Manuscript submitted to AIMS Journal

doi:10.3934/xx.xxxxxxx

SAFE TRAJECTORY TRACKING FOR UNDERACTUATED VEHICLES WITH PARTIALLY UNKNOWN DYNAMICS

Thomas Beckers*

Department of Electrical and Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104, USA

Leonardo J. Colombo

Centre for Automation and Robotics (CSIC-UPM)
Ctra. M300 Campo Real, Km 0,200, Arganda del Rey
Madrid, 28500, Spain

Sandra Hirche

Department of Electrical and Computer Engineering
Technical University of Munich
Munich, 80333, Germany

(Communicated by the associate editor name)

Abstract. Underactuated vehicles have gained much attention in the recent years due to the increasing amount of aerial and underwater vehicles as well as nanosatellites. The safe tracking control of these vehicles is a substantial aspect for an increasing range of application domains. However, external disturbances and parts of the internal dynamics are often unknown or very time-consuming to model. To overcome this issue, we present a safe tracking control law for underactuated rigid-body dynamics using a learning-based oracle for the prediction of the unknown dynamics. The presented approach guarantees a bounded tracking error with high probability where the bound is explicitly given. With additional assumptions, asymptotic stability of the tracking error is achieved. A numerical example highlights the effectiveness of the proposed control law.

1. **Introduction.** The demand for unmanned aerial and underwater vehicles is rapidly increasing in many areas such as monitoring, mapping, agriculture, and delivery. These vehicles are typically underactuated due to constructional reasons which poses several challenges from the control perspective [1, 2, 3]. The dynamics of these systems can often be expressed by rigid bodies motion with full attitude control and one translational force input. This is a classical problem in underactuated mechanics and many different types of control methods have been proposed to achieve an accurate trajectory tracking. Most of the control approaches are mainly based on feedback linearization [4, 5] and backstepping methods [6, 7] which are also analyzed in terms of stability, e.g. in [8]. However, these control approaches

2020 *Mathematics Subject Classification.* Primary: 70Q05, 93E35; Secondary: 70E60, 68T40.
Key words and phrases. tracking control, underactuated systems, learning, data-driven methods, safe control, formal methods.

*Corresponding author: Thomas Beckers.

depend on exact models of the systems and possible external disturbances to guarantee stability and precise tracking. An accurate model of typical uncertainties is hard to obtain by using first principles based techniques. Especially the impact of air/water flow on aerial/underwater vehicles or the interaction with unstructured and unknown environment further compound the uncertainty. The increase of the feedback gains to suppress the unknown dynamics is unfavorable due to the large errors in the presence of noise and the saturation of actuators.

A suitable approach to avoid the time-consuming or even unfeasible modeling process is provided by learning-based oracles such as Neural Networks or Gaussian processes (GPs). These data-driven modeling tools have shown remarkable results in many different applications including control, machine learning and system identification [9]. In data-driven control, data of the unknown system dynamics is collected and used by the oracle to predict the dynamics in areas without training data. In contrast to parametric models, data-driven models are highly flexible and are able to reproduce a large class of different dynamics, see [10].

The purpose of this article is to employ the power of learning-based approaches for the tracking control for a large class of underactuated systems. Additionally, stability and a desired level of performance of the closed-loop system should be guaranteed. The problem of tracking control of underactuated aerial/underwater vehicles with uncertainties has been addressed in [11, 12, 13] but these approaches are restricted to structured uncertainties such as uncertain parameters or use high feedback gains for compensation. Safe feedback linearization/backstepping based on GPs are introduced in [14, 15, 16] for a specific class of systems but they do not capture the general underactuated nature of the here considered mechanics or are limited to single input systems.

Learning-based approaches for Euler-Lagrange systems with stability guarantees are presented in [17, 18, 19]. However, the systems are required to be fully actuated. For a specific type of aerial vehicles, a safe Gaussian process based controller is proposed in [20] but with additional assumptions such as the existence of an initial safe controller.

The contribution of this article is a safe learning-based tracking control law for a large class of underactuated vehicles with partially unknown dynamics. The proposed control law guarantees the probabilistic boundedness of the tracking error and specifies the ultimate bound. Instead of focusing on a particular type of oracle, the proposed approach allows the usage of various learning-based oracles.

The remaining article is structured as follows: After the problem setting in Section 2, the learning-based oracles and the tracking controller are introduced in Section 3. In particular, in Section 3.1, we provide a probabilistic model error bound for the unknown dynamics and apply it in Section 3.2 to design a safe data-driven tracking control law based on a backstepping methodology. Finally, a numerical example is presented in Section 4.

2. Problem Setting.

We assume a single underactuated rigid body with position[1] $\boldsymbol{p} \in \mathbb{R}^3$ and orientation matrix $R \in SO(3)$. The body-fixed angular velocity is denoted by $\boldsymbol{\omega} \in \mathbb{R}^3$. The vehicle has mass $m \in \mathbb{R}_{>0}$ and rotational inertia $J \in \mathbb{R}^{3\times 3}$.

[1] Vectors are denoted with bold characters and matrices with capital letters. The term $A_{i,:}$ denotes the i-th row of the matrix A. The expression $\mathcal{N}(\mu, \Sigma)$ describes a normal distribution with mean μ and covariance Σ. The probability function is denoted by P. The set $\mathbb{R}_{>0}$ denotes the set of positive real numbers.

The state space of the vehicle is $S = SE(3) \times \mathbb{R}^6$ with

$$\boldsymbol{s} = ((R, \boldsymbol{p}), (\boldsymbol{\omega}, \dot{\boldsymbol{p}})) \in S$$

denoting the whole state of the system. The vehicle is actuated with control torques $\boldsymbol{\tau} \in \mathbb{R}^3$ and a control force $u \in \mathbb{R}$, which is applied in a body-fixed direction defined by a unit vector $\boldsymbol{e} \in \mathbb{R}^3$ as visualized in Fig. 1. We can model the system with the following set of differential equations

$$\begin{aligned} m\ddot{\boldsymbol{p}} &= R\boldsymbol{e}u + \boldsymbol{f}(\boldsymbol{p}, \dot{\boldsymbol{p}}), \\ \dot{R} &= R\check{\boldsymbol{\omega}}, \\ \dot{\boldsymbol{\omega}} &= J^{-1}\big(J\boldsymbol{\omega} \times \boldsymbol{\omega} + \boldsymbol{\tau} + \boldsymbol{f}_\omega(\boldsymbol{s})\big), \end{aligned} \tag{1}$$

where the hat map $\check{(\cdot)} \colon \mathbb{R}^3 \to \mathfrak{so}(3)$ is given by

$$\check{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_3 & -\omega_2 \\ \omega_3 & 0 & -\omega_1 \\ \omega_2 & \omega_1 & 0 \end{bmatrix}.$$

The functions $\boldsymbol{f} \colon \mathbb{R}^6 \to \mathbb{R}^3$ and $\boldsymbol{f}_\omega \colon S \to \mathbb{R}^3$ are state-dependent but time-invariant unknown dynamics. It is also assumed that the full state $\boldsymbol{s}$ can be measured. This assumption is very common and simply imposed for multi-rotor unmanned aerial vehicles or underwater vehicles. In both scenarios this assumption can be then fulfilled by design or by exploiting observer-based techniques, for instance, as in [21] for scenarios where no full state measurements are available. The general objective is to track a trajectory specified by the functions $(R_d, \boldsymbol{p}_d) \colon [0, T] \to SE(3)$. For simplicity, we focus here on position tracking only. The extension to rotation tracking is straightforward and will be discussed later.

