Online Learning-Based Trajectory Tracking for Underactuated Vehicles With Uncertain Dynamics

Thomas Beckers, *Member, IEEE*, Leonardo J. Colombo, *Member, IEEE*, Sandra Hirche, *Fellow, IEEE*, and George J. Pappas, *Fellow, IEEE*

***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. Trajectory 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 tracking control law for underactuated rigid-body dynamics using an online learning-based oracle for the prediction of the unknown dynamics. We show that Gaussian process models are of particular interest for the role of the oracle. The presented approach guarantees a bounded tracking error with high probability where the bound is explicitly given. A numerical example highlights the effectiveness of the proposed control law.**

***Index Terms*—Autonomous vehicles, intelligent systems, statistical learning, uncertain systems.**

I. 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]. 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 [2] and backstepping methods [3] which are also analyzed in terms of stability, e.g., in [4].

However, these control approaches 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 a-priori 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 infeasible 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. 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 for instance [5].

The purpose of this letter is to employ the power of online learning-based approaches for the tracking control for a class of underactuated systems. At the end, 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 [6], [7] but these approaches are restricted to structured uncertainties such as uncertain parameters or use high feedback gains for compensation. Safe feedback linearization based on GPs are introduced in [8], [9] for a specific class of systems but they do not capture the general underactuated nature of the here considered model class and are limited to single input systems. Learning-based approaches for Euler-Lagrange systems with stability guarantees are presented in [10], [11]. However, the systems are required to be fully actuated or are limited to the class of balancing robots. For a specific type of aerial vehicles, a safe Gaussian process based

Manuscript received September 14, 2021; revised November 24, 2021; accepted December 10, 2021. Date of publication December 27, 2021; date of current version January 6, 2022. This work was supported by the AFOSR under Grant FA9550-19-1-0265 (Assured Autonomy in Contested Environments). Recommended by Senior Editor C. Seatzu. *(Corresponding author: Thomas Beckers.)*

Thomas Beckers and George J. Pappas are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: tbeckers@seas.upenn.edu; pappasg@seas.upenn.edu).

Leonardo J. Colombo is with the Centre for Automation and Robotics (CSIC-UPM), 28500 Madrid, Spain (e-mail: leonardo.colombo@car.upm-csic.es)

Sandra Hirche is with the Chair of Information-Oriented Control, Technical University of Munich, 80333 Munich, Germany (e-mail: hirche@tum.de).

Digital Object Identifier 10.1109/LCSYS.2021.3138546

2475-1456 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:54:51 UTC from IEEE Xplore. Restrictions apply.

controller is proposed in [12] but with additional assumptions such as an initial safe controller.

The contribution of this letter is a online learning-based tracking control law for a large class of underactuated vehicles with stability and performance guarantees. Instead of focusing on a particular type of oracle, the proposed approach allows the usage of various oracles. The online learning approach allows to improve the model and, thus, the tracking performance during runtime.

The remaining article is structured as follows: After the problem setting in Section II, the online learning-based oracle and the tracking controller are introduced in Section III. Finally, a numerical example is presented in Section IV.

II. 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 tensor $J \in \mathbb{R}^{3\times3}$. 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 state of the system. The vehicle is actuated with torques $\boldsymbol{\tau} \in \mathbb{R}^3$ and a force $u \in \mathbb{R}$, which is applied in a body-fixed direction defined by a unit vector $\boldsymbol{e} \in \mathbb{R}^3$. We can model the system as

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

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

$$\check{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_3 & -\omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \tag{2}$$

with the components of the angular velocity $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^\top$. The functions $\boldsymbol{f}\colon \mathbb{R}^6 \to \mathbb{R}^3$ and $\boldsymbol{f}_\omega\colon S \to \mathbb{R}^3$ are state-depended disturbances and/or unmodeled dynamics. It is assumed that the full state $\boldsymbol{s}$ can be measured. 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.

