Proceedings of Machine Learning Research vol 120:1–11, 2020 2nd Annual Conference on Learning for Dynamics and Control

Parameter Optimization for Learning-based Control of Control-Affine Systems

Armin Lederer ARMIN.LEDERER@TUM.DE
Alexandre Capone ALEXANDRE.CAPONE@TUM.DE
Sandra Hirche HIRCHE@TUM.DE
Chair of Information-oriented Control
Department of Electrical and Computer Engineering
Technical University of Munich
D-80333 Munich, Germany

Editors: A. Bayen, A. Jadbabaie, G. J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M. Zeilinger

Abstract

Supervised machine learning is often applied to identify system dynamics where first principle methods fail. When combining learning with control methods, probabilistic regression is typically applied to increase robustness against learning errors. Although this combination of probabilistic regression and control theory allows to formulate performance guarantees for many control techniques, the bounds are usually conservative, and cannot be employed for efficient control parameter tuning. Therefore, we reformulate the parameter tuning problem using robust optimization with performance constraints based on Lyapunov theory. By relaxing the problem through scenario optimization, we derive a with high probability optimal method for control parameter tuning. We demonstrate its flexibility and efficiency on parameter tuning problems for a feedback linearizing and a computed torque controller.

Keywords: data-driven control, control parameter tuning, scenario optimization, probabilistic machine learning, safe learning-based control

1. Introduction

Due to its flexibility, supervised machine learning is increasingly used to model systems that are difficult to describe using first principle methods. However, these regression methods often suffer from modeling errors, e.g., because of a lack of training data or measurement noise. Deterministic regression techniques such as feed forward neural networks (Goodfellow et al., 2016) ignore this issue, which has prevented their widespread use in control applications. Therefore, incorporating the uncertainty of learned models is a crucial component in model-based learning control (Chua et al., 2018), and methods such as Gaussian process regression (Rasmussen and Williams, 2006), Bayesian neural networks (Depeweg et al., 2017) or ensembles of probabilistic neural networks (Lakshminarayanan et al., 2017) have gained growing attention in system identification for control.

Control laws for these probabilistic models are often designed using model-based reinforcement learning (Sutton and Barto, 2017), which uses interactions with the real system to learn the probabilistic model and computes controllers to achieve a specified goal. While many approaches addressing safety (Berkenkamp et al., 2017; Westenbroek et al., 2019) and sample-efficiency (Chua et al., 2018) have been proposed, guarantees on the achievable performance with a specific controller have received little attention. Bayesian optimization (Brochu et al., 2010) is a method which is often used in model-free reinforcement learning problems for tuning control parameters (Marco et al., 2016; Duivenvoorden et al., 2017; Schillinger et al., 2017). These controller tuning approaches typically consider discrete time systems, aim to actively reduce uncertainty in an iterative process, and learn the relationship between control parameters and the aggregate cost. Therefore, they provide very limited information about the predicted stationary tracking error and do not allow any statements about stability. Furthermore, the application of these algorithms in an offline setting can lead to poor control performance due to the active uncertainty reduction. In contrast to these controller tuning approaches, continuous-time nonlinear control techniques, such as feedback linearization and backstepping exhibit inherent robustness against model uncertainties, such that performance bounds can be guaranteed based on uniform regression error bounds (Berkenkamp and Schoellig, 2015; Umlauft et al., 2017; Beckers et al., 2017; Umlauft et al., 2018; Koller et al., 2018; Capone and Hirche, 2019). However, uniform error bounds are often conservative (Srinivas et al., 2012; Chowdhury and Gopalan, 2017; Lederer et al., 2019), leading to a poor choice of control parameters, e.g., exceedingly high gains to achieve performance specifications.

In order to mitigate these issues, we address the problem of computing control parameters for continuous-time systems, such that performance specifications are satisfied and poor choices of control parameters are avoided. We approach this problem by formulating it as robust optimization problem, in which we consider the dynamics in Lyapunov stability-based constraints. Moreover, poor choices of control parameters are penalized through the objective of the robust optimization. This ensures the satisfaction of the performance satisfaction while minimizing negative effects of the control parameter choice. By relaxing this problem using scenario optimization (Campi et al., 2009, 2015), we develop an efficient and nonconservative method to tune control parameters while considering probabilistic model uncertainties. We show that the optimality of the derived solution holds with high probability and demonstrate the efficiency of the method with two examples.

The remainder of this article is structured as follows. In Section 2 we formalize the parameter tuning problem. The uncertainty-aware scenario control parameter tuning algorithm is derived in Section 3 and evaluated in Section 4, followed by a conclusion in Section 5.

2. Problem Description

2.1. Formal Problem Statement

Consider a control-affine system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}$ with state $\boldsymbol{x} \in \mathbb{X} \subset \mathbb{R}^{d_x}$, control input $\boldsymbol{u} \in \mathbb{U} \subset \mathbb{R}^{d_u}$, unknown function $\boldsymbol{f} : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_x}$ and potentially unknown matrix function $\boldsymbol{G} : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_x \times d_u}$. We make the assumptions detailed in the sequel.

Assumption 1 *A prior model $\hat{\boldsymbol{f}}(\cdot)$ of the unknown function $\boldsymbol{f}(\cdot)$ is given and noisy training data*

$$\boldsymbol{y}^{(n)} = \boldsymbol{f}(\boldsymbol{x}^{(n)}) - \hat{\boldsymbol{f}}(\boldsymbol{x}^{(n)}) + \boldsymbol{\omega}^{(n)}, \qquad \boldsymbol{\omega}^{(n)} \sim \mathcal{N}(\boldsymbol{0}, \sigma_n^2 \boldsymbol{I}_{d_x}) \ i.i.d. \tag{1}$$

is available in the form of a training data set $\mathbb{D} = \{(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)})\}_{n=1}^N$.