2.1. **Equivalent system.** In preparation for the learning and control step, we transform the system dynamics (1) in an equivalent form. For the unknown dynamics $\boldsymbol{f}$ and $\boldsymbol{f}_\omega$, we use the estimates $\hat{\boldsymbol{f}} \colon \mathbb{R}^6 \to \mathbb{R}^3$ and $\hat{\boldsymbol{f}}_\omega \colon S \to \mathbb{R}^3$, respectively, of an oracle. The estimation error is moved to the error functions $\boldsymbol{\rho}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}(\boldsymbol{x})$ and $\boldsymbol{\rho}_\omega(\boldsymbol{s}) = \boldsymbol{f}_\omega(\boldsymbol{s}) - \hat{\boldsymbol{f}}_\omega(\boldsymbol{s})$, where $\boldsymbol{x} = [\boldsymbol{p}^\top, \dot{\boldsymbol{p}}^\top]^\top \in \mathbb{R}^6$, $\boldsymbol{s} \in S$. With the system matrix $A \in \mathbb{R}^{6\times 6}$ and input matrix $B \in \mathbb{R}^{6\times 3}$ given by

$$A = \begin{bmatrix} 0 & I_3 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{m} I_3 \end{bmatrix},$$

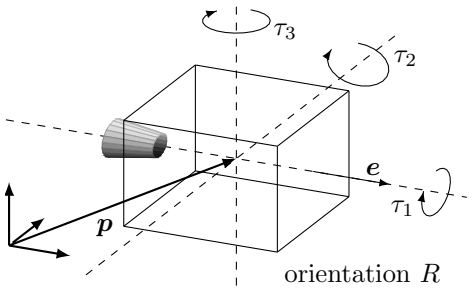

FIGURE 1. Underactuated vehicle with full attitude control and a translational force input. The position of the vehicle is described by the vector $\boldsymbol{p}$ and the orientation by the matrix R.

and $I_3 \in \mathbb{R}^{3\times 3}$ as identity matrix, we can rewrite (1) as

$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\big(\boldsymbol{g}(R,u) + \hat{\boldsymbol{f}}(\boldsymbol{x}) + \boldsymbol{\rho}(\boldsymbol{x})\big) \\ \dot{R} &= R\check{\boldsymbol{\omega}} \\ \dot{\boldsymbol{\omega}} &= J^{-1}\big(J\boldsymbol{\omega}\times\boldsymbol{\omega} + \boldsymbol{\tau} + \hat{\boldsymbol{f}}_\omega(\boldsymbol{s}) + \boldsymbol{\rho}_\omega(\boldsymbol{s})\big), \end{aligned} \tag{2}$$

where $\boldsymbol{g}\colon SO(3)\times\mathbb{R}\to\mathbb{R}^3$ is assumed as virtual control input with $g(R,u) \coloneqq R\boldsymbol{e}u$. As consequence, (2) is equivalent to (1).

3. Learning-based control.

In this section, we present the main result of the paper. First, in Section 3.1, we introduce the technical assumptions on the data set and the model error bounds. Next, we particularize to the situation when the oracle is specified by a Gaussian process model which allows us to provide a probabilistic model error bound for the unknown dynamics. Secondly, in Section 3.2, we design a data-driven tracking control law and provide safety guarantees by means of a probabilistic ultimate bound of the tracking error. Finally, we provide sufficient conditions to achieve asymptotic stability for the tracking error with high probability without any prediction error.

3.1. Learning.

For the compensation of the unknown dynamics of (1), we consider an oracle which predicts the values of $\boldsymbol{f}, \boldsymbol{f}_\omega$ for a given state $\boldsymbol{s}$. For this purpose, $N \in \mathbb{N}$ training points of the system (1) are collected such that a data set

$$\mathcal{D} = \{\boldsymbol{s}^{\{i\}}, \boldsymbol{y}^{\{i\}}\}_{i=1}^{N} \tag{3}$$

exists. The output data $\boldsymbol{y} \in \mathbb{R}^6$ are given by

$$\boldsymbol{y} = [(m\ddot{\boldsymbol{p}} - R\boldsymbol{e}u)^\top, (J\dot{\boldsymbol{\omega}} - J\boldsymbol{\omega}\times\boldsymbol{\omega}) - \boldsymbol{\tau})^\top]^\top$$

such that the first three components of $\boldsymbol{y}$ correspond to $\boldsymbol{f}$ and the remaining to $\boldsymbol{f}_\omega$. For the generation of the data set $\mathcal{D}$, the system (1) can be operated by an arbitrary controller. The only condition is that a finite sequence of training data of the system can be collected whereas stability is not necessarily required. The estimates of the oracle based on the data set $\mathcal{D}$ are denoted by $\hat{\boldsymbol{f}}(\boldsymbol{x})$ and $\hat{\boldsymbol{f}}_\omega(\boldsymbol{s})$.

Remark 1. Simple oracles can be parametric models such as a linear model, where the parameters are learned with a least-square approach based on the data set $\mathcal{D}$. More powerful oracles are given by neural networks, due to their universal function approximation property [22]. Furthermore, non-parametric oracles such as Gaussian processes and support vector machines have led to promising results as probabilistic function approximators [23, 24].

Remark 2. We focus here on state-dependent but time-invariant unknown dynamics $\boldsymbol{f}$,$\boldsymbol{f}_\omega$ such that the data set (3) is valid for all times. However, for the quasi-static case, where the change in the unknown dynamics due to time-varying elements is on a much longer time-scale than due to the state dependencies, the model can be held sufficiently accurate by retraining with a newly collected data set $\mathcal{D}$ after a decent time.

For the later stability analysis of the closed-loop, we introduce the following assumptions, which cover various types of oracles.

Assumption 1. Consider an oracle with the predictions $\hat{\boldsymbol{f}} \in \mathcal{C}^2$ and $\hat{\boldsymbol{f}}_\omega \in \mathcal{C}^0$ based on the data set $\mathcal{D}$. Let $S_\mathcal{X} \subset (SE(3) \times (\mathcal{X} \subset \mathbb{R}^6))$ be a compact set where the derivatives of $\hat{\boldsymbol{f}}$ are bounded on $\mathcal{X}$. There exists a bounded function $\bar{\rho}\colon S_\mathcal{X} \to \mathbb{R}_{\geq 0}$ such that the prediction error is given by

$$\mathrm{P}\left\{ \left\| \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}(\boldsymbol{x}) \\ \boldsymbol{f}_\omega(\boldsymbol{s}) - \hat{\boldsymbol{f}}_\omega(\boldsymbol{s}) \end{bmatrix} \right\| \leq \bar{\rho}(\boldsymbol{s}) \right\} \geq \delta$$

with probability $\delta \in (0, 1]$ for all $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{s} \in S_\mathcal{X}$.

Assumption 1 ensures that there exists an (probabilistic) upper bound for the error between the prediction $\hat{\boldsymbol{f}}(\boldsymbol{x}), \hat{\boldsymbol{f}}_\omega(\boldsymbol{s})$ and the actual $\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}_\omega(\boldsymbol{s})$ on a compact set. This assumption is fulfilled, for instance, by a Gaussian process model as oracle.