A. 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 $\boldsymbol{\rho}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}(\boldsymbol{x})$ 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 $A \in \mathbb{R}^{6\times6}$ and input matrix $B \in \mathbb{R}^{6\times3}$ given by

$$A = \begin{bmatrix} 0 & I_3 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{m} I_3 \end{bmatrix}, \tag{3}$$

[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. $||\cdot||$ denotes the Euclidean norm. $SO(3)$ and $SE(3)$ denote the special orthogonal and special Euclidean Lie groups, respectively.

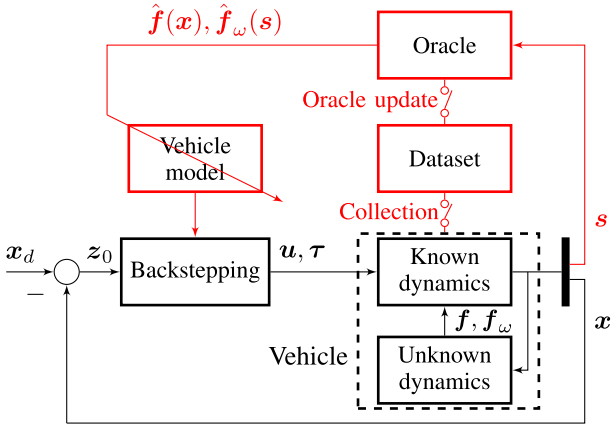


Fig. 1. Block diagram of the proposed control law.

and $I_3 \in \mathbb{R}^{3\times3}$ as identity matrix, we can rewrite (1) as

$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\Big(\boldsymbol{g}(R, u) + \hat{\boldsymbol{f}}(\boldsymbol{x}) + \boldsymbol{\rho}(\boldsymbol{x})\Big) \\ \dot{R} &= R\check{\boldsymbol{\omega}} \\ \dot{\boldsymbol{\omega}} &= J^{-1}\Big(J\boldsymbol{\omega} \times \boldsymbol{\omega} + \boldsymbol{\tau} + \hat{\boldsymbol{f}}_\omega(\boldsymbol{s}) + \boldsymbol{\rho}_\omega(\boldsymbol{s})\Big), \end{aligned} \tag{4}$$

where $\boldsymbol{g}\colon SO(3) \times \mathbb{R} \to \mathbb{R}^3$ is a virtual control input with $\boldsymbol{g}(R, u) := R\boldsymbol{e}u$. As consequence, (4) is equivalent to (1) without loss of generality.

III. Learning-Based Control

An overview about the proposed control strategy is depicted in Fig. 1. As introduced in the vehicle's dynamics (4), we assume that parts of the dynamics are known and some are unknown, i.e., $\boldsymbol{f}$ and $\boldsymbol{f}_\omega$. The proposed control strategy is based on a backstepping controller using an internal model that is updated by the predictions of an oracle. The data for the oracle is collected in arbitrary time intervals of the vehicle's dynamics during the control process. Then, the predictions of the oracle are updated based on the collected dataset and the vehicle model is improved. In the following, we present the online learning and control part in detail.

A. Learning

For the learning 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, the oracle collects $N(n)\colon \mathbb{N} \to \mathbb{N}$ training points of the system (1) to create a data set

$$\mathcal{D}_{n(t)} = \{\boldsymbol{s}^{\{i\}}, \boldsymbol{y}^{\{i\}}\}_{i=1}^{N(n)}. \tag{5}$$

