$ is the quantile function of the chi-squared distribution with n degrees of freedom. In order to simplify constraint tightening of $\mathcal{X}_i^b$, we define an outer approximation by a ball $\mathcal{R}^b \supseteq \mathcal{R}^{ell}$ as

$$\mathcal{R}^b(\Sigma_i^x) = \left\{ e \,\middle|\, \|e\| \leq \sqrt{\lambda_{\max}(\Sigma_i^x)\, \chi_n^2(p_x)} \right\} ,$$

where $\lambda_{\max}(\cdot)$ is the maximum eigenvalue. The necessary constraint tightening can then directly be expressed as

$$\mathcal{Z}_i^b(\Sigma_i^x) = \mathcal{X}_i \ominus \mathcal{R}^b(\Sigma_i^x) = \{z \,|\, \|z - x_i^c\| \le \tilde{r}_i\left(\Sigma_i^x\right)\} \ , \quad (10)$$

where $\tilde{r}_i\big(\Sigma_i^{XY}\big) = r_i - \sqrt{\chi_n^2(p_x)\lambda_{\max}(\Sigma_i^x)}$.

D. Tractable MPC Formulation with GP Model

Given the approximate normal distribution of state and input, many cost functions as in (7a) can be evaluated using mean and variance information. The most prominent example for tracking tasks is a quadratic cost on states and inputs, further examples include a saturating cost [24], risk sensitive costs [25] or radial basis function networks [9]. In addition, evaluating the cost function l at the mean of state and input can be a computationally cheap approximation of (7a). We refer to an (approximate) evaluation of the expected cost as

$$\mathbb{E}\left(l_i(x_i, u_i)\right) = c_i(\mu_i^x, \mu_i^u, \Sigma_i^x) \,,$$

and formulate a resulting tractable deterministic approximation of the MPC problem (7) as

$$\begin{aligned}
\min_{\{\mu_i^u\}} \quad & c_f(\mu_N^x, \Sigma_N^x) + \sum_{i=0}^{N-1} c_i(\mu_i^x, \mu_i^u, \Sigma_i^x) && \text{(11a)}\\
\text{s.t.} \quad & \mu_{i+1}^x = f(\mu_i^x, \mu_i^u) + B_d \mu^d(\mu_i^x, \mu_i^u) && \text{(11b)}\\
& \Sigma_{i+1}^x = \tilde{A}_i \Sigma_i^x \tilde{A}_i^\mathsf{T} + B_d(\Sigma^d(\mu_i^x, \mu_i^u) + \Sigma^w) B_d^\mathsf{T} && \text{(11c)}\\
& \mu_{i+1}^x \in \mathcal{Z}(\Sigma_{i+1}^x) && \text{(11d)}\\
& \mu_i^u \in \mathcal{V}(\Sigma_i^x) && \text{(11e)}\\
& \mu_0^x = x(k), \Sigma_0^x = 0 && \text{(11f)}
\end{aligned}$$

for $i = 0, \ldots, N{-}1$. The resulting control law is obtained in a receding horizon fashion as $\kappa(x(k)) = \mu_0^{u*}$, where μ_0^{u*} is the first element of the optimal control sequence $\{\mu_0^{u*}, \ldots, \mu_N^{u*}\}$ obtained from solving (11) at state $x(k)$.

The presented formulation of the MPC problem with a GP-based model results in a non-convex optimization problem. Assuming twice differentiability of kernel and prior mean function, second-order derivative information of all quantities is, however, available. Problems of this form can typically be solved to local optima using sequential quadratic programming (SQP) or nonlinear interior-point methods [26].

In the following, we demonstrate the proposed algorithm and its properties using one simulation and one experimental example. The first is an illustrative example of an autonomous underwater vehicle (AUV) which, around a trim point, is well described by nominal linear dynamics but is subject to nonlinear friction effects at larger deviations from the trim point. The second example is a hardware implementation demonstrating the approach for autonomous racing of miniature race cars, which are described by a nonlinear nominal model. While model complexity and fast sampling times require additional approximations in this second example, we show that we can leverage key benefits of the proposed approach in this highly demanding real-world application.