The assumption of state derivative measurements perturbed by Gaussian noise is commonly used in learning-based control, and is caused, for example, by numerical differentiation (Umlauft et al., 2017). We apply probabilistic supervised machine learning to the training data set to obtain a compensation of the model error between the prior model $\hat{\boldsymbol{f}}(\cdot)$ and the true model $\boldsymbol{f}(\cdot)$ in the form of a predictive probability density function $p : \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \mapsto \mathbb{R}_+$. The mean function $\boldsymbol{\mu} : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_x}$ of this distribution is typically used to improve the prior model $\hat{\boldsymbol{f}}(\cdot)$, while the probability $p(\cdot|\cdot)$ is used to define confidence regions. Based on the learned system model, the task is to track a continuously differentiable and bounded reference trajectory $\boldsymbol{x}_{\text{ref}}(t)$, which generates a bounded set $\mathbb{X}_{\text{ref}} = \{\boldsymbol{x} \in \mathbb{X} : \boldsymbol{x} = \boldsymbol{x}_{\text{ref}}(t), t > 0\}$. In order to address this task, we employ control laws with the structure

$$\boldsymbol{u} = \boldsymbol{\phi}_{\hat{f},\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{x}_{\text{ref}}, \dot{\boldsymbol{x}}_{\text{ref}}) + \boldsymbol{\kappa}_\theta(\boldsymbol{x}, \boldsymbol{x}_{\text{ref}}) \tag{2}$$

with feed forward term $\boldsymbol{\phi}_{\hat{f},\mu} : \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_u}$ and feedback controller $\boldsymbol{\kappa}_\theta : \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_u}$, which is parameterized by $\boldsymbol{\theta} \in \mathbb{S}$. This control structure is typically robust against perturbations and captures many frequently used learning-based control laws, e.g., feedback linearization (Umlauft et al., 2017; Shi et al., 2019), computed torque control (Beckers et al., 2019), and backstepping controllers (Capone and Hirche, 2019). Robustness is often formalized using Lyapunov stability theory, which we employ to define our performance specification as follows:

Assumption 2 *The performance specification for the tracking error $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{x}_{\text{ref}}$ is expressed through the stability condition $\dot{V}(\boldsymbol{e}) < 0$ for all $\boldsymbol{e} \in \mathbb{R}^{d_x} : \|\boldsymbol{e}\| \leq \bar{e}$, where $V : \mathbb{R}^{d_x} \to \mathbb{R}_+$ denotes a given Lyapunov function and $\bar{e}$ the prescribed upper tracking error bound.*

Although the choice of the Lyapunov function plays a crucial role in the performance specification, many system structures such as feedback linearizable systems or Euler-Lagrange systems allow a straightforward definition of $V(\cdot)$, which ensures ultimate boundedness (Umlauft et al., 2017; Beckers et al., 2019; Shi et al., 2019). Therefore, Assumption 2 typically allows a simple and flexible performance specification. In addition to the specification of the tracking error bound, we want to consider subordinate control design goals, e.g., avoiding exceedingly high control gains. These additional design goals are formulated in the following form:

Assumption 3 *Subordinate control design goals are expressed through a cost function $l : \mathbb{S} \mapsto \mathbb{R}_+$.*

The cost function $l(\cdot)$ allows to penalize control parameters leading to an undesired control behavior in a straightforward way, while the tracking error is specified in the form of a constraint. Therefore, the problem lies in finding parameters $\boldsymbol{\theta}$ that minimize the cost fuction $l(\cdot)$, but also ensure the satisfaction of the performance specification in the form of Assumption 2.

2.2. Illustrative Example

In order to illustrate the flexibility of the presented problem formulation, we consider a 2-DoF planar robotic manipulator. The manipulator dynamics are given by $\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{q} = \boldsymbol{u}$, where $\boldsymbol{q} \in \mathbb{R}^2$ and $\dot{\boldsymbol{q}} \in \mathbb{R}^2$ are the joint angles and the joint angle velocities, respectively, and $\boldsymbol{u} \in \mathbb{R}^2$ are the torque inputs. The angles and velocities are concatenated in the state $\boldsymbol{x} = [\boldsymbol{q} \quad \dot{\boldsymbol{q}}]^T$. Furthermore, we define a reference trajectory $\boldsymbol{x}_{\text{ref}} = [\boldsymbol{q}_{\text{ref}} \quad \dot{\boldsymbol{q}}_{\text{ref}}]^T$ and employ a control law formulation and Lyapunov function based on (Beckers et al., 2019), which employs a Gaussian process to learn the system dynamics, resulting in a Gaussian predictive distribution $p(\cdot|\cdot)$ with mean function $\boldsymbol{\mu}(\cdot)$. The feedforward and feedback terms of the control law are computed as

$$\boldsymbol{\phi}_{\hat{f},\mu}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{q}_{\text{ref}}, \ddot{\boldsymbol{q}}_{\text{ref}}) = \hat{\boldsymbol{H}}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\text{ref}} + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{q}_{\text{ref}} + \boldsymbol{\mu}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \tag{3}$$

$$\boldsymbol{\kappa}_\theta(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{q}_{\text{ref}}, \dot{\boldsymbol{q}}_{\text{ref}}) = -\boldsymbol{K}_p(\boldsymbol{q}, \dot{\boldsymbol{q}})(\boldsymbol{q} - \boldsymbol{q}_{\text{ref}}) - \boldsymbol{K}_d(\boldsymbol{q}, \dot{\boldsymbol{q}})(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_{\text{ref}}) \tag{4}$$