The output data $\boldsymbol{y} \in \mathbb{R}^6$ are given by $\boldsymbol{y} = [(m\ddot{\boldsymbol{p}} - R\boldsymbol{e}u)^\top, (J(\dot{\boldsymbol{\omega}} - \boldsymbol{\omega} \times \boldsymbol{\omega}) - \boldsymbol{\tau})^\top]^\top$ such that the first three components of $\boldsymbol{y}$ correspond to $\boldsymbol{f}$ and the remaining to $\boldsymbol{f}_\omega$. The data set $\mathcal{D}_{n(t)}$ with $n\colon \mathbb{R}_{\geq0} \to \mathbb{N}$ can change over time t, such that at time $t_1 \in \mathbb{R}_{\geq0}$ the data set $\mathcal{D}_{n(t_1)}$ with $N(n(t_1))$ training points exists. This allows to accumulate training data over time, i.e., the number of training points $N(n)$ in the data set $\mathcal{D}_n$ is monotonically increasing, but also "forgetting" of training data to keep $N(n)$ constant. The time-dependent estimates of the oracle is denoted by $\hat{\boldsymbol{f}}_n(\boldsymbol{x})$ and $\hat{\boldsymbol{f}}_{\omega,n}(\boldsymbol{s})$ to highlight the dependence on the corresponding data set $\mathcal{D}_n$. Note that this construction

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:54:51 UTC from IEEE Xplore. Restrictions apply.

also allows offline learning, i.e., the prediction of the oracle depends on previous collected data only, or any hybrid online/offline approach.

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 $\mathcal{D}_n$. More powerful oracles are given by neural networks, due to their universal function approximation property. Furthermore, non-parametric oracles such as Gaussian processes and support vector machines have led to promising results as probabilistic function approximators [13], [14].

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

Assumption 1: Consider an oracle with the predictions $\hat{\boldsymbol{f}}_n \in \mathcal{C}^2$ and $\hat{\boldsymbol{f}}_{\omega,n} \in \mathcal{C}^0$ based on the data set $\mathcal{D}_n$ (5). Let $S_{\mathcal{X}} \subset (SE(3) \times (\mathcal{X} \subset \mathbb{R}^6))$ be a compact set where the derivatives of $\hat{\boldsymbol{f}}_n$ are bounded on $\mathcal{X}$. There exists a bounded function $\bar{\rho}_n \colon S_{\mathcal{X}} \to \mathbb{R}_{\geq 0}$ such that, if $||\cdot||$ denotes the Euclidean norm, the prediction error is bounded by

$$\mathrm{P}\left\{ \left\| \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}_n(\boldsymbol{x}) \\ \boldsymbol{f}_\omega(\boldsymbol{s}) - \hat{\boldsymbol{f}}_{\omega,n}(\boldsymbol{s}) \end{bmatrix} \right\| \leq \bar{\rho}_n(\boldsymbol{s}) \right\} \geq \delta \tag{6}$$

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

Assumption 2: The number of data sets $\mathcal{D}_n$ is finite and there are only finitely many switches of $n(t)$ over time, such that there exists a time $T \in \mathbb{R}_{\geq 0}$ where $n(t) = n_{\text{end}}, \forall t \geq T$

Assumption 1 is fulfilled, for instance, by a Gaussian process model as oracle as shown in the next section. The second assumption is little restrictive since the number of sets is often naturally bounded due to finite computational power or memory limitations and since the unknown functions $\boldsymbol{f}, \boldsymbol{f}_\omega$ in (1) is not time-dependent, long-life learning is typically not required. Therefore, there exists a constant data set $\mathcal{D}_{n_{end}}$ for all $t > T_{end}$. Furthermore, Assumption 2 ensures that the switching between the data sets is not infinitely fast which is natural in real world applications.