IV. Online Learning for Autonomous Underwater Vehicle

We consider depth control of an AUV described by a nonlinear continuous-time model at constant surge velocity as ground truth and for simulation purposes [27]. Around a trim point of purely horizontal movement, the states and input of the system are

$[x]_1$	pitch angle relative to trim point [rad],
$[x]_2$	heave velocity relative to trim point [m/s],
$[x]_3$	pitch velocity [rad/s],
u	stern rudder deflection around trim point [rad].

We assume a given approximate linear system model that is discretized with sampling time $T_s = 100\,\text{ms}$ resulting in

$$x(k{+}1) = Ax(k) + Bu(k) + B_d\left(g(x(k)) + w(k)\right) \,,$$

which is in the form of (1). The nonlinearity results from friction effects and therefore only affects the (continuous-time) velocity states, i.e.

$$B_d = \begin{bmatrix} 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{bmatrix} \,, \; g(x(k)) = g([x(k)]_2, [x(k)]_3) : \mathbb{R}^2 \to \mathbb{R}^2 \,.$$

Note that the eigenvalues of A are given by $\{1.1, 1.03, 0.727\}$, i.e. the linear system has two unstable modes.

In the considered scenario, the goal is to track reference changes from the original zero set point to a pitch angle of 30°, back to zero and then to 45°. We furthermore consider a safety state constraint on the pitch angle of at least 10° below the reference, as well as input constraints corresponding to ±20° rudder deflection. In this example, we treat the case of *online learning*, that is we start without any data about the nonlinearity g, collect measurement data during operation and enhance performance online.

A. GP-based Reference Tracking Controller

GP data $\mathcal{D} = \{\mathbf{y}, \mathbf{z}\}$ is generated by calculating the deviation of the linear model from the measured states, as described in Section II-C, where the input data is reduced to the velocity states $z_j = [[x_j]_2, [x_j]_3]^\mathsf{T}$. Data is continuously updated during the closed-loop run. Specifically, we consider a new data point every 5 time steps, and keep track of 30 data points by discarding the oldest when adding a new point. For this conceptual example, we do not make use of the dynamic sparse approximation in Section III-B. The training data is initialized with 30 data points of zero input and zero output. We employ a squared exponential kernel (3) for both output dimensions with fixed hyperparameters $L_1 = L_2 = \text{diag}([0.35, 0.15]^\mathsf{T})$, and variances $\sigma_{f,1}^2 = 0.04$ and $\sigma_{f,2}^2 = 0.25$.

We use a quadratic stage cost, with weight matrices $Q = \text{diag}([1, 0, 10]^\mathsf{T})$, $R = 20$ and a prediction horizon $N = 35$ to track a pitch angle reference. The terminal cost P is chosen according to the solution of the associated discrete-time algebraic Riccati equation, i.e. the LQR cost. As ancillary linear controller K_i, an infinite horizon LQR control law based on the linear nominal model is designed using the same weights as in the MPC and used in all prediction steps i. This

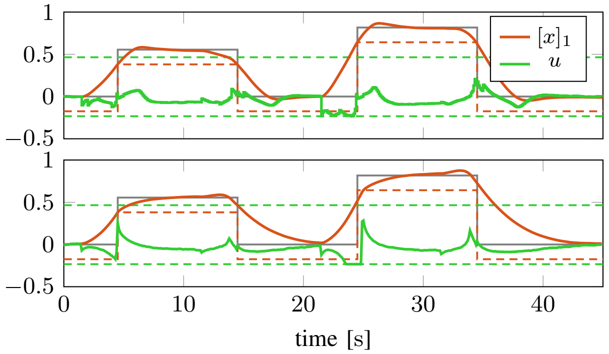


Fig. 2: Simulation results of GP-based MPC for autonomous underwater vehicle in an online learning scenario. GP-based MPC (top) and linear stochastic MPC (bottom). The solid gray line shows the reference value of the pitch angle, dashed in the respective color the state and input constraints.

stabilizes the linear system and reduces uncertainty growth over the prediction horizon. Constraints are introduced based on Remark 2 considering a maximum probability of constraint violation of 2.28%, corresponding to a 2-σ confidence bound. The MPC problem is solved using FORCES Pro [28], [29].