where $\hat{\boldsymbol{H}}(\boldsymbol{q})$ and $\hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ represent the prior model $\hat{\boldsymbol{f}}(\cdot)$. We define the diagonal matrices $\boldsymbol{K}_p, \boldsymbol{K}_d$ as state dependent functions in order to allow uncertainty adaptation of the control gains. The entries on the diagonals are given by networks of ν radial basis functions each, i.e., $K_{i,i} = \left(\sum_{k=1}^{\nu} \gamma_k^i \exp(-1/\lambda_k^i(\|\boldsymbol{q} - \boldsymbol{\zeta}_k^i\|_2 + \|\dot{\boldsymbol{q}} - \boldsymbol{\psi}_k^i\|_2))\right)^2$, where $\gamma_k^i \in \mathbb{R}$ are amplitudes, $\lambda_k^i > 0$ denote length scales, and $\boldsymbol{\zeta}_k^i, \boldsymbol{\psi}_k^i \in \mathbb{R}^2$ are radial basis function centers. All these parameters are concatenated into the parameter vector $\boldsymbol{\theta}$, and we employ the cost function $l(\boldsymbol{\theta}) = \sum_{i=1}^{2}\sum_{k=1}^{\nu}(\lambda_k^i)^2 + (\gamma_k^i)^2$ in order to obtain a feedback controller, which exhibits low gains in large regions of the state space. Finally, we define the performance specification based on the Lyapunov function

$$V(\boldsymbol{e}_q, \dot{\boldsymbol{e}}_q) = \frac{1}{2}\dot{\boldsymbol{e}}_q^{\mathrm{T}}\hat{\boldsymbol{H}}(\boldsymbol{q})\dot{\boldsymbol{e}}_q + \int_0^{\boldsymbol{e}_q}\boldsymbol{z}^{\mathrm{T}}\boldsymbol{K}_p(\boldsymbol{z}+\boldsymbol{q}_{\mathrm{ref}})d\boldsymbol{z} + \varepsilon\boldsymbol{e}_q^{\mathrm{T}}\hat{\boldsymbol{H}}(\boldsymbol{q})\dot{\boldsymbol{e}}_q, \tag{5}$$

with $\boldsymbol{e}_q = \boldsymbol{q} - \boldsymbol{q}_{\mathrm{ref}}$ and $\dot{\boldsymbol{e}}_q = \dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_{\mathrm{ref}}$, since it allows to prove ultimate boundedness of the tracking error (Beckers et al., 2019).

3. Uncertainty-Aware Control Gain Tuning

3.1. Control Parameter Tuning as Robust Optimization Problem

In order to avoid the use of uniform error bounds for tuning, we formulate the problem of determining optimal control parameters as a robust optimization problem. Hence, we employ the Lyapunov function $V(\cdot)$ from Assumption 2, and calculate its time derivative as

$$\dot{V}(\boldsymbol{e},\boldsymbol{\theta}) = (\nabla_{\boldsymbol{e}}V(\boldsymbol{e}))^T\dot{\boldsymbol{e}} = (\nabla_{\boldsymbol{e}}V(\boldsymbol{e}))^T(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{\phi}_{\hat{f},\mu}(\boldsymbol{x},\boldsymbol{x}_{\mathrm{ref}},\dot{\boldsymbol{x}}_{\mathrm{ref}}) - \boldsymbol{\kappa}_\theta(\boldsymbol{x},\boldsymbol{x}_{\mathrm{ref}})).$$

By leveraging the probabilistic nature of the model obtained from machine learning, we equivalently consider the unknown function $\boldsymbol{f}(\cdot)$ as a random process described by $\boldsymbol{f}(\boldsymbol{x}) = \hat{\boldsymbol{f}}(\boldsymbol{x}) + \boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \tilde{p}(\cdot|\boldsymbol{x})$ and $\tilde{p}(\boldsymbol{\xi}|\boldsymbol{x}) = p(\boldsymbol{\xi} - \boldsymbol{\mu}(\boldsymbol{x})|\boldsymbol{x})$. Hence, we decouple the time derivative of the Lyapunov function into a nominal component

$$\dot{V}_{\mathrm{nom}}(\boldsymbol{e},\boldsymbol{\theta}) = (\nabla_{\boldsymbol{e}}V(\boldsymbol{e}))^T(\hat{\boldsymbol{f}}(\boldsymbol{x}) + \boldsymbol{\mu}(\boldsymbol{x}) - \boldsymbol{\phi}_{\hat{f},\mu}(\boldsymbol{x},\boldsymbol{x}_{\mathrm{ref}},\dot{\boldsymbol{x}}_{\mathrm{ref}}) - \boldsymbol{\kappa}_\theta(\boldsymbol{x},\boldsymbol{x}_{\mathrm{ref}})) \tag{6}$$

