This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/LCSYS.2020.3006279

Data Selection for Multi-Task Learning Under Dynamic Constraints

Alexandre Capone, Armin Lederer, Jonas Umlauft and Sandra Hirche

***Abstract*—Learning-based techniques are increasingly effective at controlling complex systems. However, most work done so far has focused on learning control laws for individual tasks. Simultaneously learning multiple tasks on the same system is still a largely unaddressed research question. In particular, no efficient state space exploration schemes have been designed for multi-task control settings. Using this research gap as our main motivation, we present an algorithm that approximates the smallest data set that needs to be collected in order to achieve high performance across multiple control tasks. By describing system uncertainty using a probabilistic Gaussian process model, we are able to quantify the impact of potentially collected data on each learning-based control law. We then determine the optimal measurement locations by solving a stochastic optimization problem approximately. We show that, under reasonable assumptions, the approximate solution converges towards the exact one. Additionally, we provide a numerical illustration of the proposed algorithm.**

***Index Terms*—Machine learning, information theory and control, stochastic optimal control, uncertain systems, identification**

I. Introduction

THE success of data-driven techniques in control crucially depends on the quality of the available training data set [1]–[3]. In reinforcement learning, this difficulty is tackled through task-oriented exploration, i.e., by collecting data that is particularly useful for the given task [2]. However, if the task changes, e.g., the system is required to follow a different reference trajectory, then the available data might be unsuited to learn the corresponding control policy, and a new exploration phase is necessary. This type of scenario is addressed by multi-task reinforcement learning approaches, where policies are sequentially trained for different tasks in order to achieve good overall performance [4]. However, multi-task reinforcement learning approaches often do not consider constraint requirements [5]–[9]. Furthermore, if all task-related exploration requirements are amalgamated into a single exploration phase, then the number of system interactions required to obtain good control performance across all tasks is potentially reduced. This is generally desirable, as system interactions are often considered costly [10].

System exploration is closely related to the field of optimal experimental design, where the goal is to collect data maximizing information about the underlying system [11]. Most techniques for system exploration are in this spirit, i.e., they aim to achieve a globally accurate model by steering the state to regions of high model uncertainty [12], [13]. However, this is intractable for unbounded or very large state spaces, as it implies prohibitively long exploration periods. Moreover, certain regions of the state space do not need to be explored to obtain good control performance. Some methods address these issues by striving for a locally accurate model [14], albeit without any performance guarantees after data collection.

Efficiently exploring the state space of a system to gather data for multiple tasks poses a twofold challenge. Firstly, the optimal set of hypothetical system measurements needs to be determined. Secondly, an efficient exploration trajectory needs to be computed. In this work, we address this dilemma by proposing an algorithm that provably approximates the *minimal* number of hypothetical measurement points needed to satisfy predefined constraints across different control tasks. To the best of our knowledge, this is a novel technique. We use a probabilistic Gaussian process model to quantify model uncertainty, and measure control performance by computing the probability of constraint violation. Our algorithm employs a random sampling-based approximation, which we show to be exact as the number of samples tends to infinity.

The remainder of this paper is structured as follows. After a formal problem definition in Section II, the considered Bayesian model is introduced, in Section III. Section IV presents the algorithm for approximating the optimal measurement locations, which is the main contribution of our paper. A numerical illustration, in Section V, is followed by a conclusion, in Section VI.

II. Problem Statement

We consider a stochastic nonlinear system of the form[1]

$$\begin{aligned} \boldsymbol{x}_{t+1} &= \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t) + \boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{u}_t) + \boldsymbol{w}_t \\ &:= \boldsymbol{f}(\tilde{\boldsymbol{x}}_t) + \boldsymbol{g}(\tilde{\boldsymbol{x}}_t) + \boldsymbol{w}_t, \end{aligned} \tag{1}$$

where $\boldsymbol{x}_t \in \mathbb{X} \subseteq \mathbb{R}^{d_x}$, $\mathbf{u}_t \in \mathbb{U} \subseteq \mathbb{R}^{d_u}$ are the system's states and control inputs at time step $t \in \mathbb{N}_0$, respectively. The

[1]Let $\mathbb{R}$ denote the real numbers, $\mathbb{R}_-$ the negative real numbers, $\mathbb{N}$ the strictly positive integers, and $\mathbb{N}^0 := \mathbb{N} \cup \{0\}$ the non-negative integers. For $d \in \mathbb{N}$, $\mathbb{N}_{\leq d} := \{1, \dots, d\}$ and $\mathbb{N}^0_{\leq d} := \mathbb{N}_{\leq d} \cup \{0\}$ denote all non-negative integers smaller or equal to d with and without zero, respectively. The ceiling operator is denoted by $\lceil \cdot \rceil$. Boldface lowercase/uppercase letters denote vectors/matrices. For $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times n}$, $[\boldsymbol{A}\ \boldsymbol{B}]$ denotes the horizontal concatenation of $\boldsymbol{A}$ and $\boldsymbol{B}$, $[\boldsymbol{A}]_{ij}$ denotes the entry in the i-th row and j-th column of $\boldsymbol{A}$, and $\det(\boldsymbol{A})$ its determinant. $\boldsymbol{I}_d$ denotes the d-dimensional identity matrix, $\mathrm{diag}(a_1, \dots, a_d)$ a diagonal matrix with entries $a_1, \dots, a_d$. For $\boldsymbol{v}, \boldsymbol{u} \in \mathbb{R}^d$, $\boldsymbol{u} \leq \boldsymbol{v}$ denotes componentwise inequality, and $[\boldsymbol{v}]_{1:i}$ denotes the first i entries of $\boldsymbol{v}$. For a set $\mathbb{X}$, $\mathbf{1}_{\mathbb{X}}(\cdot)$ denotes its indicator function, and $|\mathbb{X}|$ its cardinality. The uniform distribution on $\mathbb{X}$ is denoted $\mathcal{U}(\mathbb{X})$. The set of all finite subsets of $\mathbb{X}$ is denoted by $\Gamma(\mathbb{X}) := \{\{\boldsymbol{x}_i\}_{i \in \mathbb{N}_{\leq n}} | n \in \mathbb{N}^0, \boldsymbol{x}_i \in \mathbb{X}\}$.

This work was supported by the European Research Council (ERC) Consolidator Grant "Safe data-driven control for human-centric systems (CO-MAN)" under grant agreement number 864686 .

All authors are with the Department of Electrical and Computer Engineering, Technical University of Munich, Germany [alexandre.capone, armin.lederer, jonas.umlauft, hirche]@tum.de

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/LCSYS.2020.3006279

system is perturbed by normally distributed process noise $\boldsymbol{w}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q})$. The vector $\tilde{\boldsymbol{x}}_t := (\boldsymbol{x}_t, \boldsymbol{u}_t) \in \tilde{\mathbb{X}} \subseteq \mathbb{R}^{d_{\tilde{x}}}$, where $d_{\tilde{x}} := d_x + d_u$, $\tilde{\mathbb{X}} := \mathbb{X} \times \mathbb{U}$, concatenates the state $\boldsymbol{x}_t$ and the control inputs $\boldsymbol{u}_t$. For the sake of simplicity, we assume $\boldsymbol{x}_0$ is fixed and known. However, the presented method extends to the case where only the probability distribution of $\boldsymbol{x}_0$ is known. The function $\boldsymbol{f} : \tilde{\mathbb{X}} \to \mathbb{X}$ is known a priori, whereas $\boldsymbol{g} : \tilde{\mathbb{X}} \to \mathbb{X}$ is an unknown function, for which we assume to have a probabilistic model. This is discussed in Section III.