Gaussian process models have been proven as very powerful oracle for nonlinear function regression. For the prediction, we concatenate the N training points of $\mathcal{D}$ in an input matrix $X = [\boldsymbol{s}^1, \boldsymbol{s}^2, \dots, \boldsymbol{s}^N]$ and a matrix of outputs $Y^\top = [\boldsymbol{y}^1, \boldsymbol{y}^2, \dots, \boldsymbol{y}^N]$, where $\boldsymbol{y}$ might be corrupted by additive Gaussian noise with $\mathcal{N}(0, \sigma^2 I_6)$, $\sigma \in \mathbb{R}_{\geq 0}$. Then, a prediction for the output $\boldsymbol{y}^* \in \mathbb{R}^6$ at a new test point $\boldsymbol{s}^* \in S_\mathcal{X}$ is given by

$$\begin{aligned} \mu_i(\boldsymbol{y}^* | \boldsymbol{s}^*, \mathcal{D}) &= m_i(\boldsymbol{s}^*) + \boldsymbol{k}(\boldsymbol{s}^*, X)^\top K^{-1} \left(Y_{:,i} - [m_i(X_{:,1}), \dots, m_i(X_{:,N})]^\top \right) \\ \mathrm{var}_i(\boldsymbol{y}^* | \boldsymbol{s}^*, \mathcal{D}) &= k(\boldsymbol{s}^*, \boldsymbol{s}^*) - \boldsymbol{k}(\boldsymbol{s}^*, X)^\top K^{-1} \boldsymbol{k}(\boldsymbol{s}^*, X). \end{aligned} \tag{4}$$

for all $i \in \{1, \dots, N\}$, where $Y_{:,i}$ denotes the i-th column of the matrix of outputs Y. The kernel $k\colon S_\mathcal{X} \times S_\mathcal{X} \to \mathbb{R}$ is a measure for the correlation of two states $(\boldsymbol{s}, \boldsymbol{s}')$, whereas the mean function $m_i\colon S_\mathcal{X} \to \mathbb{R}$ allows to include prior knowledge. The function $K\colon S_\mathcal{X}^N \times S_\mathcal{X}^N \to \mathbb{R}^{N \times N}$ is called the Gram matrix whose elements are given by $K_{j',j} = k(X_{:,j'}, X_{:,j}) + \delta(j, j')\sigma^2$ for all $j', j \in \{1, \dots, N\}$ with the delta function $\delta(j, j') = 1$ for $j = j'$ and zero, otherwise. The vector-valued function $\boldsymbol{k}\colon S_\mathcal{X} \times S_\mathcal{X}^N \to \mathbb{R}^N$, with the elements $k_j = k(\boldsymbol{s}^*, X_{:,j})$ for all $j \in \{1, \dots, N\}$, expresses the covariance between $\boldsymbol{s}^*$ and the input training data X. The selection of the kernel and the determination of the corresponding hyperparameters can be seen as degrees of freedom of the regression. A powerful kernel for GP models of physical systems is the squared exponential kernel. An overview about the properties of different kernels can be found in [23].

Remark 3. The mean function can be achieved by common system identification techniques of the unknown dynamics $\boldsymbol{f}, \boldsymbol{f}_\omega$ as described in [25]. However, without any prior knowledge the mean function is set to zero, i.e. $m_i(\boldsymbol{s}) = 0$.

Based on (4), the normal distributed components $y_i^* | \boldsymbol{s}^*, \mathcal{D}$ are combined into a multi-variable distribution $\boldsymbol{y}^* | (\boldsymbol{s}^*, \mathcal{D}) \sim \mathcal{N}(\boldsymbol{\mu}(\cdot), \Sigma(\cdot))$, where

$$\boldsymbol{\mu}(\boldsymbol{y}^* | \boldsymbol{s}^*, \mathcal{D}) = [\mu_1(\cdot), \dots, \mu_6(\cdot)]^\top, \quad \Sigma(\boldsymbol{y}^* | \boldsymbol{s}^*, \mathcal{D}) = \mathrm{diag}\left[\mathrm{var}_1(\cdot), \dots, \mathrm{var}_6(\cdot)\right].$$

Remark 4. For simplicity, we consider identical kernels for each output dimension. However, the GP model can be easily adapted to different kernels for each output dimension.

With the introduced Gaussian process model, we are now addressing Assumption 1 using [24, 17, 26]. To provide model error bounds, additional assumptions on the unknown parts of the dynamics (1) must be introduced, in line with the no-free-lunch theorem, see [27].

Assumption 2. The kernel k is selected such that $\boldsymbol{f}, \boldsymbol{f}_\omega$ have a bounded reproducing kernel Hilbert space (RKHS) norm on $\mathcal{X}$ and $S_\mathcal{X}$, respectively, in detail, $\|f_i\|_k < \infty$ and $\|f_{\omega,i}\|_k < \infty$ for all $i = 1, 2, 3$.

The norm of a function in a RKHS is a smoothness measure relative to a kernel k that is uniquely connected with this RKHS. In particular, it is a Lipschitz constant with respect to the metric of the used kernel. A more detailed discussion about RKHS norms is given in [28]. Assumption 2 requires that the kernel must be selected in such a way that the functions $\boldsymbol{f}, \boldsymbol{f}_\omega$ are elements of the associated Reproducing kernel Hilbert space. This sounds paradoxical since this function is unknown. However, there exist some kernels, namely universal kernels, which can approximate any continuous function arbitrarily precisely on a compact set, see [24, Lemma 4.55], such that the bounded RKHS norm is a mild assumption. Finally, with Assumption 2, the residual model error can be bounded as written in the following lemma.

Lemma 3.1 (adapted from [17])**.** *Consider the unknown functions $\boldsymbol{f}, \boldsymbol{f}_\omega$ and a GP model satisfying Assumption 2. The model error is bounded by*

$$\mathrm{P}\left\{\left\| \boldsymbol{\mu}\left(\begin{bmatrix} \hat{\boldsymbol{f}}(\boldsymbol{x}) \\ \hat{\boldsymbol{f}}_\omega(\boldsymbol{s}) \end{bmatrix} \middle| \boldsymbol{s}, \mathcal{D} \right) - \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}) \\ \boldsymbol{f}_\omega(\boldsymbol{s}) \end{bmatrix} \right\| \leq \left\| \boldsymbol{\beta}^\top \Sigma^{\frac{1}{2}} \left(\begin{bmatrix} \hat{\boldsymbol{f}}(\boldsymbol{x}) \\ \hat{\boldsymbol{f}}_\omega(\boldsymbol{s}) \end{bmatrix} \middle| \boldsymbol{s}, \mathcal{D} \right) \right\| \right\} \geq \delta$$

for $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{s} \in S_\mathcal{X}, \delta \in (0, 1)$ with $\boldsymbol{\beta} \in \mathbb{R}^6$ given by [17, Lemma 1]

Proof. It is a direct implication of [17, Lemma 1]. □

With Assumption 2 and the fact, that universals kernels exist which generate bounded predictions with bounded derivatives, see [26], GP models can be used as oracle to fulfill Assumption 1. In this case, the prediction error bound is given by $\bar{\rho}(\boldsymbol{s}) := \|\boldsymbol{\beta}^\top \Sigma^{\frac{1}{2}}([\hat{\boldsymbol{f}}(\boldsymbol{x})^\top, \hat{\boldsymbol{f}}_\omega(\boldsymbol{s})^\top]^\top | \boldsymbol{s}, \mathcal{D})\|$ as shown in Lemma 3.1.

Remark 5. An efficient algorithm can be used to find $\boldsymbol{\beta}$ based on the maximum information gain [29].

3.2. **Tracking control.** For the tracking control, we consider a given desired trajectory $\boldsymbol{x}_d(t) \colon \mathbb{R}_{t \geq 0} \to \mathcal{X}, \boldsymbol{x}_d \in \mathcal{C}^4$. The tracking error is denoted by $\boldsymbol{z}_0(t) = \boldsymbol{x}(t) - \boldsymbol{x}_d(t)$. Before we propose the main theorem about the *safe learning-based tracking control law*, the feedback gain matrix G is introduced. As part of the controller, G penalizes the position tracking error and the result is fed back to both inputs, the force control u and the torque control $\boldsymbol{\tau}$ of the system (1).

Property 1. The matrix $G \in \mathbb{R}^{3\times 6}$ is chosen such that there exist a positive definite matrix $P \in \mathbb{R}^{6\times 6}$ and a positive definite symmetric matrix $Q \in \mathbb{R}^{6\times 6}$ which satisfy the Lyapunov equation

$$P\big(A - BG\big) + \big(A - BG\big)^\top P = -Q.$$

Property 1 is satisfied if the real parts of all eigenvalues of $(A - BG)$ are negative. For example, this can be achieved by any $G = [G_1, G_2]$, where $G_1, G_2 \in \mathbb{R}^{3\times 3}$ are positive definite diagonal matrices.