and an uncertain component $\dot{V}_{\boldsymbol{\xi}}(\boldsymbol{e}) = (\nabla_{\boldsymbol{e}}V(\boldsymbol{e}))^T\boldsymbol{\xi}$. The Lyapunov function can often be chosen such that the nominal derivative $\dot{V}_{\mathrm{nom}}(\boldsymbol{e},\boldsymbol{\theta})$ is negative definite (Umlauft et al., 2017; Beckers et al., 2019). However, the control parameters $\boldsymbol{\theta}$ influence its magnitude. In contrast, the uncertain component $\dot{V}_{\boldsymbol{\xi}}(\boldsymbol{e})$ is independent of the control parameters $\boldsymbol{\theta}$ but usually exhibits positive values. Employing this decoupling, we formulate the performance specification $\|\boldsymbol{e}\| \leq \bar{e}$ as $\dot{V}_{\boldsymbol{\xi}}(\boldsymbol{e}) < -\dot{V}_{\mathrm{nom}}(\boldsymbol{e},\boldsymbol{\theta})$, $\forall \boldsymbol{e}\colon \|\boldsymbol{e}\| > \bar{e}$, such that the nominal Lyapunov function derivative magnitude can be interpreted as a stability margin which is independent of the uncertainty. In order to formulate an optimization problem that includes the Lyapunov stability condition as a constraint, we define the set of states for the stability constraint as $\mathbb{T}(\boldsymbol{x}') = \{\boldsymbol{x} \in \mathbb{X}\colon\ \|\boldsymbol{x} - \boldsymbol{x}'\| > \bar{e}\}$. Then, the optimal control parameters are obtained by solving the robust optimization problem

$$\tilde{\boldsymbol{\theta}}^* = \arg\min_{\boldsymbol{\theta}\in\mathbb{S}} l(\boldsymbol{\theta}) \tag{7a}$$

$$\text{such that } \dot{V}_{\boldsymbol{\xi}}(\boldsymbol{x}-\boldsymbol{x}') < -\dot{V}_{\mathrm{nom}}(\boldsymbol{x}-\boldsymbol{x}',\boldsymbol{\theta}), \qquad \forall\boldsymbol{\xi}\sim\tilde{p}(\cdot|\boldsymbol{x}),\ \forall\boldsymbol{x}\in\mathbb{T}(\boldsymbol{x}'),\ \forall\boldsymbol{x}'\in\mathbb{X}_{\mathrm{ref}}. \tag{7b}$$

3.2. Problem Relaxation through Scenario Optimization

Although the parameter tuning problem can be elegantly cast as a robust optimization problem, it is well known that this type of problem is in general intractable (Calafiore and Campi, 2006). Scenario optimization (Campi et al., 2009) offers a solution to this problem by considering only a finite number of sample constraints each with fixed reference states $\boldsymbol{x}' \in \mathbb{X}_{\mathrm{ref}}$, states $\boldsymbol{x} \in \mathbb{T}(\boldsymbol{x}')$ and disturbances $\boldsymbol{\xi} \sim \tilde{p}(\cdot|\boldsymbol{x})$. This contrasts with the single constraint in robust optimization, which has to hold jointly for all possible realizations of these variables. For this reason, we take a fully probabilistic point of view by defining an equivalent joint probability distribution $p(\boldsymbol{\xi},\boldsymbol{x},\boldsymbol{x}') =$

Algorithm 1 Greedy Optimization Algorithm

Function `Optimize(`$\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(M)}, \boldsymbol{x}'^{(1)}, \dots, \boldsymbol{x}'^{(M)}, \boldsymbol{\xi}^{(1)}, \dots, \boldsymbol{\xi}^{(M)}$`)`**:**

  $\boldsymbol{s}_M \leftarrow \mathbf{0}$; $\boldsymbol{t}_M \leftarrow \mathbf{0}$; $\boldsymbol{h} \leftarrow \boldsymbol{h}_0$

  repeat

    $i \leftarrow \arg\max_{i=1,\dots,M} \dot{V}_{\boldsymbol{\xi}^{(i)}}(\boldsymbol{x}^{(i)} - \boldsymbol{x}'^{(i)}) + \dot{V}_{\text{nom}}(\boldsymbol{x}^{(i)} - \boldsymbol{x}'^{(i)}, \boldsymbol{\theta})$; $s_{M,i} \leftarrow 1$

    solve (10) to update $\boldsymbol{\theta}$ using deterministic, numerical optimization algorithm

  until $\dot{V}_{\boldsymbol{\xi}^{(i)}}(\boldsymbol{x}^{(i)} - \boldsymbol{x}'^{(i)}) < -\dot{V}_{\text{nom}}(\boldsymbol{x}^{(i)} - \boldsymbol{x}'^{(i)}, \boldsymbol{\theta}^*)$, $\forall i = 1, \dots, M$;

return $\boldsymbol{\theta}$, $\|\boldsymbol{s}_M\|_1$

$\tilde{p}(\boldsymbol{\xi}|\boldsymbol{x})p(\boldsymbol{x}|\boldsymbol{x}')p(\boldsymbol{x}')$, with the distributions $p(\boldsymbol{x}|\boldsymbol{x}') = \mathcal{U}(\mathbb{T}(\boldsymbol{x}'))$, $p(\boldsymbol{x}') = \mathcal{U}(\mathbb{X}_{\text{ref}})$ and the mean-free learned predictive distribution $\tilde{p}(\cdot, \cdot)$. Based on these probability distributions, we relax the robust constraint (7b) to a chance constraint

$$P\left(\boldsymbol{\xi} \sim \tilde{p}(\cdot|\boldsymbol{x}), \boldsymbol{x} \sim \mathcal{U}(\mathbb{T}(\boldsymbol{x}')), \boldsymbol{x}' \sim \mathcal{U}(\mathbb{X}_{\text{ref}}) \colon \dot{V}_{\boldsymbol{\xi}}(\boldsymbol{x} - \boldsymbol{x}') < -\dot{V}_{\text{nom}}(\boldsymbol{x} - \boldsymbol{x}', \boldsymbol{\theta})\right) \geq 1 - \bar{\epsilon}, \tag{8}$$