The resulting closed-loop trajectory, during which the system learns from online data, is shown in the top plot of Fig. 2. For comparison, we run the same simulation with a linear stochastic MPC formulation, subject to the prior model uncertainty, i.e. considering the nominal linear system subject to increased noise of variance $\Sigma^w + \text{diag}([\sigma_{f,1}^2,\ \sigma_{f,2}^2]^\mathsf{T})$ instead of the learned nonlinear GP term. We use the same cost function and chance constraint formulation in the stochastic MPC formulation, the results of which are shown in the bottom plot of Fig. 2. The simulations demonstrate improved performance of the GP-based MPC over the linear stochastic MPC, while simultaneously improving constraint satisfaction. Under the chosen uncertainty description, the constraint on minimum pitch angle is violated with the linear stochastic MPC control law, which does not account for the nonlinear friction effects during this large reference change. When pitching back down, improved performance under the GP-based MPC can be observed. In particular, compared to the linear stochastic MPC, the GP-based formulation keeps less 'safety distance' to the constraints, since the friction terms are accurately identified and residual model uncertainty is small. The GP-based formulation therefore allows for improved control performance with reduced conservatism, while maintaining safety of the AUV.

V. Autonomous Racing

As a second example we consider an autonomous racing scenario, in which the goal is to drive a car around a track as quickly as possible, while keeping the vehicle safe, i.e. while avoiding collision with the track boundaries. The controller is based on a model predictive contouring control formulation [30], [31] which has been applied to the problem of autonomous racing in [32]. Preliminary simulation results were published in [21]. This hardware experiment does not allow for direct access to state measurements and the controller and learning formulation is instead based on state estimates obtained from a localization system using an infrared camera and extended Kalman filtering. Details on the state estimation can be found in [32].

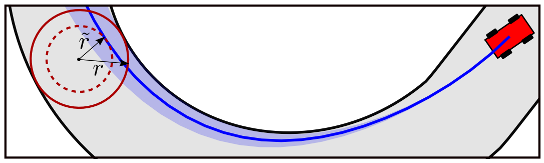


Fig. 3: Illustration of the constraint tightening procedure. The effective track radius is adjusted based on the predicted position uncertainty.

A. Car Dynamics

The race cars are modeled by continuous-time nominal dynamics $\dot{x} = f^c(x, u)$ obtained from a bicycle model with nonlinear tire forces given by a simplified Pacejka tire model. This results in states and inputs

$$x = [X, Y, \Phi, v_x, v_y, \omega]^\mathsf{T},\ \ u = [p, \delta]^\mathsf{T}\,,$$

with position $x^{XY} = [X, Y]^\mathsf{T}$, orientation Φ, longitudinal and lateral velocities v_x and v_y, and yaw rate ω. The inputs to the system are the motor duty cycle p and the steering angle δ. For details on the system modeling please refer to [21], [32].

For use in the MPC formulation, we discretize the system using a Runge-Kutta method with a sampling time of $T_s = 20\,\text{ms}$. We add $g(x, u)$ capturing unmodeled dynamics, as well as additive Gaussian noise w. Due to the structure of the nominal model, we assume that the model uncertainty, as well as the process noise w, only affect the velocity states v_x, v_y and ω of the system, and that the unmodeled dynamics do not depend on the position states, i.e.

$$B_d = [0\ I]^\mathsf{T},\ \ g(x, u) = g(v_x, v_y, \omega, p, \delta) : \mathbb{R}^5 \to \mathbb{R}^3\,.$$