Theorem 3.2. *Consider the underactuated rigid-body system given by (1) with unknown dynamics $\boldsymbol{f}, \boldsymbol{f}_\omega$ and the existence of an oracle satisfying Assumption 1.*

Let $G_{z_1}, G_{z_2} \in \mathbb{R}^{3\times 3}$ be positive definite symmetric matrices. Assuming Property 1 is satisfied, the control law

$$\begin{aligned} \boldsymbol{\tau} &= J(\boldsymbol{e} \times (R^\top \boldsymbol{g}_{\ddot{d}} - \check{\boldsymbol{\omega}}^2 \boldsymbol{e} u - 2\check{\boldsymbol{\omega}}\boldsymbol{e}\dot{u})u^{-1}) - J\boldsymbol{\omega} \times \boldsymbol{\omega} - \hat{\boldsymbol{f}}_\omega(\boldsymbol{s}), \\ \ddot{u} &= \boldsymbol{e}^\top (R^\top \boldsymbol{g}_{\ddot{d}} - \check{\boldsymbol{\omega}}^2 \boldsymbol{e} u - 2\check{\boldsymbol{\omega}}\boldsymbol{e}\dot{u}), \end{aligned} \tag{5}$$

with the desired virtual control input derivative

$$\begin{aligned} \boldsymbol{g}_{\ddot{d}} = {} & m p_d^{(4)} - G\left(\frac{\partial \dot{\hat{\boldsymbol{x}}}}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}} - \ddot{\boldsymbol{x}}_d\right) - BP(\dot{\hat{\boldsymbol{x}}} - \dot{\boldsymbol{x}}_d) \\ & - (G_{z_1} + G_{z_2})\left(\dot{\boldsymbol{g}} - m p_d^{(3)} + G(\dot{\hat{\boldsymbol{x}}} - \dot{\boldsymbol{x}}_d) + \frac{\partial \hat{\boldsymbol{f}}}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}}\right) \\ & - (G_{z_2}G_{z_1} + I_3)\left(\boldsymbol{g} - m\ddot{\boldsymbol{p}}_d + G\boldsymbol{z}_0 + \hat{\boldsymbol{f}}(\boldsymbol{x})\right) - G_{z_2}B^\top P \boldsymbol{z}_0 - \frac{\partial}{\partial \boldsymbol{x}}\left[\frac{\partial \hat{\boldsymbol{f}}}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}}\right]\dot{\hat{\boldsymbol{x}}}. \end{aligned} \tag{6}$$

$$\dot{\hat{\boldsymbol{x}}} = A\boldsymbol{x} + B\left(\boldsymbol{g}(R,u) + \hat{\boldsymbol{f}}(\boldsymbol{x})\right) \tag{7}$$

guarantees that the tracking error is uniformly ultimately bounded in probability by

$$\mathrm{P}\left\{\|\boldsymbol{z}_0(t)\| \leq \max_{\boldsymbol{s}\in S_{\mathcal{X}}} \bar{\rho}(\boldsymbol{s})\lambda\sqrt{\frac{\max\{\mathrm{eig}(P),1\}}{\min\{\mathrm{eig}(P),1\}}}, \forall t \geq T\right\} \geq \delta$$

with constants $\lambda, T \in \mathbb{R}_{\geq 0}$ on $S_{\mathcal{X}}$.

Remark 6. The control law does not depend on any state derivatives, which are typically noisy in measurements. The derivatives, i.e. the translational and angular accelerations, are only necessary for the training of the oracle, see (3), which can often deal with noisy data. For instance, GP models can handle additive Gaussian noise on the output [23].

Remark 7. The torque control law of (5) is only applicable if $u \neq 0$, as otherwise no tracking control is possible in general. A reasonable trajectory planning can avoid a zero force input, see [11].

We prove the boundedness of the tracking error by the step-wise construction of a Lyapunov function using backstepping methods following the construction proposed in [12] for the nominal case.

Proof. The term $\boldsymbol{g}(R,\boldsymbol{u})$ in (2) is assumed as virtual control input with the desired force

$$\boldsymbol{g}_d(t,\boldsymbol{x}) = m\ddot{\boldsymbol{p}}_d - G\boldsymbol{z}_0 - \hat{\boldsymbol{f}}(\boldsymbol{x}), \tag{8}$$

where G is the feedback gain matrix. The tracking error dynamics are given by

$$\dot{\boldsymbol{z}}_0 = A\boldsymbol{x} + B\left(\boldsymbol{g}(R,u) + \hat{\boldsymbol{f}}(\boldsymbol{x}) + \boldsymbol{\rho}(\boldsymbol{x})\right) - \begin{bmatrix}\dot{\boldsymbol{p}}_d \\ \ddot{\boldsymbol{p}}_d\end{bmatrix}. \tag{9}$$

Using the desired acceleration $\ddot{\boldsymbol{p}}_d$ of (8) in (9) leads to

$$\dot{\boldsymbol{z}}_0 = \big(A - BG\big)\boldsymbol{z}_0 + B\big(\boldsymbol{g}(R,u) - \boldsymbol{g}_d(t,\boldsymbol{x}) + \boldsymbol{\rho}(\boldsymbol{x})\big).$$

In the next step, the boundedness of the tracking error $\boldsymbol{z}_0$ is proven. For this purpose, we use the matrices P, Q of Property 1 to construct the Lyapunov function $V_0(\boldsymbol{z}_0) = \frac{1}{2}\boldsymbol{z}_0^\top P \boldsymbol{z}_0 \geq 0$ and compute its evolution

$$\dot{V}_0 = -\frac{1}{2}\boldsymbol{z}_0^\top Q \boldsymbol{z}_0 + (B^\top P \boldsymbol{z}_0)^\top \big(\boldsymbol{g}(R,u) - \boldsymbol{g}_d + \boldsymbol{\rho}(\boldsymbol{x})\big).$$

The first summand is negative for all $\boldsymbol{z}_0 \in \mathbb{R}^6$. Next, we extend the previous Lyapunov function with the error term $\boldsymbol{z}_1 \in \mathbb{R}^3$ given by

$$\boldsymbol{z}_1(t, \boldsymbol{x}, R, u) = \boldsymbol{g}(R, u) - \boldsymbol{g}_d(t, \boldsymbol{x}), \tag{10}$$

which describes the error between the virtual and the desired control input. For this purpose, a quadratic Lyapunov function $V_1(\boldsymbol{z}_0, \boldsymbol{z}_1) = \frac{1}{2}\boldsymbol{z}_1^\top \boldsymbol{z}_1 + V_0 \geq 0$ is considered. The derivative of V_1 leads to

$$\dot{V}_1 = \dot{V}_0 + \boldsymbol{z}_1^\top \left(\dot{\boldsymbol{g}} - m\boldsymbol{p}_d^{(3)} + G\dot{\boldsymbol{z}}_0 + \dot{\hat{\boldsymbol{f}}}(\boldsymbol{x}) \right), \tag{11}$$

where $\boldsymbol{p}_d^{(3)}$ denotes the third time-derivative of the desired position $\boldsymbol{p}_d$, and where we have used (10) and (8). As in (8), following again the idea of a desired virtual input proposed for the nominal case, see [12], we construct a desired value of $\dot{\boldsymbol{g}}$ with

$$\boldsymbol{g}_{\dot{d}} = m\boldsymbol{p}_d^{(3)} - G(\dot{\hat{\boldsymbol{x}}} - \dot{\boldsymbol{x}}_d) - B^\top P\boldsymbol{z}_0 - G_{z_1}\boldsymbol{z}_1 - \frac{\partial \hat{\boldsymbol{f}}}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}}. \tag{12}$$