Remark 1: Assuming $\boldsymbol{f}(\cdot)$ is known is not a very restrictive requirement. If, for example, no prior system knowledge is given, then $\boldsymbol{f}(\tilde{\boldsymbol{x}}_t) = \boldsymbol{x}_t$ or $\boldsymbol{f}(\tilde{\boldsymbol{x}}_t) = \mathbf{0}$ can be employed.

We assume to have $L \in \mathbb{N}$ data-driven control laws $\boldsymbol{u}^j : \mathbb{X} \times \Gamma(\tilde{\mathbb{X}} \times \mathbb{X}) \times \mathbb{N} \to \mathbb{U}$, $j \in \mathbb{N}_{\leq L}$. Their arguments correspond to the state $\boldsymbol{x}_t$, system measurement data $\mathcal{D}_N := \{\tilde{\boldsymbol{x}}^{(i)}, \boldsymbol{f}(\tilde{\boldsymbol{x}}^{(i)}) + \boldsymbol{g}(\tilde{\boldsymbol{x}}^{(i)}) + \boldsymbol{w}^{(i)}\}_{i \in \mathbb{N}_{\leq N}}$, where $N \in \mathbb{N}$, and the time step t. The system measurements $\mathcal{D}_N$ are to be collected, e.g., via system exploration. This type of control law is frequently employed in learning-based settings [15], [16]. We assume that the control laws satisfy some regularity conditions with respect to the state $\boldsymbol{x}_t$, as detailed in the following.

Assumption 1: The control laws $\boldsymbol{u}^j(\cdot, \mathcal{D}_N, t)$ are real analytic in $\mathbb{X}$ for all $\mathcal{D}_N \in \Gamma(\tilde{\mathbb{X}} \times \mathbb{X})$ and all $t \in \mathbb{N}$.

In particular, this implies that the control laws $\boldsymbol{u}^j(\cdot, \cdot, \cdot)$ are smooth with respect to the state. This applies for many commonly used control laws, e.g., PID-controllers and neural networks with smooth activation functions.

Each control law $\boldsymbol{u}^j(\cdot, \cdot, \cdot)$ is required to fulfill a different task, which is expressed as a series of constraints

$$\boldsymbol{h}_t^j\left(\tilde{\boldsymbol{x}}_t^j\right) \leq \mathbf{0}, \quad \forall\, t \in \mathbb{N}_{\leq H}^0,\ j \in \mathbb{N}_{\leq L} \tag{2}$$

over a finite time horizon of H steps. Here $\boldsymbol{h}_t^j : \tilde{\mathbb{X}} \to \mathbb{R}^S$ are nonlinear constraint functions, $S \in \mathbb{N}$ denotes the number of constraints per control law, and $\tilde{\boldsymbol{x}}_t^j := (\boldsymbol{x}_t, \boldsymbol{u}^j(\boldsymbol{x}_t, \mathcal{D}_N))$. Such constraints are often linear, e.g., in the case of energy or input saturation constraints, or polynomial, e.g., in the case of tracking error performance requirements. In this work, we consider the following, more general case:

Assumption 2: The entries $[\boldsymbol{h}_t^j(\cdot)]_i$ of the functions $\boldsymbol{h}_t^j(\cdot)$ are non-constant and real analytic.

Note that Assumption 2 accommodates regions with non-smooth boundaries, e.g., the intersection of linear constraints.

Remark 2: The proposed method extends straightforwardly to the more general case where both the horizon H and number of constraints S are different for each control law. However, we do not consider this case for notational convenience.

We aim to obtain the smallest possible set of measurement locations $\tilde{\mathcal{X}}^* := \{\tilde{\boldsymbol{x}}^{(i),*}\}_{i=1,\dots,N^*}$, such that the corresponding data set $\mathcal{D}^*$, if collected and used to design the control laws $\boldsymbol{u}^j(\cdot, \cdot, \cdot)$, yields system trajectories that satisfy (2) with high probability. This is expressed by the chance-constrained problem

$$\begin{aligned} \tilde{\mathcal{X}}^* = \arg\min_{\tilde{\mathcal{X}}_N \in \Gamma(\tilde{\mathbb{X}})} & N \\ \text{s.t.} \quad \mathcal{D}_N = & \left\{\tilde{\boldsymbol{x}}^{(i)}, \boldsymbol{f}\left(\tilde{\boldsymbol{x}}^{(i)}\right) + \boldsymbol{g}\left(\tilde{\boldsymbol{x}}^{(i)}\right) + \boldsymbol{w}^{(i)}\right\}_{\tilde{\boldsymbol{x}}^{(i)} \in \tilde{\mathcal{X}}_N} \\ & C_N\left(\tilde{\mathcal{X}}_N\right) > 1 - \delta, \end{aligned} \tag{3}$$

where

$$C_N\left(\tilde{\mathcal{X}}_N\right) := \mathrm{P}\left(\boldsymbol{h}_t^j\left(\tilde{\boldsymbol{x}}_t^j\right) \leq \mathbf{0}, \forall\, t \in \mathbb{N}_{\leq H}^0,\ j \in \mathbb{N}_{\leq L}\right), \tag{4}$$

is the probability of constraint satisfaction given $\tilde{\mathcal{X}}_N$, which we require to be bounded from below by $1-\delta$. Here $\delta \in (0, 1)$ is a design parameter that specifies the desired probability of constraint violation. Here $\tilde{\mathcal{X}}_N := \{\tilde{\boldsymbol{x}}^{(i)}\}_{i \in \mathbb{N}_{\leq N}}$ denotes the locations of N system measurements. The probability operator $\mathrm{P}(\cdot)$ describes the probability of an event given process noise $\boldsymbol{w}_t$ and the a priori distribution that we assume for the unknown function $\boldsymbol{g}(\cdot)$, as discussed in Section III.

Remark 3: Since the system dynamics are unknown, the measurements in an arbitrary data set $\mathcal{D}_N$ are hypothetical. However, by assuming a distribution over $\boldsymbol{g}(\cdot)$, we are able to determine the impact of measurement locations $\tilde{\mathcal{X}}_N$ on control performance.

Finding an optimal set $\tilde{\mathcal{X}}^*$ under uncertainty is generally impossible without considering further assumptions. Hence, we restrict ourselves to the case where the controllers are specified in a way that the desired closed-loop behavior is achievable:

Assumption 3: The optimization problem (3) is feasible for a finite $\tilde{\mathcal{X}}^*$, i.e., $|\tilde{\mathcal{X}}^*| =: N^* < \infty$.

Furthermore, we assume that the optimal data set is contained within a known compact subset of $\mathbb{X}$:

Assumption 4: There exists a known compact subset $\tilde{\mathbb{X}}^* \subset \tilde{\mathbb{X}}$, such that $\tilde{\boldsymbol{x}}^{(i),*} \in \tilde{\mathbb{X}}^*$ for all $i \in \{1, \dots, N^*\}$.