where the parameter $\bar{\epsilon} \in [0, 1]$ measures the allowed violation probability for fixed control parameters $\boldsymbol{\theta}$. Due to this chance constraint relaxation, a solution $\boldsymbol{\theta}^*$ merely approximately satisfies the original problem (7a). Hence, it is called $\bar{\epsilon}$-robust solution (Calafiore and Campi, 2006). In spite of this relaxation, an optimization problem with constraint (8) is generally infeasible. Therefore, we sample from the joint probability distribution $p(\boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{x}')$ sequentially as $\boldsymbol{x}'^{(i)} \sim \mathcal{U}(\mathbb{X}_{\text{ref}})$, $\boldsymbol{x}^{(i)} \sim \mathcal{U}(\mathbb{T}(\boldsymbol{x}'^{(i)}))$ and $\boldsymbol{\xi} \sim \tilde{p}(\cdot|\boldsymbol{x}^{(i)})$, $i = 1, \dots, M$. These samples are used to define scenario constraints, which yield the scenario control parameter tuning problem

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \mathbb{S}} l(\boldsymbol{\theta}) \tag{9a}$$

$$\text{such that } \dot{V}_{\boldsymbol{\xi}^{(i)}}(\boldsymbol{x}^{(i)} - \boldsymbol{x}'^{(i)}) < -\dot{V}_{\text{nom}}(\boldsymbol{x}^{(i)} - \boldsymbol{x}'^{(i)}, \boldsymbol{\theta}), \qquad \forall i = 1, \dots, M. \tag{9b}$$

Since the solution $\boldsymbol{\theta}^*$ depends only on a finite number of constraints with fixed $\boldsymbol{x}$, $\boldsymbol{x}'$ and $\boldsymbol{\xi}$, we can use gradient based optimization methods to find (local) minima. However, due to the randomly sampled scenario constraints, $\boldsymbol{\theta}^*$ becomes a random variable itself. Therefore, we can only aim to obtain a solution $\boldsymbol{\theta}^*$ of (9), which satisfies the chance constraint (8) with probability $\bar{\epsilon}$ with a confidence, or risk of failure, β. Furthermore, it is not necessary to consider all constraints during the optimization, since typically only a small subset of constraints is active, i.e., solving the reduced scenario optimization problem

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \mathbb{S}} l(\boldsymbol{\theta}) \tag{10a}$$

$$\text{such that } \dot{V}_{\boldsymbol{\xi}^{(i)}}(\boldsymbol{x}^{(i)} - \boldsymbol{x}'^{(i)}) < -\dot{V}_{\text{nom}}(\boldsymbol{x}^{(i)} - \boldsymbol{x}'^{(i)}, \boldsymbol{\theta}), \quad \forall i = 1, \dots, M \colon s_{M,i} = 1, \tag{10b}$$

where $\boldsymbol{s}_M$ is a binary vector with a 1 at active constraints and 0 everywhere else, yields a design parameter $\boldsymbol{\theta}^*$ satisfying all scenario constraints (9b). Since determining the active constraints a priori is generally not possible, we propose the greedy optimization in Algorithm 1, which adds the constraint with highest violation to the active constraints until all scenario constraints are satisfied. Based on this greedy optimization algorithm, we iteratively generate scenarios until the relationship between the number of active scenario constraints m and all scenario constraints M is sufficient given a confidence requirement β that is expressed through a function $\epsilon(m, M, \beta)$. This procedure is summarized in Algorithm 2. For suitable choices of $\epsilon(m, M, \beta)$ and feasible scenario optimization problems, the proposed algorithms guarantee satisfaction of the robust constraints (7b) with high probability as shown in the following theorem.

Algorithm 2 Scenario control parameter tuning

Function `SCPT(`$\bar{\epsilon}$, β`)`**:**
  $M \leftarrow 0$; $m \leftarrow 0$
  while $\bar{\epsilon} > \epsilon(M, m, \beta)$ **do**
    $M \leftarrow M + 1$
    sample $\boldsymbol{x}'^{(M)} \sim \mathcal{U}(\mathbb{X}_{\text{ref}}),\ \boldsymbol{x}^{(M)} \sim \mathcal{U}(\mathbb{T}(\boldsymbol{x}'^{(M)})),\ \boldsymbol{\xi}^{(M)} \sim \tilde{p}(\cdot|\boldsymbol{x}^{(M)})$
    if $\dot{V}_{\boldsymbol{\xi}^{(M)}}(\boldsymbol{x}^{(M)} - \boldsymbol{x}'^{(M)}) \geq -\dot{V}_{\text{nom}}(\boldsymbol{x}^{(M)} - \boldsymbol{x}'^{(M)}, \boldsymbol{\theta}^*)$ **then**
      $[\boldsymbol{\theta}^*, m] \leftarrow$ `Optimize(`$\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(M)}, \boldsymbol{x}'^{(1)}, \dots, \boldsymbol{x}'^{(M)}, \boldsymbol{\xi}^{(1)}, \dots, \boldsymbol{\xi}^{(M)}$`)`
    end
  end
return $\boldsymbol{\theta}^*$