Instead of having dependencies on the typical noisy state derivative $\dot{\boldsymbol{x}}$, we use the estimation $\dot{\hat{\boldsymbol{x}}} \in \mathbb{R}^6$ given by

$$\dot{\hat{\boldsymbol{x}}} = A\boldsymbol{x} + B\left(\boldsymbol{g}(R, u) + \hat{\boldsymbol{f}}(\boldsymbol{x}) \right),$$

which only contains the known parts of the system dynamics (2). Then, the expression (12) is used to substitute $\dot{\boldsymbol{g}}$ in (11). This last substitution leads to the evolution

$$\dot{V}_1 = -\frac{1}{2}\boldsymbol{z}_0^\top Q\boldsymbol{z}_0 - \boldsymbol{z}_1^\top G_{z_1}\boldsymbol{z}_1 + \boldsymbol{z}_0^\top PB\boldsymbol{\rho}(\boldsymbol{x}) + \boldsymbol{z}_1^\top \left(\left[\frac{\partial \hat{\boldsymbol{f}}}{\partial \boldsymbol{x}} + G \right] B\boldsymbol{\rho}(\boldsymbol{x}) + \dot{\boldsymbol{g}} - \boldsymbol{g}_{\dot{d}} \right).$$

Next, we define the error $\boldsymbol{z}_2 \in \mathbb{R}^3$ with $\boldsymbol{z}_2(t, \boldsymbol{x}, R, u) = \dot{\boldsymbol{g}}(R, u) - \boldsymbol{g}_{\dot{d}}(t, \boldsymbol{x}, R, u)$, and an extended Lyapunov function

$$V(\boldsymbol{z}_0, \boldsymbol{z}_1, \boldsymbol{z}_2) = V_1 + \frac{1}{2}\boldsymbol{z}_2^\top \boldsymbol{z}_2 \geq 0. \tag{13}$$

The derivative of V leads to

$$\dot{V} = \dot{V}_1 + \boldsymbol{z}_2^\top \left(\ddot{\boldsymbol{g}} - mp_d^{(4)} + G\ddot{\boldsymbol{z}}_0 + BP\dot{\boldsymbol{z}}_0 + \frac{d}{dt}\left[\frac{\partial \hat{\boldsymbol{f}}}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}} \right] \right)$$

and we construct a desired value of $\ddot{\boldsymbol{g}}$ with $\boldsymbol{g}_{\ddot{d}}$ given by

$$\begin{aligned}
\boldsymbol{g}_{\ddot{d}} = {} & mp_d^{(4)} - G\left(\frac{\partial \dot{\hat{\boldsymbol{x}}}}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}} - \ddot{\boldsymbol{x}}_d \right) - BP(\dot{\hat{\boldsymbol{x}}} - \dot{\boldsymbol{x}}_d) \\
& - (G_{z_1} + G_{z_2}) \left(\dot{\boldsymbol{g}} - mp_d^{(3)} + G(\dot{\hat{\boldsymbol{x}}} - \dot{\boldsymbol{x}}_d) + \frac{\partial \hat{\boldsymbol{f}}}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}} \right) \\
& - (G_{z_2}G_{z_1} + I_3) \left(\boldsymbol{g} - m\ddot{\boldsymbol{p}}_d + G\boldsymbol{z}_0 + \hat{\boldsymbol{f}}(\boldsymbol{x}) \right) - G_{z_2}B^\top P\boldsymbol{z}_0 - \frac{\partial}{\partial \boldsymbol{x}}\left[\frac{\partial \hat{\boldsymbol{f}}}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}} \right]\dot{\hat{\boldsymbol{x}}}.
\end{aligned}$$

Then, it is substituted into $\dot{V}$ to obtain

$$\begin{aligned}
\dot{V} = {} & -\frac{1}{2}\boldsymbol{z}_0^\top Q\boldsymbol{z}_0 - \boldsymbol{z}_1^\top G_{z_1}\boldsymbol{z}_1 - \boldsymbol{z}_2^\top G_{z_2}\boldsymbol{z}_2 \\
& + (\boldsymbol{z}_0^\top P + \boldsymbol{z}_1^\top D(\boldsymbol{x}) + \boldsymbol{z}_2^\top E(\boldsymbol{x}))B\boldsymbol{\rho}(\boldsymbol{x}) + \boldsymbol{z}_2^\top(\ddot{\boldsymbol{g}} - \boldsymbol{g}_{\ddot{d}}), \qquad (14) \\
D(\boldsymbol{x}) \coloneqq {} & \frac{\partial \hat{\boldsymbol{f}}}{\partial \boldsymbol{x}} + G,
\end{aligned}$$

$$E(\boldsymbol{x}) \coloneqq BP + G_{z_1} D + G\frac{\partial \dot{\hat{\boldsymbol{x}}}}{\partial \boldsymbol{x}} + \frac{\partial}{\partial \boldsymbol{x}}\Big(\frac{\partial \hat{\boldsymbol{f}}}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}}\Big).$$

To eliminate the last summand in (14) except for the estimation error $\boldsymbol{\rho}_\omega$, we note that

$$\ddot{\boldsymbol{g}}(R,u) = R(\check{\boldsymbol{\omega}}^2 \boldsymbol{e}u + 2\check{\boldsymbol{\omega}}\boldsymbol{e}\dot{u} + \dot{\boldsymbol{\omega}} \times \boldsymbol{e}u + \boldsymbol{e}\ddot{u}),$$

such that $\ddot{\boldsymbol{g}} - \boldsymbol{g}_{\ddot{d}} = -RJ^{-1}\boldsymbol{\rho}_\omega(\boldsymbol{s}) \times \boldsymbol{g}$ for

$$\dot{\boldsymbol{\omega}} \times \boldsymbol{e}u + \boldsymbol{e}\ddot{u} = R^\top \boldsymbol{g}_{\ddot{d}} - \check{\boldsymbol{\omega}}^2 \boldsymbol{e}u - 2\check{\boldsymbol{\omega}}\boldsymbol{e}\dot{u}. \tag{15}$$

Using (15) and Assumption 1, the evolution of the Lyapunov function V can be upper bounded by

$$\begin{aligned} \mathrm{P}\{\dot{V} \leq -\begin{bmatrix}\boldsymbol{z}_0^\top, \boldsymbol{z}_1^\top, \boldsymbol{z}_2^\top\end{bmatrix} \begin{bmatrix} Q & 0 & 0 \\ 0 & G_{z_1} & 0 \\ 0 & 0 & G_{z_2}\end{bmatrix} \begin{bmatrix}\boldsymbol{z}_0 \\ \boldsymbol{z}_1 \\ \boldsymbol{z}_2\end{bmatrix} \\ + \big\|(\boldsymbol{z}_0^\top P + \boldsymbol{z}_1^\top \bar{D} + \boldsymbol{z}_2^\top \bar{E})B + \bar{c}\boldsymbol{z}_2\big\|\bar{\rho}(\boldsymbol{s})\} \geq \delta, \end{aligned} \tag{16}$$

with the upper bounds $\bar{D}, \bar{E} \in \mathbb{R}^{3\times 3}$ and $\bar{c} \in \mathbb{R}$, which exist due to Assumption 1. Thus, the evolution is negative with probability δ for all $\boldsymbol{z}$ with

$$\|\boldsymbol{z}\| > \max_{\boldsymbol{s}\in S_{\mathcal{X}}} \bar{\rho}(\boldsymbol{s}) \underbrace{\frac{\|PB\| + \big\|\bar{D}B\big\| + \big\|\bar{E}B\big\| + \bar{c}}{\min\{\mathrm{eig}(Q), \mathrm{eig}(G_{z_1}), \mathrm{eig}(G_{z_2})\}}}_{=:\lambda}$$

where a maximum of $\bar{\rho}$ exists regarding to Assumption 1. Finally, the Lyapunov function (13) is lower and upper bounded by $\alpha_1(\|\boldsymbol{z}\|) \leq V(\boldsymbol{z}) \leq \alpha_2(\|\boldsymbol{z}\|)$, where $\alpha_1(r) = \frac{1}{2}\min\{\mathrm{eig}(P), 1\}r^2$ and $\alpha_2(r) = \frac{1}{2}\max\{\mathrm{eig}(P), 1\}r^2$. Thus, we can compute the radius $b \in \mathbb{R}_{\geq_0}$ of the bound by

$$b = \max_{\boldsymbol{s}\in S_{\mathcal{X}}} \bar{\rho}(\boldsymbol{s})\lambda\sqrt{\frac{\max\{\mathrm{eig}(P), 1\}}{\min\{\mathrm{eig}(P), 1\}}}. \tag{17}$$

Hence, the tracking error of the closed-loop is uniformly ultimately bounded in probability by $\mathrm{P}\{\|\boldsymbol{z}_0(t)\| \leq b, \forall t \geq T\} \geq \delta$. □

Remark 8. Extension to the attitude are analogously to perform with additional terms in the Lyapunov function as given in [11, 8].