This does not constitute a very restrictive assumption, since control tasks are typically restricted to a compact subset of the state space, i.e., we only require information from a compact subset to achieve good performance.

III. Probabilistic Model

In order to assess how data collected in the future will potentially affect control performance, we need to quantify how model uncertainty decreases after new data points have been added. To this end, we consider *hypothetical* data points $\tilde{\boldsymbol{s}}^{(1)}, \dots, \tilde{\boldsymbol{s}}^{(N)} \in \tilde{\mathbb{X}}$ and system trajectories $\tilde{\boldsymbol{s}}_0^j, \dots, \tilde{\boldsymbol{s}}_H^j \in \tilde{\mathbb{X}}$, which are drawn from a GP distribution, as explained in the sequel. For simplicity of exposition, we define a single set that subsumes both hypothetical data and sampled system trajectories as

$$\mathcal{S}_{\tilde{N}} := \{\tilde{\boldsymbol{s}}_n, \boldsymbol{f}(\tilde{\boldsymbol{s}}_n) + \boldsymbol{g}^s(\tilde{\boldsymbol{s}}_n) + \boldsymbol{w}_n\}_{n \in \mathbb{N}_{\leq \tilde{N}-1}^0} \tag{5}$$

where $\tilde{N} := N + L(H+1)$. Here we use the superscript s to emphasize that $\boldsymbol{g}^s(\cdot)$ is a *sample* function evaluation, as opposed to an evaluation of the true function $\boldsymbol{g}(\cdot)$. The first N elements of $\mathcal{S}_{\tilde{N}}$ correspond to a hypothetical data set, and the remaining elements correspond to sample trajectories, i.e.,

$$\tilde{\boldsymbol{s}}_n = \begin{cases} \tilde{\boldsymbol{s}}^{(d_n)}, & n = 0 \dots, N-1 \\ \tilde{\boldsymbol{s}}_{t_n}^{j_n} & n = N, \dots, \tilde{N}-1 \end{cases} \tag{6}$$

where we reorganize indices as $j_n := \lceil (n - N + 1)/(H+1) \rceil$, $d_n := n + 1$, and $t_n := n - N - (j_n - 1)(H+1)$.

Remark 4: A GP model can be trained using measurement data from the true system (1). For the sake of notational

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/LCSYS.2020.3006279

simplicity, we analyze the setting where no prior measurement data from the *true system* is available, and show exclusively how to draw samples from a GP in a recursive fashion. However, this does not constitute a loss of generality, since the a posterior GP distribution after training satisfies the requirements used in this paper [17].

We begin by introducing GPs for the case where $d_x = 1$, and then describe how they can be generalized to a multivariate setting. Formally, a GP is a collection of random variables, of which any finite subset is jointly normally distributed [17]. It is fully specified by a mean function, which we set to zero without loss of generality [17], and a positive definite kernel $k : \mathbb{R}^{d_{\tilde{x}}} \times \mathbb{R}^{d_{\tilde{x}}} \to \mathbb{R}$. Given a sample data set $\mathcal{S}_n$, a subsequent sample evaluation at an arbitrary augmented state $\tilde{\boldsymbol{x}}$ is normally distributed, i.e., $g^s(\tilde{\boldsymbol{x}}) \sim \mathcal{N}(\mu_{n+1}(\tilde{\boldsymbol{x}}), \sigma^2_{n+1}(\tilde{\boldsymbol{x}}))$, with mean and variance

$$\mu_{n+1}(\tilde{\boldsymbol{x}}) := \mu(\tilde{\boldsymbol{x}}|\mathcal{S}_n) = \boldsymbol{k}_n^{\mathrm{T}}(\tilde{\boldsymbol{x}})\, \boldsymbol{K}_n^{-1} \boldsymbol{y}_n, \tag{7}$$

$$\sigma^2_{n+1}(\tilde{\boldsymbol{x}}) := \sigma^2(\tilde{\boldsymbol{x}}|\mathcal{S}_n) = k(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{x}}) - \boldsymbol{k}_n^{\mathrm{T}}(\tilde{\boldsymbol{x}}) \boldsymbol{K}_n^{-1} \boldsymbol{k}_n(\tilde{\boldsymbol{x}}), \tag{8}$$

where $[\boldsymbol{k}_n(\tilde{\boldsymbol{x}})]_i = k(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{s}}_i)$, $[\boldsymbol{y}_n]_i = g^s(\tilde{\boldsymbol{s}}_i)$, and the entries of the covariance matrix are given by $[\boldsymbol{K}_n]_{ij} = k(\tilde{\boldsymbol{s}}_i, \tilde{\boldsymbol{s}}_j)$.

Using (7) and (8), we are able to sample data sets and system trajectories from the GP distribution as $g^s(\tilde{\boldsymbol{x}}) := \mu_{n+1}(\tilde{\boldsymbol{x}}) + \sigma_{n+1}(\tilde{\boldsymbol{x}})\, \zeta$, where $\zeta \sim \mathcal{N}(0, 1)$. If $d_x > 1$, we model each dimension with a separate GP, i.e., $\boldsymbol{g}^s(\tilde{\boldsymbol{x}}) \sim \mathcal{N}\left(\boldsymbol{\mu}_n(\tilde{\boldsymbol{x}}), \boldsymbol{\Sigma}_n^2(\tilde{\boldsymbol{x}})\right)$, where $[\boldsymbol{\mu}_n(\tilde{\boldsymbol{x}})]_d = \mu(\tilde{\boldsymbol{x}}|\mathcal{S}_{d,n})$, $\boldsymbol{\Sigma}_n(\tilde{\boldsymbol{x}}) = \operatorname{diag}\Big(\sigma(\tilde{\boldsymbol{x}}|\mathcal{S}_{1,n}), \ldots, \sigma(\tilde{\boldsymbol{x}}|\mathcal{S}_{d_x,n})\Big)$, and the measurement data and samples are separated for each dimension $d \in \{1, \ldots, d_x\}$ as $\mathcal{S}_{d,n} = \{\tilde{\boldsymbol{s}}_i, [\boldsymbol{f}(\tilde{\boldsymbol{s}}_i)]_d + [\boldsymbol{g}^s(\tilde{\boldsymbol{s}}_i)]_d + [\boldsymbol{w}_i]_d\}_{i=\mathbb{N}^0_{i-1}}$. This corresponds to conditionally independent state transition function entries, which is a common assumption for multivariate systems [2].

We assume the GP kernel $k(\cdot, \cdot)$ correctly captures the prior knowledge about the entries of $\boldsymbol{g}(\cdot)$:

Assumption 5: The entries of $\boldsymbol{g}(\cdot)$ are sampled from a GP with mean zero and known real analytic kernel $k(\cdot, \cdot)$.