Theorem 1 *Define*

$$\epsilon(M, m, \beta) = \begin{cases} 1 & \text{if } m = M \\ 1 - \sqrt[M-m]{\frac{\beta}{M\binom{M}{m}}} & \text{if } m \neq M. \end{cases} \tag{11}$$

and fix a confidence level $\beta \in (0, 1)$ *and violation probability bound* $\bar{\epsilon} \in (0, 1)$. *If the scenario optimization problem* (9) *is feasible for all* $\boldsymbol{x}'^{(i)} \sim \mathcal{U}(\mathbb{X}_{\text{ref}})$, $\boldsymbol{x}^{(i)} \sim \mathcal{U}(\mathbb{T}(\boldsymbol{x}'^{(i)}))$ *and* $\boldsymbol{\xi} \sim \tilde{p}(\cdot|\boldsymbol{x}^{(i)})$, *the optimal parameters* $\boldsymbol{\theta}^*$ *obtained through scenario parameter tuning with Algorithm 2 and greedy optimization in Algorithm 1 satisfy the constraints* (7b) $\bar{\epsilon}$*-robustly with probability of at least* $1 - \beta$, *i.e.,*

$$P\left(\boldsymbol{x}' \sim \mathcal{U}(\mathbb{X}_{\text{ref}}), \boldsymbol{x} \sim \mathcal{U}(\mathbb{T}(\boldsymbol{x}')), \boldsymbol{\xi} \sim p(\cdot|\boldsymbol{x}) :\ \dot{V}_{\boldsymbol{\xi}}(\boldsymbol{x} - \boldsymbol{x}') < -\dot{V}_{\text{nom}}(\boldsymbol{x} - \boldsymbol{x}', \boldsymbol{\theta}^*)\right) \geq 1 - \bar{\epsilon} \tag{12}$$

holds with confidence $1 - \beta$.

Proof Since the optimization problem (9) is feasible, Algorithm 2 returns a solution deterministically. Therefore, it fulfills the requirements of (Campi et al., 2018, Assumption 1). Furthermore, (11) satisfies the conditions of (Campi et al., 2018, Theorem 1) which yields the $\bar{\epsilon}$-robust satisfaction of the constraints (7b) with confidence of at least $1 - \beta$. ■

The condition of feasibility for all possible scenarios basically requires that the magnitude of the nominal derivative can be increased sufficiently to compensate for the effects of the uncertainties. Therefore, this condition can be ensured independently of the uncertainty by choosing a control parameterization which is capable of achieving arbitrarily high magnitudes of the nominal derivative. It can be easily seen that this condition is satisfied for many well-established control structures such as feedback linearization (Umlauft et al., 2017) and computed torque control (Beckers et al., 2019).

4. Numerical Evaluation

4.1. PD Control Gain Tuning

We evaluate the proposed method[1] for static control gain tuning on an example system introduced in (Beckers et al., 2019). This system has internal dynamics

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} x_2 \\ -1 - x_2 - \frac{x_2^2 \sin(x_1 - c) - \sin(c)}{\cos(x_1 - c) - \frac{11}{10\cos(x_1 - c)}} \end{bmatrix} \tag{13}$$

1. Code is available at https://gitlab.lrz.de/alederer/pop4lbc

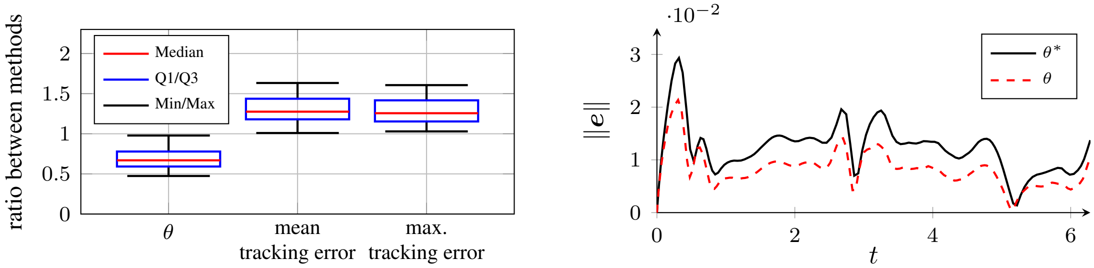

Figure 1: Left: Performance comparison between scenario tuned control gains θ^* and constant gains $\theta = 10$ in feedback linearization using Gaussian process models. All values are normalized by the respective value for the fixed gain θ. Right: Example tracking error trajectories of dynamics (13) with $c = 5.73$ controlled with $\theta^* = 5.92$ and constant gain $\theta = 10$.

with random but fixed $c \sim \mathcal{U}[0, 2\pi]$ and $G(\boldsymbol{x}) = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$. We ignore the nonlinearity in the approximate model, which leads to the prior model $\hat{\boldsymbol{f}}(\boldsymbol{x}) = \begin{bmatrix} x_2 & -1 - x_2 \end{bmatrix}^T$. For the control of this system we employ a feedback linearizing controller (Khalil, 2002) described by

$$\phi_{\hat{f},\mu}(\boldsymbol{x}, \boldsymbol{x}_{\text{ref}}, \dot{\boldsymbol{x}}_{\text{ref}}) = \dot{x}_{2,\text{ref}} - \begin{bmatrix} 0 & 1 \end{bmatrix} \hat{\boldsymbol{f}}(\boldsymbol{x}) - \mu(\boldsymbol{x}) \tag{14}$$

$$\kappa_\theta(\boldsymbol{x}, \boldsymbol{x}_{\text{ref}}) = -\theta(x_1 - x_{\text{ref},1}) - (\theta + 1)(x_2 - x_{\text{ref},2}), \qquad \theta > 0. \tag{15}$$