Note that the proof of Theorem 3.2 shows that the bound of the tracking error (17) depends on the prediction error $\bar{\rho}$ of the oracle. Depending on prior knowledge about the unknown functions $\boldsymbol{f}, \boldsymbol{f}_\omega$ and the oracle used, the prediction error can vanish which leads to asymptotic stability of the tracking error. In order to achieve this, we introduce the following assumption.

Assumption 3. Let be the data set $\mathcal{D}$, such that the model error of the oracle is bounded by

$$\mathrm{P}\left\{\left\|\begin{bmatrix}\boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}(\boldsymbol{x}) \\ \boldsymbol{f}_\omega(\boldsymbol{s}) - \hat{\boldsymbol{f}}(\boldsymbol{s})\end{bmatrix}\right\| = 0\right\} \geq \delta. \tag{18}$$

Simply speaking, the oracle must be able to reproduce the unknown dynamics with a certain probability without any prediction error. Even though this seems to be a strong assumption, there exist these types of oracles if additional prior knowledge about the unknown $\boldsymbol{f}, \boldsymbol{f}_\omega$ is available, see [23] and [24] for instance, as we explain in the below.

Remark 9. With a GP model as oracle, (18) is satisfied if the posterior variance $\Sigma([\hat{\boldsymbol{f}}(\boldsymbol{x})^\top, \hat{\boldsymbol{f}}_\omega(\boldsymbol{s})^\top]^\top | \boldsymbol{s}, \mathcal{D})$ is zero on $S_\mathcal{X}$, as shown in Lemma 3.1, for the data set $\mathcal{D}$. If the kernel function of the GP has a finite dimensional feature space, the posterior variance vanishes for a finite number of distinct, noise-free data points, see [30] for further details. A finite dimensional feature space is given, for instance, by the linear or the polynomial kernel, see [23].

With (18), asymptotic stability of the tracking error is achieved which is formally written in the next corollary.

Corollary 1. *Consider the underactuated rigid-body system given by (1) with unknown dynamics $\boldsymbol{f}, \boldsymbol{f}_\omega$ and the existence of an oracle satisfying Assumption 1, Property 1, and (18). Then, the control law (5) renders the tracking error $\boldsymbol{z}_0$ asymptotically stable on $S_\mathcal{X}$ with probability δ.*

Proof. Using the result about the upper bound of the Lyapunov derivative given by (16) together with (18), i.e. $\bar{\rho}(\boldsymbol{s}) = 0$, leads to

$$\mathrm{P}\left\{ \dot{V} \leq -\begin{bmatrix}\boldsymbol{z}_0^\top, \boldsymbol{z}_1^\top, \boldsymbol{z}_2^\top\end{bmatrix} \begin{bmatrix} Q & 0 & 0 \\ 0 & G_{z_1} & 0 \\ 0 & 0 & G_{z_2} \end{bmatrix} \begin{bmatrix}\boldsymbol{z}_0 \\ \boldsymbol{z}_1 \\ \boldsymbol{z}_2\end{bmatrix} \right\} \geq \delta, \tag{19}$$

such that $\dot{V}$ is always negative on $S_\mathcal{X}$ with probability δ. As consequence, the tracking error is asymptotically stable on $S_\mathcal{X}$ with probability δ. □

4. Numerical example.

To demonstrate the application relevance of our proposed approach, we consider the task of an quadcopter to explore a terrain with unknown thermals. The dynamics of the vehicle are described by (1) with mass $m = 1\,\mathrm{kg}$, inertia $J = I_3 \mathrm{kgm}^2$ and the direction $\boldsymbol{e} = [0, 0, 1]^\top$ of the force input u. The data of the thermals is taken from publicly available paragliding data[2]. The thermals are assumed to act on the quadcopter as a disturbance in the direction of x_3, i.e., the altitude, as well as an angular momentum in the direction of ω_1.

A GP model is then used as oracle to predict $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{f}_\omega(\boldsymbol{s})$ based on the collected data set with the squared exponential kernel, see [23]. The prior knowledge about the existing gravity in $\boldsymbol{f}(\boldsymbol{x})$ is considered as estimate in the mean function of the GP with $m_3(\boldsymbol{s}) = -10$. First, we start with the collection of training data for the GP model. For this purpose, the control inputs for the aerial vehicle are generated by a controller as proposed in Theorem 3.2 but without an oracle, i.e. $\hat{\boldsymbol{f}}(\boldsymbol{x}) = \hat{\boldsymbol{f}}_\omega(\boldsymbol{s}) = \boldsymbol{0}, \forall \boldsymbol{x} \in \mathbb{R}^6, \boldsymbol{s} \in \mathcal{S}$. The feedback gain matrix is set to

$$G = \begin{bmatrix} 10 & 0 & 0 & 10 & 0 & 0 \\ 0 & 10 & 0 & 0 & 10 & 0 \\ 0 & 0 & 20 & 0 & 0 & 10 \end{bmatrix}$$

and $G_{z_1} = G_{z_2} = 2I_3$. The desired trajectory is given by $x_{d,1}(t) = \sin(t), x_{d,2}(t) = \cos(t) - 1, x_{d,3}(t) = t/10$. Figure 2 visualizes the normalized magnitude of the thermal updraft which is assumed to be unknown. Every 0.1 s a training point has been recorded. Each training point consists of the actual state $\boldsymbol{s}$ and $\boldsymbol{y}$ as given by (3). Since the training points depend on the typically noisy measurement of the accelerations $\ddot{\boldsymbol{p}}$ and $\dot{\boldsymbol{\omega}}$, a Gaussian distributed noise $\mathcal{N}(0, 0.08^2 I_3)$ is added to the measurement. After the simulation time of 15 s, the data set $\mathcal{D}$ consists of 150

[2] https://thermal.kk7.ch/

training points. Based on this data set, a GP model is trained and the hyperparameters are optimized by means of the likelihood function, see [24], for instance.

Figure 3 shows the tracking error of the quadcopter for the training step, i.e, without oracle (blue) and for the proposed learning-based control law (red). Due to the impact of $\boldsymbol{f}$ and $\boldsymbol{f}_\omega$, a large tracking error occurs without an accurate model. The trained oracle allows us to significantly decrease the tracking error and to let it converges to a tight set around zero which is also captured in Fig. 4 by the bounded Lyapunov function. The Lyapunov function might be non-decreasing in some time sections after it enters this set which results in the "humps" that can be observed

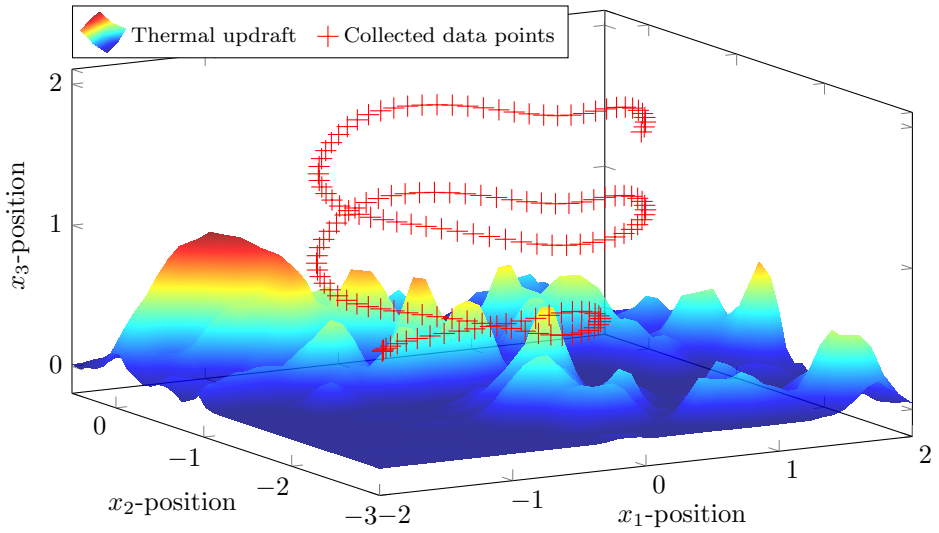

FIGURE 2. Visualization of the normalized magnitude of the thermal updraft acting on the quadcopter and the recorded training data points (red crosses).