Assumption 5 effectively assumes a probability distribution over $\boldsymbol{g}(\cdot)$, specified by the choice of kernel $k(\cdot, \cdot)$. As $k(\cdot, \cdot)$ only encodes high-level properties of $\boldsymbol{g}(\cdot)$, the resulting function space is generally far richer than in the case of a fixed model structure with unknown parameters, which is habitual in system identification [18]. Assumption 5 is frequently used in practice, e.g., for robotic systems [2]. In particular, it implies that the expected value of an arbitrary state $\tilde{\boldsymbol{x}}_t^j$ at time t under control law j is obtained by integrating over $\boldsymbol{g}^s(\tilde{\boldsymbol{x}}) \sim \mathcal{N}\left(\boldsymbol{\mu}_n(\tilde{\boldsymbol{x}}), \boldsymbol{\Sigma}_n^2(\tilde{\boldsymbol{x}})\right)$, i.e.,

$$\mathrm{E}_{\boldsymbol{g},\boldsymbol{w}}\left(\boldsymbol{x}_t^j\right) = \int_{\mathbb{X}^{2\tilde{N}}} \boldsymbol{s}_{n_{j,t}} \prod_{i=0}^{\tilde{N}-1} \mathrm{p}(\boldsymbol{\zeta}_i) d\boldsymbol{\zeta}_i, \tag{9}$$

where $n_{j,t} := N + (j-1)(H+1) + t$, and $\mathrm{E}_{\boldsymbol{g},\boldsymbol{w}}(\cdot)$ denotes the expected value with respect to the unknown function $\boldsymbol{g}(\cdot)$ and the process noise $\boldsymbol{w}_t$. The samples are computed recursively using

$$\begin{aligned}
\boldsymbol{s}_{n+1} &= \boldsymbol{f}(\tilde{\boldsymbol{s}}_n) + \boldsymbol{\mu}_n(\tilde{\boldsymbol{s}}_n) + [\boldsymbol{\Sigma}_n(\tilde{\boldsymbol{s}}_n)\ \boldsymbol{Q}]\, \boldsymbol{\zeta}_n, \quad n+1 \neq n_{j,0}, \\
\boldsymbol{s}_{n_{j,0}} &= \boldsymbol{x}_0, \qquad \tilde{\boldsymbol{s}}_n = \left(\boldsymbol{s}_n, \boldsymbol{u}^j(\boldsymbol{s}_n, \mathcal{S}_N, t)\right), \qquad \forall\ j \in \mathbb{N}_{\leq L} \\
\mathcal{S}_i &= \left\{\tilde{\boldsymbol{s}}_n, \boldsymbol{f}(\tilde{\boldsymbol{s}}_n) + \boldsymbol{\mu}_n(\tilde{\boldsymbol{s}}_n) + [\boldsymbol{\Sigma}_n(\tilde{\boldsymbol{s}}_n) \quad \boldsymbol{Q}]\, \boldsymbol{\zeta}_n\right\}_{n \in \mathbb{N}^0_{\leq i-1}}
\end{aligned}$$

Here $\mathrm{p}(\boldsymbol{\zeta}_n) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{2d_x})$. Note that we require the random variables $\boldsymbol{\zeta}_i$ to have dimension $2d_x$ in order for the GP samples $\boldsymbol{g}^s(\tilde{\boldsymbol{s}}_n) = \boldsymbol{\mu}_i(\tilde{\boldsymbol{s}}_n) + \boldsymbol{\Sigma}_i(\tilde{\boldsymbol{s}}_n)\, [\boldsymbol{\zeta}_n]_{1:d_x}$ to be uniquely defined [17].

Since each control law $\boldsymbol{u}^j(\cdot, \cdot, \cdot)$ is independent given the training data $\mathcal{D}_N$, the probability of constraint satisfaction for a set of measurement points $\tilde{\mathcal{X}}_N$ is given by

$$C_N\left(\tilde{\mathcal{X}}_N\right) = \prod_{n=0}^{\tilde{N}-1} \int_{\mathbb{X}^2} \mathbf{1}_{\mathbb{R}^{d_{\tilde{x}}}_-}\left(\boldsymbol{h}^{j_n}_{t_n}(\tilde{\boldsymbol{s}}_n)\right) p\left(\tilde{\boldsymbol{s}}_n \middle| \mathcal{S}_n\right) d\boldsymbol{\zeta}_n. \tag{10}$$

Here we set $\mathbf{1}_{\mathbb{R}^{d_{\tilde{x}}}_-}(\boldsymbol{h}^{j_n}_{t_n}(\tilde{\boldsymbol{s}}_n)) := 1$ for all $n \leq N-1$ for simplicity of exposition.

As our goal is to find the smallest possible set of measurement points $\tilde{\mathcal{X}}^*$, it is reasonable to assume that $\tilde{\mathcal{X}}^*$ does not contain any measurement locations that provide identical information. In terms of a GP distribution, this is expressed as follows.

Assumption 6: Let $\tilde{\mathcal{X}}^*$ be the minimizer of (3). Then $\boldsymbol{\Sigma}_n(\tilde{\boldsymbol{x}}^{(n+1),*})$ is invertible for $n \in \mathbb{N}_{\leq N^*-1}$.

Intuitively, Assumption 6 states that the control performance does not benefit from performing measurements at the same location multiple times. This is the case, e.g., if process noise $\boldsymbol{w}_t$ is small compared to $\boldsymbol{g}(\cdot)$. For nondegenerate kernels, e.g., the squared-exponential kernel, Assumption 6 implies that the elements of $\tilde{\mathcal{X}}^*$ are all different.

IV. Two Stage Optimization

A. Sampling-Based Approximation

Solving the optimization problem (3) exactly is generally impossible, since the corresponding chance constraints (4) are intractable to compute. Hence, we employ a two-stage approach to approximate a minimizer of (3), which is detailed in Algorithm 1. First, we fix the size N of the data $\tilde{\mathcal{X}}_N$, then minimize a sample average approximation [19] of the chance constraints (4), given by

$$C_N(\tilde{\mathcal{X}}_N) \approx C_N^M(\tilde{\mathcal{X}}_N^M) := \frac{1}{M} \sum_{m=1}^{M} \prod_{n=0}^{\tilde{N}-1} \mathbf{1}_{\mathbb{R}^{d_{\tilde{x}}}_-}\left(\boldsymbol{h}^{j_n}_{t_n}(\tilde{\boldsymbol{s}}_n^m)\right). \tag{11}$$

Here $\tilde{\boldsymbol{s}}_n^m$ denotes the m-th sample corresponding to the n-th entry in $\mathcal{S}_N$, and $M \in \mathbb{N}$ is the total number of sample sets $\mathcal{S}_N^m$. If the maximal approximate probability of constraint satisfaction is lower than the desired bound $1-\delta$, the number of data points N is increased and the procedure is repeated.

Remark 5: The main driver of computational complexity in Algorithm 1 is the inversion of the matrices $\boldsymbol{K}_n$, required for the GP mean and variance. In practice, the size of $\boldsymbol{K}_n$ can be reduced, e.g., by employing sparse GP methods [20].