B. Gaussian Process as Oracle

Gaussian process models have been proven as very powerful oracle for nonlinear function regression. For the prediction, we concatenate the $N(n)$ training points of $\mathcal{D}_n$ in an input matrix $X = [\boldsymbol{s}^1, \boldsymbol{s}^2, \ldots, \boldsymbol{s}^{N(n)}]$ and a matrix of outputs $Y^\top = [\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^{N(n)}]$, where $\boldsymbol{y}$ might be corrupted by additive Gaussian noise with $\mathcal{N}(0, \sigma I_6)$. 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

$$\begin{aligned} \mu_i(\boldsymbol{y}^* | \boldsymbol{s}^*, \mathcal{D}_n) &= m_i(\boldsymbol{s}^*) + \boldsymbol{k}(\boldsymbol{s}^*, X)^\top K^{-1} \\ &\qquad \left(Y_{:,i} - [m_i(X_{:,1}), \ldots, m_i(X_{:,N})]^\top \right) \\ \mathrm{var}_i(\boldsymbol{y}^* | \boldsymbol{s}^*, \mathcal{D}_n) &= k(\boldsymbol{s}^*, \boldsymbol{s}^*) - \boldsymbol{k}(\boldsymbol{s}^*, X)^\top K^{-1} \boldsymbol{k}(\boldsymbol{s}^*, X). \end{aligned} \tag{7}$$

for all $i \in \{1, \ldots, 6\}$, 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 $j', j \in \{1, \ldots, N\}$ 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 $j \in \{1, \ldots, N\}$, 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. The hyperparameters and the variance σ of the Gaussian noise in the training data can be estimated by optimizing the marginal log. likelihood, see [13]. 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 [13]. As we use the oracle in an online setting where new data is collected over time, the dataset $\mathcal{D}_n$ for the prediction (7) changes over time. The GP model allows to integrate new training data in a simple way by exploiting that every subset follows a multivariate Gaussian distribution.

Remark 2: The mean prior can be achieved by common system identification techniques of the unknown dynamics $\boldsymbol{f}, \boldsymbol{f}_\omega$ as described in [15]. However, without any prior knowledge the mean function is set to zero, i.e., $m_i(\boldsymbol{s}) = 0$.

Based on (7), the normal distributed components $y_i^* | \boldsymbol{s}^*, \mathcal{D}_n$ are combined into a multi-variable distribution which leads to $\boldsymbol{y}^* | (\boldsymbol{s}^*, \mathcal{D}_n) \sim \mathcal{N}(\boldsymbol{\mu}(\cdot), \Sigma(\cdot))$, where

$$\boldsymbol{\mu}(\boldsymbol{y}^* | \boldsymbol{s}^*, \mathcal{D}_n) = [\mu_1(\cdot), \ldots, \mu_6(\cdot)]^\top \tag{8}$$

$$\Sigma(\boldsymbol{y}^* | \boldsymbol{s}^*, \mathcal{D}_n) = \mathrm{diag}[\mathrm{var}_1(\cdot), \ldots, \mathrm{var}_6(\cdot)]. \tag{9}$$

Remark 3: For notational simplicity, we consider identical kernels for each output dimension. However, the GP model can be easily adapted to different kernels.

With the introduced GP model, we are now addressing Assumption 1 using [10], [14], [16]. To provide model error bounds for the posterior mean (8) as estimation of $\boldsymbol{f}$ and $\boldsymbol{f}_\omega$, additional assumptions on the unknown parts of the dynamics (1) must be introduced, in line with the no-free-lunch theorem, see [17].

Assumption 3: 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, i.e., $\|f_i\|_k < \infty$, $\|f_{\omega,i}\|_k < \infty$ 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 [18]. Assumption 3 requires that the kernel must be selected in such a way that the functions $\boldsymbol{f}, \boldsymbol{f}_\omega$ are elements of the associated RKHS. 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 [14, Lemma 4.55] such that the bounded RKHS norm is a mild assumption. Finally, with Assumption 3, the model error can be bounded as written in the following lemma.

Lemma 1 (Adapted From [10]): Consider the unknown functions $\boldsymbol{f}, \boldsymbol{f}_\omega$ and a GP model satisfying Assumption 3. The model error is bounded by

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

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:54:51 UTC from IEEE Xplore. Restrictions apply.

for $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{s} \in S_{\mathcal{X}}, \delta \in (0,1)$ with $\boldsymbol{\beta}_n \in \mathbb{R}^6$ given by [10, Lemma 1].