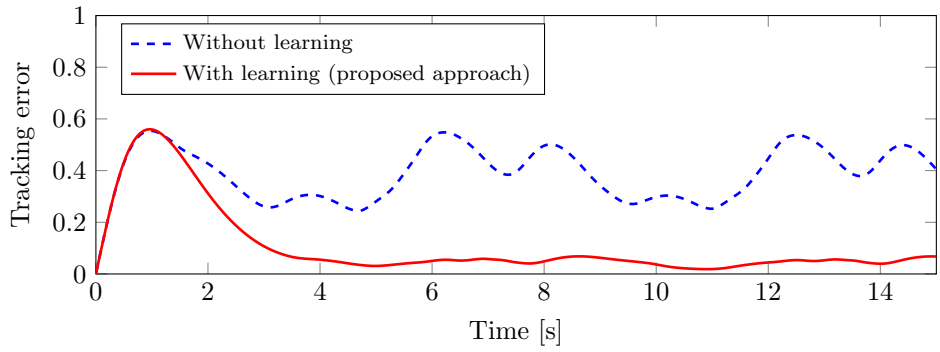

FIGURE 3. Tracking error of the quadcopter with control law (5) without learning (blue) and with our proposed learning-based approach (red).

in Fig. 4. However, as guaranteed by Theorem 3.2, the Lyapunov function stays in this set after the initial entry for all time with high probability. The size of this set can be further reduced by improving the accuracy of the model with more training data. The corresponding control inputs generated by the learning-based control law (5) are depicted in Fig. 5. Figure 6 shows the prediction of the GP model (solid) and the ground truth (dashed) of the unknown functions $\boldsymbol{f}$ and $\boldsymbol{f}_\omega$. Even though these functions are highly nonlinear over time, the GP model can approximate the dynamics accurately which leads to the reduced tracking error.

Finally, we test the learning-base control law on a set of randomly generated desired trajectories. For this purpose, we generate 30 trajectories, each created by the sampling of a start point, an endpoint and 3 waypoints from a uniform distribution over $[-1.5, 1.5] \times [-2.5, 0] \times [0, 1]$. The full trajectory is then generated via quintic polynomials [31]. The left plot of Fig. 7 shows three examples of these randomly generated trajectories. The set of 30 trajectories is split in a training set consisting of i) 5 trajectories (375 training points) and ii) 10 trajectories (750 training points) as well as 20 test trajectories. A GP model is trained based on the two training sets as previously described. The evaluation of the L^2-norm of the

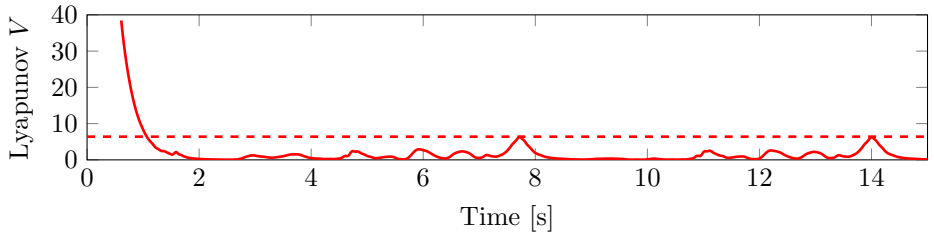

FIGURE 4. Lyapunov function (solid) converges to a tight set around zero (dashed line) and stays inside this set with high probability as guaranteed by Theorem 3.2.

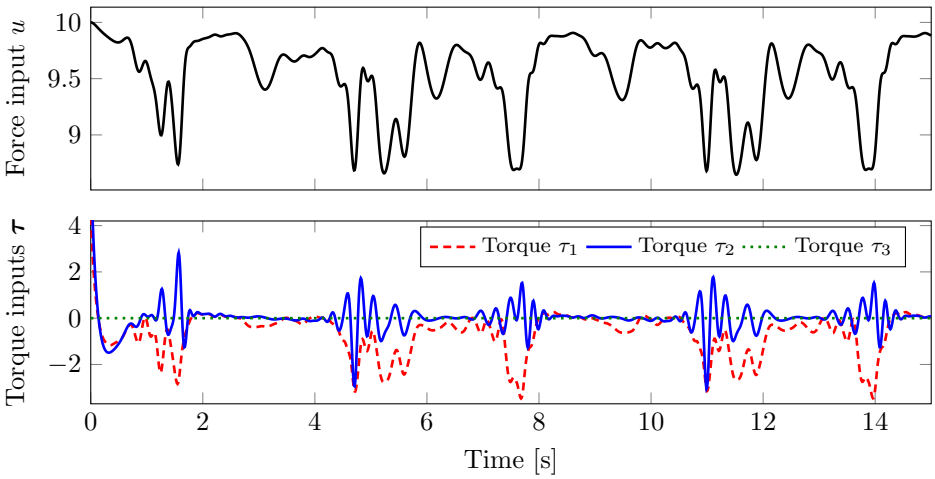

FIGURE 5. Control inputs for the quadcopter. Top: Control input for the thrust u. Bottom: Control input for the torques τ_1, τ_2, τ_3.

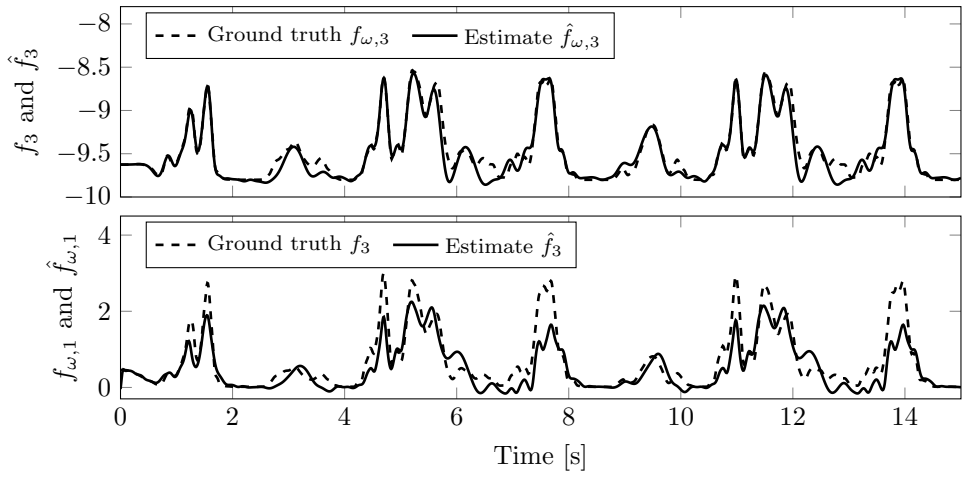

FIGURE 6. Ground truth (dashed) of the unknown dynamics and estimates of the GP (solid). Top: Unknown dynamics acting on the x_3-position of the quadcopter. Bottom: Unknown dynamics acting on the first component of the angular acceleration $\dot{\boldsymbol{\omega}}$ of the quadcopter.