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/LCSYS.2020.3006279

Algorithm 1 Data selection for multi-task learning

Input: M, δ, $\boldsymbol{f}(\cdot)$, $\boldsymbol{Q}$, $\boldsymbol{\zeta}_1^1, \boldsymbol{\zeta}_1^2, \boldsymbol{\zeta}_2^2, \dots$

Set $N = 0$

while $C_N^M(\tilde{\mathcal{X}}_N^M) \leq 1 - \delta$ **do**

Set $N \leftarrow N + 1$

Solve

$$\tilde{\mathcal{X}}_N^M = \arg\max_{\tilde{\mathcal{X}}_N} C_N^M(\tilde{\mathcal{X}}_N)$$

$$\text{s.t. } \forall\, m \in \mathbb{N}_{\leq M},\ j \in \mathbb{N}_{\leq L},\ n \in \mathbb{N}_{\leq \tilde{N}-1}^0,\ n+1 \neq n_{j,0}$$

$$\boldsymbol{s}_{n+1}^m = \boldsymbol{f}(\tilde{\boldsymbol{s}}_n^m) + \boldsymbol{\mu}_n^m(\tilde{\boldsymbol{s}}_n^m) + [\boldsymbol{\Sigma}_n^m(\tilde{\boldsymbol{s}}_n^m) \quad \boldsymbol{Q}]\, \boldsymbol{\zeta}_n^m,$$

$$\boldsymbol{s}_{n_{j,0}}^m = \boldsymbol{x}_0, \qquad \tilde{\boldsymbol{s}}_n^m = \left(\boldsymbol{s}_n^m, \boldsymbol{u}^j(\tilde{\boldsymbol{s}}_n^m, \mathcal{S}_N^m, t_n)\right)$$

$$\mathcal{S}_n^m = \Big\{ \tilde{\boldsymbol{s}}_i^m, \boldsymbol{f}(\tilde{\boldsymbol{s}}_i^m) + \boldsymbol{\mu}_i^m(\tilde{\boldsymbol{s}}_i^m) + [\boldsymbol{\Sigma}_i^m(\tilde{\boldsymbol{s}}_i^m)\ \boldsymbol{Q}]\, \boldsymbol{\zeta}_i^m \Big\}_{i \in \mathbb{N}_{\leq n}}$$

end while

Set $\tilde{\mathcal{X}}_N^{M,*} = \tilde{\mathcal{X}}_N^M$

return $\tilde{\mathcal{X}}_N^{M,*}$

B. Theoretical Analysis

We now derive formal guarantees for the approximate solution $\tilde{\mathcal{X}}_N^M$ obtained with Algorithm 1. To this end, we prove some preliminary results.

Lemma 1: Let $d_x = 1$, let Assumption 6 hold and let $\mathcal{S}_n$ be a sample data set. Furthermore, let $\sigma_n^2(\cdot)$ be the corresponding posterior covariance and let $\boldsymbol{u}_t^j(\cdot)$ be a control law that satisfies Assumption 1. Then $\sigma_n^2\left(\boldsymbol{x}, \boldsymbol{u}^j(\boldsymbol{x})\right) \neq 0$ holds for all $\boldsymbol{x} \in \mathbb{X}$ up to a set of measure zero.

Proof: Non-zero real analytic functions are non-zero almost everywhere, and the concatenation of real analytic functions is also real analytic [21]. Since $\boldsymbol{u}^j(\cdot, \mathcal{D}_N, t)$ is real analytic, and $\sigma_n^2(\cdot)$ corresponds to a sum of kernel evaluations, $\sigma_n^2\left(\boldsymbol{x}, \boldsymbol{u}^j(\boldsymbol{x})\right)$ is a real analytic function of $\boldsymbol{x}$. □

Remark 6: Lemma 1 implies $\det(\boldsymbol{\Sigma}_n(\tilde{\boldsymbol{x}}^{(n+1)})) \neq 0$ for almost every $\tilde{\mathcal{X}}_N$, hence we are able to search for a minimizer of (3) using gradient-descent-based approaches. This is illustrated in Section V.

This enables us to show that the state is on a predefined set of measure zero with probability zero.

Lemma 2: Let Assumptions 2, 5 and 6 hold, and let $\mathbb{X}_0 \subset \mathbb{X}$ be an arbitrary subset of the state space with measure zero. Then $\mathrm{P}(\boldsymbol{x}_t^j \in \mathbb{X}_0) = 0$ holds for all $j \in \mathbb{N}_{\leq L}$ and $t \in \mathbb{N}_{\leq H}^0$.

Proof: Assume, without loss of generality, that $j = 1$. We first prove the result for $N = 0$, and then discuss how it extends to an arbitrary $N \in \mathbb{N}$. Since $N = 0$ and $j = 1$, the probability that the state lies within an arbitrary set of measure zero at time step t is given by

$$\mathrm{P}\left(\boldsymbol{x}_t^1 \in \mathbb{X}_0\right) = \int_{\mathbb{X}^{2t}} \mathbf{1}_{\mathbb{X}_0}\Big(\boldsymbol{f}(\tilde{\boldsymbol{s}}_{t-1}) + \boldsymbol{\mu}_{t-1}(\tilde{\boldsymbol{s}}_{t-1}) + [\boldsymbol{\Sigma}_{t-1}(\tilde{\boldsymbol{s}}_{t-1}) \quad \boldsymbol{Q}]\, \boldsymbol{\zeta}_{t-1}\Big) \prod_{i=0}^{t-1} \mathrm{p}(\boldsymbol{\zeta}_i) d\boldsymbol{\zeta}_i \tag{12}$$

As $\boldsymbol{f}(\tilde{\boldsymbol{s}}_{t-1})$ and $\boldsymbol{\mu}_{t-1}(\tilde{\boldsymbol{s}}_{t-1})$ are constant with respect to $\boldsymbol{\zeta}_{t-1}$, and the measure of $\mathbb{X}_0$ is translation-invariant, it suffices to show

$$\int_{\mathbb{X}^2} \mathbf{1}_{\mathbb{X}_0}\Big([\boldsymbol{\Sigma}_{t-1}(\tilde{\boldsymbol{s}}_{t-1}) \quad \boldsymbol{Q}]\, \boldsymbol{\zeta}_{t-1}\Big) \mathrm{p}(\boldsymbol{\zeta}_{t-1}) d\boldsymbol{\zeta}_{t-1} \stackrel{!}{=} 0$$

for all $t \in \mathbb{N}_{\leq H}^0$, which we achieve by induction. For $t = 1$,

$$\int_{\mathbb{X}^2} \mathbf{1}_{\mathbb{X}_0}\left(\left[\boldsymbol{\Sigma}_0(\tilde{\boldsymbol{s}}_0^j) \quad \boldsymbol{Q}\right] \boldsymbol{\zeta}_0\right) \mathrm{p}(\boldsymbol{\zeta}_0) d\boldsymbol{\zeta}_0$$

$$= \int_{\mathbb{X}} \left(\int_{\mathbb{X}} \mathbf{1}_{\mathbb{X}_0}(\boldsymbol{x})\, \mathrm{p}\left(\boldsymbol{\Sigma}_0^{-1}(\tilde{\boldsymbol{s}}_0)\boldsymbol{x} - \boldsymbol{\zeta}_0''\right) \boldsymbol{\Sigma}_0^{-1}(\tilde{\boldsymbol{s}}_0^j) d\boldsymbol{x} \right) \mathrm{p}(\boldsymbol{\zeta}_0'') d\boldsymbol{\zeta}_0''$$