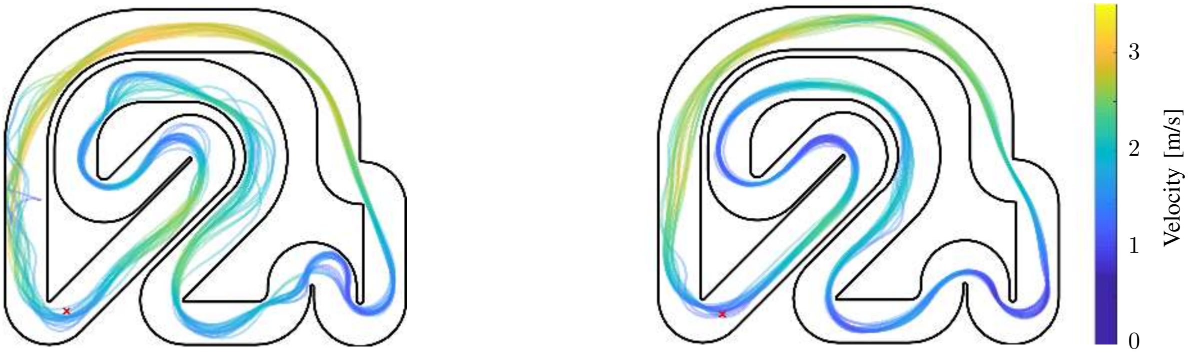
The system is subject to input constraints $\mathcal{U}$, i.e. the steering angle is limited to lie in $\pm\delta_{\max}$ and the duty cycle has to lie in $[-0.1, 1]$, where the negative values correspond to braking.

B. GP-based Racing Controller

We consider a race track given by its centerline and a fixed track width. The centerline is described by a piecewise cubic spline polynomial, which is parameterized by the path length Θ. Progress along the track is characterized by Θ_i, which is introduced as an additional state and enters the considered cost function linearly, encouraging a maximization of progress along the track. Given a Θ_i, we can evaluate the corresponding centerline position $C(\Theta) = [X_c(\Theta), Y_c(\Theta)]^\mathsf{T}$, such that the track constraint can be expressed as

$$\mathcal{X}(\Theta_i) := \{x^{XY} \mid \|x^{XY} - C(\Theta_i)\| \leq r\} \subset \mathbb{R}^2\,,$$

where r is half the track width. Making use of (10), we arrive at tightened constraints on the mean position $\mu_i^{XY} \in$



(a) Nominal controller

(b) GP-based controller

Fig. 4: Comparison of race lines with nominal and GP-based controller. The red cross is the starting point of the race car.

$\mathcal{Z}(\theta_i, \Sigma_i^{XY})$. Fig. 3 exemplifies the predicted evolution of the cars position and the resulting constraint tightening.

The state is extended by previously applied inputs and large input changes are additionally penalized. The prediction horizon is chosen as $N = 30$ and we formulate the chance constraints (10) with $\chi_2^2(p_x) = 1$. We do not make use of an ancillary controller, i.e. $K_i = 0$, and tighten constraints only for the first 20 prediction steps, applying the constraints to the mean for the remainder of the prediction horizon, similar to the method used in [33]. We reduce computation times by making use of the dynamic sparse approximations with 10 inducing points as outlined in Section III-B, placing the inducing inputs regularly along the previous solution trajectory. To allow for real-time computation at 20 ms sampling times, two additional approximations are applied. The variance dynamics are pre-evaluated based on the previous MPC solution, which enables the pre-computation of state constraints (10) and avoids optimization over the variances. Additionally, we neglect the mean prediction of the lateral velocity error since the state is difficult to estimate reliably and the error generally small, such that we observed no improvement when including it in the control formulation.

C. Results

We start out racing the car using the nominal controller, i.e. the controller without an added GP term. Since the nominal model is not well tuned, driving behavior is somewhat erratic and there are a number of small collisions with the track boundaries, as can be observed in Fig. 4a. Using the collected data, we train the GP error model d using 325 data points. We infer the hyperparameters in (3) as well as the noise level Σ^w using maximum likelihood optimization.

The resulting race lines of 20 laps with the GP-based controller are displayed in Figure 4b, generally showing a much more consistent and safe racing behavior. In particular, it can be seen that almost all of the systematic and persistent problems in the race line of the nominal controller can be alleviated. Fig. 5 shows the encountered dynamics error in the yaw-rate and the predicted error during the first lap with the sparse GP-based controller. Mean and residual uncertainty predicted by the GP matches the encountered errors well. It is important to note that the apparent volatility in the plot is not due to overfitting, but is instead due to fast changes in the input and matches the validation data, i.e. the measured errors.