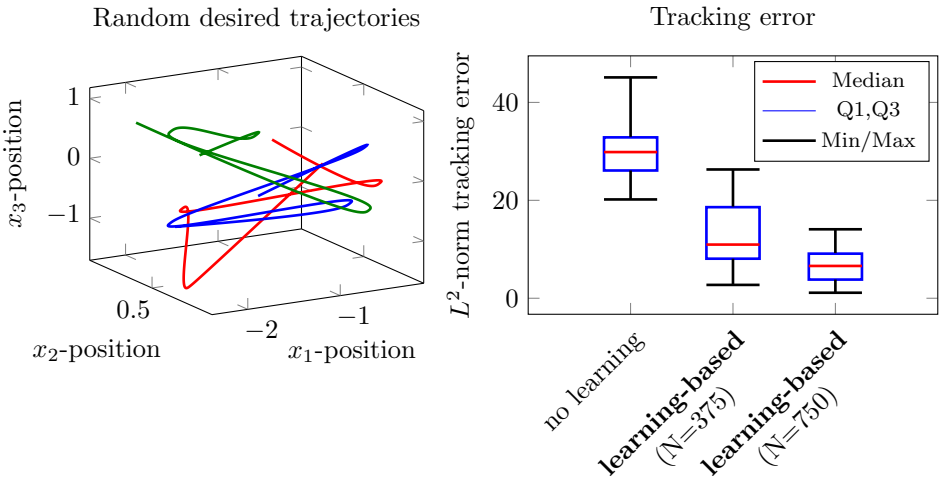

FIGURE 7. Left: Three exemplary trajectories of the 30 randomly generated trajectories for the training and test set. Right: L^2-norm of the tracking error of the test set for the standard control law (left box), the proposed learning-based control law with 375 training points (middle box) and the proposed learning-based control with 750 training points (right box).

tracking error based for the test set is visualized in the right plot of Fig. 7. The first boxplot shows the performance of the standard control law without learning. The second boxplot represents the performance for the proposed learning-based control law with a training set of 375 state measurements which significantly reduces the tracking error. The performance can be further improved by collecting more data as shown in the most right boxplot for a training set of 750 state measurements.

Conclusion. We present a safe learning-based tracking control law for a class of underactuated systems with unknown dynamics typical for aerial and underwater vehicles. Using a various type of oracles, the tracking error is proven to be bounded in probability and the size of the bound is explicitly given. Furthermore, additional assumptions lead to asymptotic stability. Even though no particular oracle is assumed, we show that Gaussian process models fulfill all requirements to be used as oracle in the proposed control scheme. Finally, a numerical example visualizes the effectiveness of the control law.

Acknowledgments. This work was supported by the European Research Council (ERC) Consolidator Grant "Safe data-driven control for human-centric systems (COMAN)" under grant agreement number 864686, by the European Commission project "SeaClear" under grant agreement number 871295, and by a 2020 Leonardo Grant for Researchers and Cultural Creators, BBVA Foundation.

REFERENCES

[1] M. Reyhanoglu, A. van der Schaft, N. H. McClamroch, and I. Kolmanovsky, "Dynamics and control of a class of underactuated mechanical systems," *IEEE Transactions on Automatic Control*, vol. 44, no. 9, pp. 1663–1671, 1999.

[2] M. W. Spong, "Underactuated mechanical systems," in *Control Problems in Robotics and Automation*, pp. 135–150, Springer Berlin Heidelberg, 1998.

[3] A. M. Bloch, *Nonholonomic Mechanics and Control*. Interdisciplinary Applied Mathematics, Springer-Verlag New York, 2015.

[4] D. Lee, H. J. Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of Control, Automation and Systems*, vol. 7, no. 3, pp. 419–428, 2009.

[5] S. A. Al-Hiddabi, "Quadrotor control using feedback linearization with dynamic extension," in *International Symposium on Mechatronics and its Applications*, pp. 1–3, IEEE, 2009.

[6] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "Backstepping/nonlinear H-∞ control for path tracking of a quadrotor unmanned aerial vehicle," in *Proc. of the American Control Conference*, pp. 3356–3361, IEEE, 2008.

[7] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proc. of the International Conference on Robotics and Automation*, pp. 2247–2252, IEEE, 2005.

[8] E. Frazzoli, M. A. Dahleh, and E. Feron, "Trajectory tracking control design for autonomous helicopters using a backstepping algorithm," in *Proc. of the American Control Conference*, pp. 4102–4107, IEEE, 2000.

[9] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[10] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.

[11] R. Mahony and T. Hamel, "Robust trajectory tracking for a scale model autonomous helicopter," *International Journal of Robust and Nonlinear Control*, vol. 14, no. 12, pp. 1035–1059, 2004.

[12] M. Kobilarov, "Trajectory tracking of a class of underactuated systems with external disturbances," in *Proc. of the American Control Conference*, pp. 1044–1049, IEEE, 2013.

[13] I.-H. Choi and H.-C. Bang, "Adaptive command filtered backstepping tracking controller design for quadrotor unmanned aerial vehicle," *Proc. of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 226, no. 5, pp. 483–497, 2012.
[14] J. Umlauft and S. Hirche, "Feedback linearization based on Gaussian processes with event-triggered online learning," *IEEE Transactions on Automatic Control*, 2020.
[15] M. Greeff and A. P. Schoellig, "Exploiting differential flatness for robust learning-based tracking control using Gaussian processes," *IEEE Control Systems Letters*, vol. 5, no. 4, pp. 1121–1126, 2021.
[16] A. Capone and S. Hirche, "Backstepping for partially unknown nonlinear systems using Gaussian processes," *IEEE Control Systems Letters*, vol. 3, no. 2, pp. 416–421, 2019.
[17] T. Beckers, D. Kulić, and S. Hirche, "Stable Gaussian process based tracking control of Euler-Lagrange systems," *Automatica*, no. 103, pp. 390–397, 2019.
[18] M. K. Helwa, A. Heins, and A. P. Schoellig, "Provably robust learning-based approach for high-accuracy tracking control of Lagrangian systems," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1587–1594, 2019.
[19] F. S. Barbosa, L. Lindemann, D. V. Dimarogonas, and J. Tumova, "Provably safe control of Lagrangian systems in obstacle-scattered environments," in *Proc. of the Conference on Decision and Control*, pp. 2056–2061, IEEE, 2020.
[20] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with Gaussian processes," in *Proc. of the International Conference on Robotics and Automation*, pp. 491–496, IEEE, May 2016.
[21] A. Capone and S. Hirche, "Interval observers for a class of nonlinear systems using Gaussian process models," in *Proc. of the European Control Conference*, pp. 1350–1355, 2019.
[22] F. Scarselli and A. C. Tsoi, "Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results," *Neural networks*, vol. 11, no. 1, pp. 15–37, 1998.
[23] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*, vol. 1. MIT press Cambridge, 2006.
[24] I. Steinwart and A. Christmann, *Support vector machines*. Springer Science+Business Media, 2008.
[25] K. J. Åström and P. Eykhoff, "System identification—a survey," *Automatica*, vol. 7, no. 2, pp. 123–162, 1971.
[26] T. Beckers and S. Hirche, "Stability of Gaussian process state space models," in *Proc. of the European Control Conference*, pp. 2275–2281, 2016.
[27] D. H. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural computation*, vol. 8, no. 7, pp. 1341–1390, 1996.
[28] G. Wahba, *Spline models for observational data*. SIAM, 1990.
[29] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for Gaussian process optimization in the bandit setting," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.
[30] T. Beckers and S. Hirche, "Prediction with approximated Gaussian process dynamical models," *IEEE Transactions on Automatic Control (Early Access)*, 2021.
[31] A. Piazzi and C. G. L. Bianco, "Quintic G^2-splines for trajectory planning of autonomous vehicles," in *Proc. of the Intelligent Vehicles Symposium*, pp. 198–203, IEEE, 2000.

Received xxxx 20xx; revised xxxx 20xx; early access xxxx 20xx.

E-mail address: tbeckers@seas.upenn.edu
E-mail address: leonardo.colombo@car.upm-csic.es
E-mail address: hirche@lsr.ei.tum.de