Proof: It is a direct implication of [10, Lemma 1]. ■

With Assumption 3 and the fact, that universals kernels exist which generate bounded predictions with bounded derivatives, see [16], GP models can be used as oracle to fulfill Assumption 1. In this case, the prediction error bound is given by $\bar{\rho}_n(\boldsymbol{s}) := \|\boldsymbol{\beta}_n{}^{\top}\Sigma^{\frac{1}{2}}([\hat{\boldsymbol{f}}_n(\boldsymbol{x})^{\top}, \hat{\boldsymbol{f}}_{\omega,n}(\boldsymbol{s})^{\top}]^{\top}|\boldsymbol{s}, \mathcal{D}_n)\|$ as shown in Lemma 1.

Remark 4: An efficient greedy algorithm can be used to find $\boldsymbol{\beta}_n$ based on the maximum information gain. Even though the values of the elements of $\boldsymbol{\beta}$ are typically increasing with the number of training data, it is possible to learn the true function $\boldsymbol{f}, \boldsymbol{f}_{\omega}$ arbitrarily exactly due to the shrinking variance Σ, see [19]. In general, the prediction error bound $\bar{\rho}_n(\boldsymbol{s})$ is large if the uncertainty of the oracle's prediction is high and vice versa. Additionally, the bound is typically increasing if the set $S_{\mathcal{X}}$ is expanded. The stochastic nature of the bound is due to the fact that just a finite number of noisy training points are available.

C. Tracking Control

For the tracking control, we consider a given desired trajectory $\boldsymbol{x}_d(t): \mathbb{R}_{t\geq 0} \to \mathcal{X}, \boldsymbol{x}_d \in \mathcal{C}^4$. The tracking error is denoted by $z_0(t) = \boldsymbol{x}(t) - \boldsymbol{x}_d(t)$. Before we propose the main theorem about the safe learning-based tracking control law, the feedback gain matrix G_n is introduced. As part of the controller, G_n 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). The feedback gain matrix is allowed to be adapted with any update of the oracle based on a new data set $\mathcal{D}_n$ to lower the feedback gains when the oracle's accuracy is improved.

Property 1: The matrix $G_n \in \mathbb{R}^{3\times 6}$ is chosen such that there exist a symmetric positive definite matrix $P_n \in \mathbb{R}^{6\times 6}$ and a positive definite matrix $Q_n \in \mathbb{R}^{6\times 6}$ which satisfy the Lyapunov equation

$$P_n(A - BG_n) + (A - BG_n)^{\top}P_n = -Q_n \tag{10}$$

for each switch of $n(t)$.

Property 1 is satisfied if the real parts of all eigenvalues of $(A - BG_n)$ are negative. For example, this can be achieved by any $G_n = [G_{n,1}, G_{n,2}]$, where $G_{n,1}, G_{n,2} \in \mathbb{R}^{3\times 3}$ are positive definite diagonal matrices.

Theorem 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 Assumptions 1 and 2. Let $G_{z_1}, G_{z_2} \in \mathbb{R}^{3\times 3}$ be positive definite symmetric matrices. With Property 1, the control law

$$\begin{aligned}\boldsymbol{\tau} &= J(\boldsymbol{e} \times (R^{\top}\boldsymbol{g}_{\ddot{d}} - \check{\boldsymbol{\omega}}^2\boldsymbol{e}u - 2\check{\boldsymbol{\omega}}\boldsymbol{e}\dot{u})u^{-1}) - J\boldsymbol{\omega} \times \boldsymbol{\omega} - \hat{\boldsymbol{f}}_{\omega}(\boldsymbol{s}),\\ \ddot{u} &= \boldsymbol{e}^{\top}(R^{\top}\boldsymbol{g}_{\ddot{d}} - \check{\boldsymbol{\omega}}^2\boldsymbol{e}u - 2\check{\boldsymbol{\omega}}\boldsymbol{e}\dot{u}),\end{aligned} \tag{11}$$