We introduce the filtered state $r = e_1 + e_2$ and employ the Lyapunov function $V(r) = r^2$ for stability analysis, which allows to prove ultimate boundedness of the error $\boldsymbol{e}$ (Umlauft et al., 2017). Therefore, the nominal and uncertain component of the Lyapuonv function derivative are given by $\dot{V}_{\text{nom}}(r, \theta) = -\theta r^2$ and $\dot{V}_\xi(r) = r\xi$. For determining the probabilistic model we employ Gaussian process regression (Rasmussen and Williams, 2006), such that the mean $\mu(\cdot)$ corresponds to the posterior mean function of the Gaussian process, while the uncertainty distribution is zero mean Gaussian, i.e., $\tilde{p}(\cdot|\boldsymbol{x}) = \mathcal{N}(0, \sigma^2(\boldsymbol{x}))$, where $\sigma^2(\boldsymbol{x})$ is the posterior variance of the Gaussian process. We train the Gaussian process with 241 data point on a uniform grid over $[-1, 1]^2$ and corresponding outputs perturbed by zero mean Gaussian noise with $\sigma_n^2 = 0.1$. The goal is to track the reference trajectory $\boldsymbol{x}_{\text{ref}} = [\sin(t) \quad \cos(t)]^T$. We tune the controller gain θ on $\mathbb{X} = [-1.2, 1.2]^2$ using Algorithm 1 and Algorithm 2 with the cost function $l(\theta) = \theta^2$ in order to obtain a low gain controller. We define the performance specification $\bar{e} = 0.1$, which corresponds to the upper bound $\bar{r} = \sqrt{2}\bar{e}$ for the filtered state r. Furthermore, we allow a violation probability of $\bar{\epsilon} = 0.01$ with confidence level $\beta = 10^{-9}$.

We evaluate the performance of our approach for 20 randomly sampled values of c and compare it to the performance of the augmented computed torque control with fixed control gain $\theta = 10$. The results are depicted in Fig. 1. The tuned control gain θ^* is smaller than the fixed gain θ on average. Although it can lead to higher tracking errors, the specified performance $\bar{e} = 0.1$ is achieved in all simulations with a maximum observed error of $\|\boldsymbol{e}\| = 0.0420$ for the tuned gains θ^*.

4.2. Neural Network Weight Tuning

We apply the proposed technique to tune the parameters of the radial basis function network as introduced in Section 2.2. We use $\nu = 10$ radial basis functions and define the unknown dynamics following (Murray et al., 1994) for manipulator arm masses of $m_1 = m_2 = 1\,\text{kg}$ and lengths of $l_1 = l_2 = 1\,\text{m}$. We define as reference trajectory $\boldsymbol{x}_{\text{ref}} = [\sin(t) \quad \cos(t) \quad \cos(t) \quad -\sin(t)]^T$. The

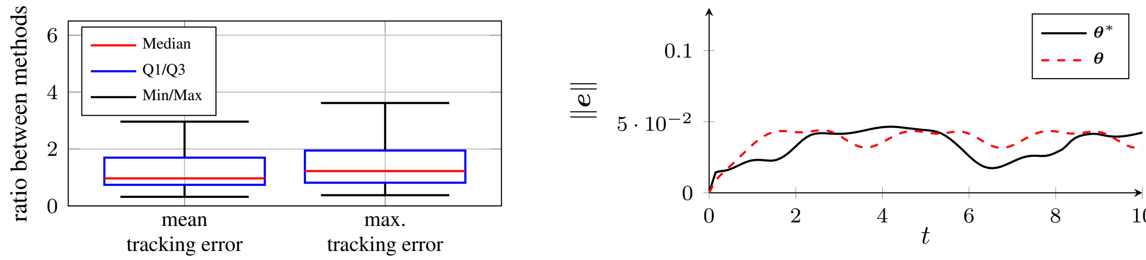

Figure 2: Left: Performance comparison between scenario tuned controller and constant gains $\boldsymbol{K}_p, \boldsymbol{K}_d$ in computed torque control augmented by Gaussian process models. All values are normalized by the respective value for the fixed gains $\boldsymbol{K}_p, \boldsymbol{K}_d$. Right: Example tracking error trajectories of 2-DoF robot controlled by CTC-GP and fixed gains.

components of the time derivative of the Lyapunov function in (5) are given by

$$\dot{V}_{\text{nom}}(\boldsymbol{e}_q, \dot{\boldsymbol{e}}_q, \boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{e}_q \\ \dot{\boldsymbol{e}}_q \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} -\boldsymbol{K}_d(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \varepsilon \hat{\boldsymbol{H}}(\boldsymbol{q}) & \frac{\varepsilon}{2}(-\boldsymbol{K}_d^{\mathrm{T}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})) \\ \frac{\varepsilon}{2}(-\boldsymbol{K}_d(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \hat{\boldsymbol{C}}^{\mathrm{T}}(\boldsymbol{q}, \dot{\boldsymbol{q}})) & -\varepsilon \boldsymbol{K}_p(\boldsymbol{q}, \dot{\boldsymbol{q}}) \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_q \\ \dot{\boldsymbol{e}}_q \end{bmatrix} \tag{16}$$

$$\dot{V}_{\boldsymbol{\xi}}(\boldsymbol{e}_q, \dot{\boldsymbol{e}}_q) = (\dot{\boldsymbol{e}}_q + \varepsilon \boldsymbol{e}_q)^{\mathrm{T}} \boldsymbol{\xi}. \tag{17}$$

We set the performance specification to $\bar{e} = 0.1$ and use a violation probability bound of $\bar{\epsilon} = 0.01$. Furthermore, we apply our method to 20 different samples of the estimated parameters $\eta_1^m, \eta_2^m, \eta_1^l l_1$ and η_2^l. The performance of the resulting control laws is compared to the one obtained using a controller with static gains $\boldsymbol{K}_p = \boldsymbol{K}_d = \text{diag}(1,1)$ in Fig. 2. Moreover, the tracking error of a sample control law is shown. As demonstrated in Fig. 2, the scenario optimization allows straightforward computation of radial basis function network controllers satisfying the desired performance, while also minimizing the amplitudes γ_k^i (average: 1.12) and length scales λ_k^i (average: 0.96).