To quantify performance of the proposed controllers we compare in Table I average lap time $\overline{T}_l$, minimum lap time $T_{l,\min}$ as well as the average 2-norm error in the system dynamics $\overline{\|e\|}$, i.e. the difference between the mean state after one prediction step and the realized state, $e(k{+}1) = \mu_1^x - x(k{+}1)$. We see that the GP-based controller is able to improve significantly on all these quantities, with an average lap time improvement of 0.71 s, or almost 7%, which constitutes a large improvement in the considered racing task. This is in part due to the improved system model, as evident in the average dynamics error $\overline{\|e\|}$, but also due to the cautious nature of the controller, which helps to further reduce collisions and large problems in the race line. Due to the cautious nature, the minimum lap time gains are slightly less pronounced. In fact, the nominal controller consistently displays higher top speeds. Often times, however, this leads to significant problems at the brake point to a slow corner. We evaluate the effect of the sparse dynamic approximation on the prediction quality by computing the predicted error under a *full* GP on the recorded data as $\overline{\|e_f\|} = 0.33$ and find that the increase in error when using the proposed sparse approximation is around 3%. Computation times are reported as average solve times $\overline{T}_c$ and the percentage of solutions in under 20 ms, $T_c < 20$ ms.

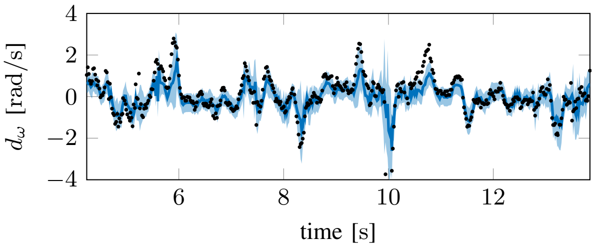


Fig. 5: Dynamic sparse GP compensation of the yaw-rate error with 10 inducing inputs during the first race lap. The black dots show the measured error on the yaw rate at each time step, while the blue line shows the error predicted by the GP. The shaded region is the 2-σ confidence interval.

TABLE I: Experimental results

Controller	$\overline{T}_l$ [s]	$T_{l,\min}$ [s]	$\overline{\|e\|}$ [-]	$\overline{T}_c$ [ms]	$T_c < 20\,\mathrm{ms}$
Nominal	10.32	9.65	0.65	18.2	76.3%
GP-based	9.61	9.27	0.34	17.2	99.8%

Average solution times of nominal and GP-based controller are similar. The percentiles of solutions in under 20 ms, however, differ significantly, likely due to frequent large re-planning with the nominal controller. The results therefore demonstrate that the presented GP-based controller can significantly improve performance while maintaining safety in a hardware implementation of a complex system with fast sampling times.

VI. Conclusion

The paper discussed the use of Gaussian processes to learn nonlinearities for a safe improvement of performance in model predictive control. Combining GP dynamics with a nominal system and the use of computational approximations rendered GP-based MPC formulation computationally feasible with low sampling times. A simulation example as well as a hardware implementation have shown cautious control with improved performance. In particular, we have demonstrated in experiments that both performance and safety in an autonomous racing setting can be significantly improved by using cautious data-driven techniques, showing the practical feasibility and potential of learning approaches in high performance control systems.

Acknowledgement

We would like to thank the Automatic Control Laboratory (IfA) at ETH Zurich and in particular Alexander Liniger for his valuable input and support with the hardware implementation on the miniature race cars.

References

[1] E. D. Klenske, M. N. Zeilinger, B. Schölkopf, and P. Hennig, "Gaussian process-based predictive control for periodic error correction," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 1, pp. 110–121, 2016.

[2] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard, "Gaussian process model based predictive control," in *American Control Conference*, Jun. 2004, pp. 2214–2219.

[3] A. Grancharova, J. Kocijan, and T. A. Johansen, "Explicit stochastic predictive control of combustion plants based on Gaussian process models," *Automatica*, vol. 44, no. 6, pp. 1621–1631, 2008.

[4] G. Cao, E. M.-k. Lai, and F. Alam, "Gaussian process model predictive control of unmanned quadrotors," in *International Conference on Control, Automation and Robotics*, 2016.

[5] X. Yang and J. M. Maciejowski, "Fault tolerant control using Gaussian processes and model predictive control," *International Journal of Applied Mathematics and Computer Science*, vol. 25, no. 1, pp. 133–148, 2015.