with the desired virtual control input derivative

$$\begin{aligned}\boldsymbol{g}_{\ddot{d}} = m\boldsymbol{p}_d^{(4)} &- G_n\left(\frac{\partial \dot{\hat{\boldsymbol{x}}}}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}} - \ddot{\boldsymbol{x}}_d\right) - BP_n(\dot{\hat{\boldsymbol{x}}} - \dot{\boldsymbol{x}}_d)\\ &- (G_{z_1} + G_{z_2})\left(\dot{\boldsymbol{g}} - m\boldsymbol{p}_d^{(3)} + G_n(\dot{\hat{\boldsymbol{x}}} - \dot{\boldsymbol{x}}_d) + \frac{\partial \hat{\boldsymbol{f}}_n}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}}\right)\\ &- (G_{z_2}G_{z_1} + I_3)\Big(\boldsymbol{g} - m\ddot{\boldsymbol{p}}_d + G_nz_0 + \hat{\boldsymbol{f}}_n(\boldsymbol{x})\Big)\\ &- G_{z_2}B^{\top}P_nz_0 - \frac{\partial}{\partial \boldsymbol{x}}\left[\frac{\partial \hat{\boldsymbol{f}}_n}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}}\right]\dot{\hat{\boldsymbol{x}}}.\end{aligned} \tag{12}$$

$$\dot{\hat{\boldsymbol{x}}} = A\boldsymbol{x} + B\Big(\boldsymbol{g}(R,u) + \hat{\boldsymbol{f}}_n(\boldsymbol{x})\Big) \tag{13}$$

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

$$\mathrm{P}\{\|z_0(t)\| \leq \max_{\boldsymbol{s}\in S_{\mathcal{X}}} \bar{\rho}_{n_{\mathrm{end}}}(\boldsymbol{s})b_{n_{\mathrm{end}}}, \forall t \geq T\} \geq \delta \tag{14}$$

with $b_{n_{\mathrm{end}}} = (\max\{\mathrm{eig}(P_{n_{\mathrm{end}}}), 1\}/\min\{\mathrm{eig}(P_{n_{\mathrm{end}}}), 1\})^{1/2}$, and time constant $T \in \mathbb{R}_{\geq 0}$ on $S_{\mathcal{X}}$.

Remark 5: 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 (5), which can often deal with noisy data. For instance, GP models can handle additive Gaussian noise on the output [13].

We prove the stability of the closed-loop with the proposed control law with multiple Lyapunov function, where the n-th function is active when the oracle predicts based on the corresponding training set $\mathcal{D}_n$. Note that due to a finite number of switching events, the switching between stable systems can not lead to an unbounded trajectory, see [20].

Proof: The term $\boldsymbol{g}(R, \boldsymbol{u})$ in (4) is assumed as virtual control input with the desired force

$$\boldsymbol{g}_d(t, n, \boldsymbol{x}) = m\ddot{\boldsymbol{p}}_d - G_nz_0 - \hat{\boldsymbol{f}}_n(\boldsymbol{x}) \tag{15}$$

where G_n can change by the switching of $n(t)$. The tracking error dynamics are given by

$$\dot{z}_0 = A\boldsymbol{x} + B\Big(\boldsymbol{g}(R,u) + \hat{\boldsymbol{f}}_n(\boldsymbol{x}) + \boldsymbol{\rho}_n(\boldsymbol{x})\Big) - \begin{bmatrix}\dot{\boldsymbol{p}}_d\\ \ddot{\boldsymbol{p}}_d\end{bmatrix}. \tag{16}$$

Using the desired acceleration $\ddot{\boldsymbol{p}}_d$ of (15) in (16) leads to

$$\dot{z}_0 = (A - BG_n)z_0 + B\big(\boldsymbol{g}(R,u) - \boldsymbol{g}_d(t,n,\boldsymbol{x}) + \boldsymbol{\rho}_n(\boldsymbol{x})\big).$$