5. Conclusion

We propose a scenario-based approach for tuning parameters of control laws in control-affine settings. The presented technique guarantees any chosen tracking performance with high probability. In simulation settings, we show that the gains obtained with the scenario-based approach achieve a performance similar to the one obtained with high control gains. In the future, we want to extend our approach to allow its application in an iterative fashion with exploration objectives.

Acknowledgments

This work has received funding from the Horizon 2020 research and innovation programme of the European Union under grant agreement n° 871767 of the project ReHyb: Rehabilitation based on hybrid neuroprosthesis. A. L. gratefully acknowledges financial support from the German Academic Scholarship Foundation.

References

Thomas Beckers, Jonas Umlauft, Dana Kulić, and Sandra Hirche. Stable Gaussian Process based Tracking Control of Lagrangian Systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5180–5185, 2017.

Thomas Beckers, Dana Kulić, and Sandra Hirche. Stable Gaussian Process based Tracking Control of Euler-Lagrange Systems. *Automatica*, 103(23):390–397, 2019.

Felix Berkenkamp and Angela P. Schoellig. Safe and Robust Learning Control with Gaussian Processes. In *Proceedings of the European Control Conference*, pages 2496–2501, 2015.

Felix Berkenkamp, Matteo Turchetta, Angela P. Schoellig, and Andreas Krause. Safe Model-based Reinforcement Learning with Stability Guarantees. In *Advances in Neural Information Processing Systems*, pages 908–918, 2017.

Eric Brochu, Vlad M. Cora, and Nando de Freitas. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. 2010. URL http://arxiv.org/abs/1012.2599.

G.C. Calafiore and M.C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

M. C. Campi, Simone Garatti, and Federico Alessandro Ramponi. A general scenario theory for non-convex optimization and decision making. *IEEE Transactions on Automatic Control*, 63(12): 4067–4078, 2018.

Marco C. Campi, Simone Garatti, and Maria Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149–157, 2009.

Marco C. Campi, Simone Garatti, and Federico A. Ramponi. Non-convex scenario optimization with application to system identification. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4023–4028, 2015.

Alexandre Capone and Sandra Hirche. Backstepping for Partially Unknown Nonlinear Systems Using Gaussian Processes. *IEEE Control Systems Letters*, 3(2):416–421, 2019.

Sayak Ray Chowdhury and Aditya Gopalan. On Kernelized Multi-armed Bandits. In *Proceedings of the International Conference on Machine Learning*, pages 844–853, 2017.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. In *Advances in Neural Information Processing Systems*, pages 4754–4765, 2018.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Learning and Policy Search in Stochastic Dynamical Systems with Bayesian Neural Networks. In *International Conference on Learning Representations*, pages 1–14, 2017.

Rikky R.P.R. Duivenvoorden, Felix Berkenkamp, Nicolas Carion, Andreas Krause, and Angela P. Schoellig. Constrained Bayesian Optimization with Particle Swarms for Safe Adaptive Controller Tuning. *Proceedings of the IFAC World Congress*, 50(1):11800–11807, 2017.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Hassan K. Khalil. *Nonlinear Systems*. Prentice-Hall, Upper Saddle River, NJ, third edition, 2002.

Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based Model Predictive Control for Safe Exploration. In *Proceedings of the IEEE Conference on Decision and Control*, pages 6059–6066, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems*, pages 6405–6416, 2017.

Armin Lederer, Jonas Umlauft, and Sandra Hirche. Uniform Error Bounds for Gaussian Process Regression with Application to Safe Control. In *Advances in Neural Information Processing Systems*, 2019.

Alonso Marco, Philipp Hennig, Jeannette Bohg, Stefan Schaal, and Sebastian Trimpe. Automatic LQR Tuning based on Gaussian Process Global Optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 270–277, 2016.

Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994. ISBN 9781351469791. doi: 10.1201/9781315136370.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA, 2006.

Mark Schillinger, Benjamin Hartmann, Patric Skalecki, Mona Meister, Duy Nguyen-Tuong, and Oliver Nelles. Safe Active Learning and Safe Bayesian Optimization for Tuning a PI-Controller. *Proceedings of the IFAC World Congress*, 50(1):5967–5972, 2017.

Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural Lander: Stable Drone Landing Control Using Learned Dynamics. In *Proceedings of the International Conference on Robotics and Automation*, pages 9784–9790, 2019.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2017.

Jonas Umlauft, Thomas Beckers, Melanie Kimmel, and Sandra Hirche. Feedback Linearization using Gaussian Processes. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5249–5255, 2017.

Jonas Umlauft, Lukas Pöhler, and Sandra Hirche. An Uncertainty-Based Control Lyapunov Approach for Control-Affine Systems Modeled by Gaussian Process. *IEEE Control Systems Letters*, 2(3):483–488, 2018.

Tyler Westenbroek, David Fridovich-Keil, Eric Mazumdar, Shreyas Arora, Valmik Prabhu, S. Shankar Sastry, and Claire J. Tomlin. Feedback Linearization for Unknown Systems via Reinforcement Learning. 2019. URL http://arxiv.org/abs/1910.13272.