$$= 0,$$

holds, since $\mathbf{1}_{\mathbb{X}_0}(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathbb{X}$ up to a set of measure zero. Here we employ the fact that that $\boldsymbol{\Sigma}_0(\tilde{\boldsymbol{s}}_0) = \mathrm{diag}(k(\tilde{\boldsymbol{s}}_0, \tilde{\boldsymbol{s}}_0), \dots, k(\tilde{\boldsymbol{s}}_0, \tilde{\boldsymbol{s}}_0))$ is invertible for all non-zero kernels, which allows us to integrate using the substitution $\boldsymbol{x} = \boldsymbol{\Sigma}_0(\tilde{\boldsymbol{s}}_0^j)\boldsymbol{\zeta}_0' + \boldsymbol{Q}\boldsymbol{\zeta}_0''$ and $\boldsymbol{\zeta}_0' \coloneqq [\boldsymbol{\zeta}_i]_{1:d_x}$, $\boldsymbol{\zeta}_0'' \coloneqq [\boldsymbol{\zeta}_i]_{d_x+1:2d_x}$. The expression $\mathrm{p}(\boldsymbol{\Sigma}_0^{-1}(\tilde{\boldsymbol{s}}_0^j)\boldsymbol{x} - \boldsymbol{\zeta}_0'')$ corresponds to a normal distribution with center $\boldsymbol{\zeta}_0''$ and scaling matrix $\boldsymbol{\Sigma}_0(\tilde{\boldsymbol{s}}_0^j))^{-1}$, hence it is smooth and integrable with respect to $\boldsymbol{x}$. Consequently, the result holds for $t = 1$. Note that, due to Lemma 1, this implies that $\boldsymbol{\Sigma}_1(\tilde{\boldsymbol{s}}_1)$ is invertible for almost every $\boldsymbol{\zeta}_0$. Hence, we can assume that $\boldsymbol{\Sigma}_{t-1}(\tilde{\boldsymbol{s}}_{t-1})$ is invertible for a fixed $t-1$ and almost every $\tilde{\boldsymbol{s}}_{t-1}$, and we can apply the same argument as in the case $t = 1$ to obtain the desired result for an arbitrary t and $j = 1$. Due to Assumption 6, the matrix $\boldsymbol{\Sigma}_N(\cdot)$ is invertible for data sets of size $N \neq 0$, which enables us to extend the proof to arbitrary N using the same argument. □

This directly yields the following result:

Lemma 3: Let Assumptions 2, 5 and 6 be satisfied. Then $\mathrm{P}(\boldsymbol{h}_t^j(\tilde{\boldsymbol{x}}_t^j) = \mathbf{0}) = 0$ holds for all $t \in \mathbb{N}_{\leq H}^0$, $j \in \mathbb{N}_{\leq L}$.

Proof: Since $[\boldsymbol{h}_t^j(\tilde{\boldsymbol{x}})]_i$ are real analytic and non-constant by assumption, $[\boldsymbol{h}_t^j(\tilde{\boldsymbol{x}})]_i \neq 0$ holds for all $i \in \mathbb{N}_{\leq S}$ and all $\tilde{\boldsymbol{x}} \in \tilde{\mathbb{X}}$ up to a set of measure zero. By employing Lemma 2 and the union bound, we obtain

$$\mathrm{P}\left(\boldsymbol{h}_t^j(\tilde{\boldsymbol{x}}_t^j) = \mathbf{0}\right) \leq \bigcup_{i \in \mathbb{N}_{\leq S}} \mathrm{P}\left(\left[\boldsymbol{h}_t^j(\tilde{\boldsymbol{x}}_t^j)\right]_i = 0\right) = 0. \quad \square$$

We now show that the approximations used in Algorithm 1 converge to the true probabilities of constraint satisfaction (10).

Lemma 4: Let Assumptions 1–6 hold, and let $\tilde{\mathbb{X}}^*$ be given as in Assumption 4. Then, for an arbitrary $N \in \mathbb{N}$, the expected value of $C_N(\cdot)$ is finite valued and continuously differentiable on $(\tilde{\mathbb{X}}^*)^N$, and $C_N^M(\cdot)$ converges to $C_N(\cdot)$ with probability 1 uniformly in $(\tilde{\mathbb{X}}^*)^N$ as $M \to \infty$.

Remark 7: The proofs of Lemma 4 and Theorem 1, which we state in the following, require Theorem 7.48 and Theorem 5.4 from [19], respectively. Due to space limitations, we do not include them here. However, to facilitate interpretation, we enumerate the technical statements in the proofs of Lemma 4 and Theorem 1, such that they correspond to Theorem 7.48 and Theorem 5.4 from [19].

Proof of Lemma 4: We show that $C_N^M(\cdot)$ satisfies all conditions of [19, Theorem 5.4], enumerated in the sequel as i)-iii), which directly yields the desired result.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/LCSYS.2020.3006279

i) We employ an argument from [19]. Due to Lemma 3, the functions $\mathbf{1}_{\mathbb{R}^S_-}(\boldsymbol{h}_t^j(\tilde{\boldsymbol{s}}^m_{n_{j,t}}))$ are uniquely defined and continuous for an arbitrary $t, j \in \mathbb{N}$ and almost every sample $\boldsymbol{\zeta}_n^m$. Hence, $C_N^M(\tilde{\mathcal{X}}_N)$ is continuously differentiable at any $\tilde{\mathcal{X}}_N \in (\tilde{\mathbb{X}}^*)^N$ for almost every sample $\boldsymbol{\zeta}_n^m$.
ii) Since $C_N^M(\tilde{\mathcal{X}}_N) \leq 1$ and $\tilde{\mathcal{X}}_N \in (\tilde{\mathbb{X}}^*)^N$ is compact, the absolute value of $C_N^M(\tilde{\mathcal{X}}_N)$ is upper bounded by an integrable function on $\tilde{\mathcal{X}}_N \in (\tilde{\mathbb{X}}^*)^N$.
iii) The samples $\boldsymbol{\zeta}_n^m$ are i.i.d. □

Lemma 5: Let Assumptions 1–6 hold, and let $C_N(\cdot)$ be the probability of constraint satisfaction for a data set of size N. Let $C_N^M(\cdot)$ correspond to its sample average approximation, and let $\tilde{\mathcal{X}}_N^{M,*}$ denote the output of Algorithm 1. Then, with probability 1, for every $\varepsilon \geq 0$, there exists an M_ε, such that $C_N(\tilde{\mathcal{X}}_N^{M,*}) - C_N^* \leq \varepsilon$ holds for all $M \geq M_\varepsilon$.

Proof: We show that the conditions of [19, Theorem 5.4] are satisfied by $C_N(\cdot)$ and $C_N^M(\cdot)$, which yields the desired result. In the following, we employ i)-iv) to enumerate the required conditions, which corresponds to the enumeration in [19, Theorem 5.4].

i) Due to Assumption 4, $(\tilde{\mathbb{X}}^*)^N$ is non-empty and compact.
ii) Due to Lemma 4, $C_N(\cdot)$ is finite valued and continuously differentiable on $(\tilde{\mathbb{X}}^*)^N$.
iii) Due to Lemma 4, $C_M(\cdot)$ converges to $C_N(\cdot)$ with probability 1 as $M \to \infty$, uniformly in $(\tilde{\mathbb{X}}^*)^N$.
iv) Since we restrict ourselves to the set $(\tilde{\mathbb{X}}^*)^N$, $\tilde{\mathcal{X}}_N^{M,*} \in (\tilde{\mathbb{X}}^*)^N$ holds trivially for all M. □

We now state the main result of this paper, namely that Algorithm 1 is able to approximate an optimal solution arbitrarily accurately with probability 1 using a high enough but finite number of random samples M.