[6] Y. Pan, X. Yan, E. A. Theodorou, and B. Boots, "Prediction under uncertainty in sparse spectrum gaussian processes with applications to filtering and control," in *International Conference on Machine Learning*, 2017, pp. 2760–2768.

[7] 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.

[8] C. D. McKinnon and A. P. Schoellig, "Experience-based model selection to enable long-term, safe control for repetitive tasks under changing conditions," in *International Conference on Intelligent Robots and Systems*, 2018, pp. 2977–2984.

[9] S. Kamthe and M. P. Deisenroth, "Data-efficient reinforcement learning with probabilistic model predictive control," in *International Conference on Artificial Intelligence and Statistics*, 2018.

[10] L. Hewing and M. N. Zeilinger, "Stochastic model predictive control for linear systems using probabilistic reachable sets," *Conference on Decision and Control*, 2018.

[11] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. The MIT Press, 2006.

[12] M. Lázaro-Gredilla, J. Quiñonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal, "Sparse spectrum gaussian process regression," *Journal of Machine Learning Research*, vol. 11, no. Jun, pp. 1865–1881, 2010.

[13] J. Quiñonero-Candela, C. E. Rasmussen, and R. Herbrich, "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, vol. 6, pp. 1935–1959, 2005.

[14] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems*, Y. Weiss, B. Schölkopf, and J. C. Platt, Eds., 2006, pp. 1257–1264.

[15] V. Tresp, "A Bayesian committee machine," *Neural Computation*, vol. 12, no. 11, pp. 2719–2741, 2000.

[16] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, 2016.

[17] F. Oldewurtel, C. N. Jones, and M. Morari, "A tractable approximation of chance constrained stochastic mpc based on affine disturbance feedback," in *Conference on Decision and Control*, 2008, pp. 4731–4736.

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

[19] J. Quionero-Candela, A. Girard, and C. Rasmussen, "Prediction at an uncertain input for Gaussian processes and relevance vector machines - application to multiple-step ahead time-series forecasting," Max Planck Institute for Biological Cybernetics, Tübingen, Germany, Tech. Rep. IMM-2003-18, 2003.

[20] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration and reinforcement learning," *Conference on Decision and Control*, 2018.

[21] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious NMPC with Gaussian process dynamics for miniature race cars," *European Control Conference*, 2018.

[22] E. Kofman, J. A. De Don, and M. M. Seron, "Probabilistic set invariance and ultimate boundedness," *Automatica*, vol. 48, no. 10, pp. 2670–2676, 2012.

[23] L. Hewing, A. Carron, K. Wabersich, and M. N. Zeilinger, "On a correspondence between probabilistic and robust invariant sets for linear systems," *European Control Conference*, 2018.

[24] M. Deisenroth, "Efficient reinforcement learning using Gaussian processes," Ph.D. dissertation, Karlsruhe Institute for Technology, Karlsruhe, 2010.

[25] P. Whittle, "Risk-sensitive linear/quadratic/gaussian control," *Advances in Applied Probability*, vol. 13, no. 4, p. 764777, 1981.

[26] M. Diehl, H. J. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear MPC and moving horizon estimation problem formulation," *Nonlinear model predictive control*, pp. 391–417, 2009.

[27] M. S. Naik and S. N. Singh, "State-dependent Riccati equation-based robust dive plane control of AUV with control constraints," *Ocean Engineering*, vol. 34, no. 11-12, pp. 1711–1723, 2007.

[28] A. Domahidi and J. Jerez, "FORCES Professional," embotech GmbH (http://embotech.com/FORCES-Pro), 2014.

[29] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs," *International Journal of Control*, 2017.

[30] T. Faulwasser, B. Kern, and R. Findeisen, "Model predictive path-following for constrained nonlinear systems," *Conference on Decision and Control and Chinese Control Conference*, pp. 8642–8647, 2009.

[31] D. Lam, C. Manzie, and M. Good, "Model predictive contouring control," in *Conference on Decision and Control*, 2010, pp. 6137–6142.

[32] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[33] J. V. Carrau, A. Liniger, X. Zhang, and J. Lygeros, "Efficient implementation of randomized MPC for miniature race cars," *European Control Conference*, pp. 957–962, 2016.