Theorem 1: Let Assumptions 1–6 hold, and let $\tilde{\mathcal{X}}_N^{M,*}$ denote the output of Algorithm 1. Then, with probability 1, for every $\varepsilon > 0$, there exists an M_ε, such that $C_N(\tilde{\mathcal{X}}_N^{M,*}) - C^* \leq \varepsilon$ holds for all $M \geq M_\varepsilon$.

Proof: The result holds if the approximate optima $C_N^M(\tilde{\mathcal{X}}_N^M), \quad N = 1, \ldots, N^*$, obtained in Line 4 of Algorithm 1 converge uniformly to the true solutions $C_N(\tilde{\mathcal{X}}_N^M)$. Due to Lemma 4, the conditions required by Lemma 5 hold for every fixed N. Furthermore, since the inequality $C_{N^*}^* < 1 - \delta$ holds strictly, Algorithm 1 returns a solution of size at most N^* with probability 1 for M large enough. As the samples drawn for each problem are i.i.d., we have

$$\mathrm{P}\Big(\lim_{M\to\infty} C_N^{M,*} = C^*, \lim_{M\to\infty} |\tilde{\mathcal{X}}_N^M| = N^*, \forall\ N \in \mathbb{N}_{\leq N^*}\Big)$$
$$= \prod_{N=1}^{N^*} \mathrm{P}\left(\lim_{M\to\infty} C_N^M(\tilde{\mathcal{X}}_N^{M,*}) = C^*, \lim_{M\to\infty} |\tilde{\mathcal{X}}_N^M| = N^*\right) = 1.$$

□

In particular, Theorem 1 implies that, for M large enough, the difference between $C_N^{M,*}$ and the exact optimal probability of constraint satisfaction C^* can be made arbitrarily small.

V. Numerical Illustration

We illustrate the proposed approach with a system of the form given by (1), where $\boldsymbol{f}(\tilde{\boldsymbol{x}}) = (u_1,\ u_2)^{\mathrm{T}}$,

$$\boldsymbol{g}(\tilde{\boldsymbol{x}}) = \begin{pmatrix} x_1 + (\cos(2\pi x_1) - 1)x_2 \\ \frac{1}{1+\exp(-5x_1)-\frac{1}{2}+\cos(\pi x_2)} \end{pmatrix},$$

and $\boldsymbol{w}_t \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(0.01, 0.01))$. Due to its highly nonlinear dynamics, it is impossible to extrapolate the system's behavior from locally collected data. Hence, control tasks that correspond to different portions of the state space require distinct measurements to achieve good performance.

We assume to know that $\boldsymbol{g}(\cdot)$ depends exclusively on $\boldsymbol{x}$, hence we use a GP that takes only the state $\boldsymbol{x}$ as input. Moreover, we employ a squared-exponential kernel $k(\cdot,\cdot)$ for the GP, which is able to approximate a continuous function arbitrarily accurately on compact sets [22]. We employ GP-based feedback linearizing control laws $\boldsymbol{u}^j(\boldsymbol{x}, \mathcal{D}_N, t) = -\boldsymbol{\mu}_{N,t}(\boldsymbol{x}) + \boldsymbol{x}^j_{\mathrm{ref}}(t)$ with 3 different reference trajectories

$$\boldsymbol{x}^1_{\mathrm{ref}}(t) = \mathbf{0} \tag{13}$$
$$\boldsymbol{x}^2_{\mathrm{ref}}(t) = \begin{pmatrix}\sin(2\pi t/50) & \cos(2\pi t/50)\end{pmatrix}^{\mathrm{T}} \tag{14}$$
$$\boldsymbol{x}^3_{\mathrm{ref}}(t) = \begin{pmatrix}2\sin(2\pi t/25) & \cos(2\pi t/100)\end{pmatrix}^{\mathrm{T}}. \tag{15}$$

The GP used to compute the mean $\boldsymbol{\mu}_{N,t}(\cdot)$ is identical to the one used to obtain the approximate optimal data set $\tilde{\mathcal{X}}_N^{M,*}$. Each control law is required to fulfill a single tracking performance requirement $h_t^j(\boldsymbol{x}) \leq 0$, $j = 1, 2, 3$ where

$$h_t^j(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}_{\mathrm{ref},j}(t)\|_2 - \varphi(t),\ j = 1, 2, \tag{16}$$
$$h_t^3(\boldsymbol{x}) = |x_1| - 5/2, \tag{17}$$

and $\varphi(t) := \max\{3\exp(-t/5), 0.1\}$, over a time horizon of $H = 100$ steps. We assume that the optimal data set is contained within $\tilde{\mathbb{X}}^* = [-3, 3]^2$, since the control objectives are restricted to this region. Furthermore, we are given 100 prior measurements taken from random samples of the true system, which we use to train the GP. The number of samples used to obtain the approximate optimal data set $\tilde{\mathcal{X}}_N^{M,*}$ is set to $M = 100$, and the desired probability of constraint violation is set to $\delta = 0.01$.

In order to solve the approximate optimization problem, we search for a solution by minimizing the surrogate function

$$\frac{1}{M}\sum_{m=1}^{M}\prod_{n=0}^{\tilde{N}-1} h_{t_n}^{j_n}(\tilde{\boldsymbol{s}}_n^m)\mathbf{1}_{\mathbb{R}^{d_{\tilde{x}}}_-}\left(h_{t_n}^{j_n}(\tilde{\boldsymbol{s}}_n^m)\right),$$

which enables us to employ gradient-based methods.

We apply the proposed technique 10 times using randomly sampled starting points $\boldsymbol{x}_0 \in \mathcal{U}([-3, 3]^2)$ each time, and obtain an approximate optimal data set $\tilde{\mathcal{X}}_N^{M,*}$ after $N \in \{6, \ldots, 12\}$ iterations of Algorithm 1. The approximate probability of constraint violation as a function of N is shown in Fig. 1. The prior system measurements, the desired trajectories, and an approximate optimal set $\tilde{\mathcal{X}}_N^{M,*}$ obtained after applying Algorithm 1 can be seen in Fig. 2.

All approximate optimal sets $\tilde{\mathcal{X}}_N^{M,*}$ correspond roughly to points within the circle given by $\boldsymbol{x}_d^2(t)$. This result is intuitive, since this is the region where the desired trajectories specified by (16) and (17) overlap the most. Moreover, as can be seen in Fig. 2, the approximate optimal solution $\tilde{\mathcal{X}}_N^{M,*}$ lies in regions that are both unexplored and of interest to the individual control tasks. However, since we employed a gradient-based solver to a non-convex problem, sub-optimal solutions are to be expected. This can also be seen in Fig. 2, where some data points are close to already available prior data, i.e., a local minimum was found.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/LCSYS.2020.3006279

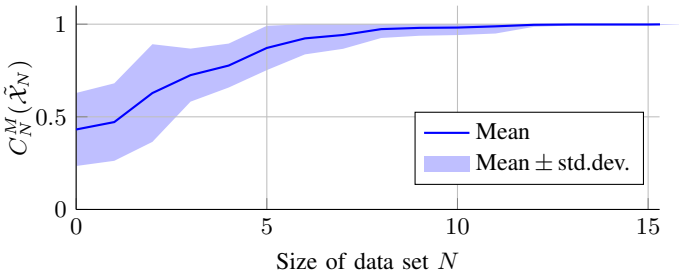

Fig. 1: Maximal approximate probability of constraint satisfaction $C_N^M(\tilde{\mathcal{X}}_N^M)$ as a function of data set size N for 10 repetitions of Algorithm 1. Desired probability of constraint satisfaction $1-\delta$ is achieved after $N \in \{6, \ldots, 12\}$ iterations of Algorithm 1.

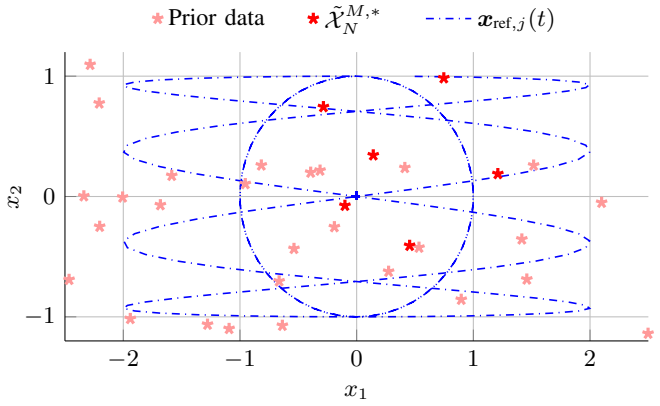

Fig. 2: Prior measurement data, reference trajectories $\boldsymbol{x}_{\text{ref},j}(t)$, and approximate optimal measurement locations $\tilde{\mathcal{X}}_N^{M,*}$ obtained with a single application of Algorithm 1 using $M = 50$.

After every completion of Algorithm 1, measurements of the true system at the approximate optimal set $\tilde{\mathcal{X}}_N^{M,*}$ are collected, and we carry out 100 Monte Carlo simulations of the true system. This results in no constraint violations except for task $j = 2$. However, constraint violations are small, as can be seen in Fig. 3, which indicates that the proposed method yielded a good approximate optimal data set $\tilde{\mathcal{X}}_N^{M,*}$.

VI. CONCLUSION AND FUTURE WORK

We have presented an algorithm that approximates the smallest training set required for achieving high performance across multiple learning-based control tasks with high probability. We use a sample-based approximation that approximates the correct solution arbitrarily well with probability 1 as the number of samples increases. In a numerical simulation, the approximate optimal data sets computed with the proposed method yielded adequate control laws for multiple tasks after. Extensions of the present paper include investigating the sample complexity of the proposed algorithm, and using it to design system exploration approaches.

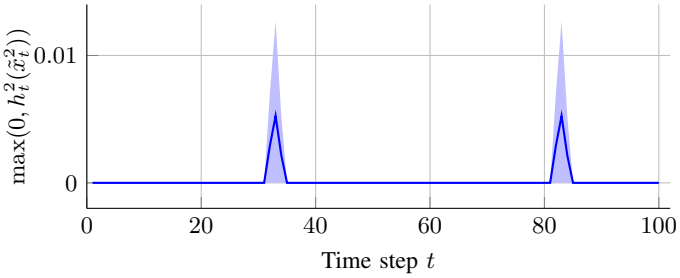

Fig. 3: Constraint violations yielded by applying control law $u_t^2(\cdot)$ to true system after data was collected at approximate optimal set $\tilde{\mathcal{X}}_N^{M,*}$ computed by Algorithm 1.

REFERENCES

[1] J. Umlauft, T. Beckers, A. Capone, A. Lederer, and S. Hirche, "Smart forgetting for safe online learning with Gaussian processes," in *2nd Learning for Dynamics and Control Conference (L4DC)*. PMLR, 2020.

[2] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 408–423, 2015.

[3] J. Kocijan, *Modelling and control of dynamic systems using Gaussian process models*. Springer, 2016.

[4] A. Wilson, A. Fern, S. Ray, and P. Tadepalli, "Multi-task reinforcement learning: A hierarchical Bayesian approach," in *Proc. of the 24th International Conference on Machine Learning*. Association for Computing Machinery, 2007, p. 1015–1022.

[5] M. P. Deisenroth, P. Englert, J. Peters, and D. Fox, "Multi-task policy search for robotics," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 3876–3881.

[6] M. P. Deisenroth, G. Neumann, and J. Peters, "A survey on policy search for robotics," *Foundations and Trends in Robotics*, vol. 2, no. 1–2, pp. 1–142, 2013.

[7] M. Hessel, H. Soyer, L. Espeholt, W. Czarnecki, S. Schmitt, and H. van Hasselt, "Multi-task deep reinforcement learning with popart," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3796–3803.

[8] Y. Teh, V. Bapst, W. M. Czarnecki, J. Quan, J. Kirkpatrick, R. Hadsell, N. Heess, and R. Pascanu, "Distral: Robust multitask reinforcement learning," in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 4496–4506.

[9] B. C. Da Silva, G. Konidaris, and A. G. Barto, "Learning parameterized skills," in *Proc. of the 29th International Conference on Machine Learning*, 2012, pp. 1443–1450.

[10] H. Durrant-Whyte, N. Roy, and P. Abbeel, *Learning to Control a Low-Cost Manipulator Using Data-Efficient Reinforcement Learning*, 2012, pp. 57–64.

[11] F. Pukelsheim, *Optimal Design of Experiments*. SIAM, 2006.

[12] C. Zimmer, M. Meister, and D. Nguyen-Tuong, "Safe active learning for time-series modeling with Gaussian processes," in *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 2730–2739.

[13] J. Schreiter, D. Nguyen-Tuong, M. Eberts, B. Bischoff, H. Markert, and M. Toussaint, "Safe exploration for active learning with Gaussian processes," in *Machine Learning and Knowledge Discovery in Databases*. Springer International Publishing, 2015, pp. 133–149.

[14] A. Capone, G. Noske, J. Umlauft, T. Beckers, A. Lederer, and S. Hirche, "Localized active learning of gaussian process state space models," in *2nd Learning for Dynamics and Control Conference (L4DC)*. PMLR, 2020.

[15] A. Capone and S. Hirche, "Backstepping for partially unknown nonlinear systems using Gaussian processes," *IEEE Control Systems Letters*, vol. 3, pp. 416–421, 2019.

[16] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *2018 IEEE Conference on Decision and Control*, 2018, pp. 6059–6066.

[17] C. E. Rasmussen and C. K. Williams, "Gaussian processes for machine learning. 2006," *The MIT Press, Cambridge, MA, USA*, 2006.

[18] L. Ljung, "System identification," in *Signal analysis and prediction*. Springer, 1998, pp. 163–173.

[19] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.

[20] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *Artificial Intelligence and Statistics*, 2009, pp. 567–574.

[21] S. G. Krantz and H. R. Parks, *A primer of real analytic functions*. Springer Science & Business Media, 2002.

[22] G. Wahba, *Spline models for observational data*. SIAM, 1990, vol. 59.

Copyright (c) 2020 IEEE. Personal use is permitted. For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org.