In the next step, the boundedness of the tracking error z_0 is proven. For this purpose, we use the matrices P_n, Q_n of Property 1 to construct the Lyapunov function $V_{0,n}(z_0) = 0.5z_0^{\top}P_nz_0$ and compute its evolution

$$\dot{V}_{0,n} = -z_0^{\top}Q_nz_0 + (B^{\top}P_nz_0)^{\top}\big(\boldsymbol{g}(R,u) - \boldsymbol{g}_d + \boldsymbol{\rho}_n(\boldsymbol{x})\big).$$

The first summand is negative for all $z_0 \in \mathbb{R}^6$. In the next step, we extend the previous Lyapunov function with the error term $z_1 \in \mathbb{R}^3$ with $z_1(t,n,\boldsymbol{x},R,u) = \boldsymbol{g}(R,u) - \boldsymbol{g}_d(t,n,\boldsymbol{x})$, which describes the error between the virtual and the desired control input. Thus, it leads to a switching Lyapunov function $V_{1,n}(z_0, z_1) = V_{0,n} + 0.5z_1^{\top}z_1 \geq 0$. The derivative of $V_{1,n}$ leads to

$$\dot{V}_{1,n} = \dot{V}_{0,n} + z_1^{\top}\Big(\dot{\boldsymbol{g}} - m\boldsymbol{p}_d^{(3)} + G_n\dot{z}_0 + \dot{\hat{\boldsymbol{f}}}_n(\boldsymbol{x})\Big), \tag{17}$$

where $\boldsymbol{p}_d^{(3)}$ denotes the third time-derivative of the desired position $\boldsymbol{p}_d$. Following again the idea of a desired virtual input as in (15), we construct a desired value of $\dot{\boldsymbol{g}}$ with

$$\boldsymbol{g}_{\dot{d}} = m\boldsymbol{p}_d^{(3)} - G_n(\dot{\hat{\boldsymbol{x}}} - \dot{\boldsymbol{x}}_d) - B^{\top}P_nz_0 - G_{z_1}z_1 - \frac{\partial \hat{\boldsymbol{f}}_n}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}}.$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:54:51 UTC from IEEE Xplore. Restrictions apply.

Instead of having dependencies on the typical noisy state derivative $\dot{\boldsymbol{x}}$, we use the estimation $\hat{\dot{\boldsymbol{x}}} \in \mathbb{R}^6$ given by (13), which only contains the known parts of the system dynamics (4). Then, the last expression is used to substitute $\dot{\boldsymbol{g}}$ in (17). This leads to the evolution

$$\dot{V}_{1,n} = -z_0^\top Q_n z_0 - z_1^\top G_{z_1} z_1 + z_0^\top P_n B \boldsymbol{\rho}_n(\boldsymbol{x}) + z_1^\top \left(\left[\frac{\partial \hat{\boldsymbol{f}}_n}{\partial \boldsymbol{x}} + G_n \right] B \boldsymbol{\rho}_n(\boldsymbol{x}) + \dot{\boldsymbol{g}} - \boldsymbol{g}_{\dot{d}} \right). \quad (18)$$

Next, we define the error $z_2 \in \mathbb{R}^3$ with $z_2 = \dot{\boldsymbol{g}} - \boldsymbol{g}_{\dot{d}}$ and an extended Lyapunov function

$$V_n(z_0, z_1, z_2) = V_{1,n} + \frac{1}{2} z_2^\top z_2 \geq 0. \quad (19)$$

The derivative of V_n leads to

$$\dot{V}_n = \dot{V}_{1,n} + z_2^\top \left(\ddot{\boldsymbol{g}} - m \boldsymbol{p}_d^{(4)} + G_n \ddot{z}_0 + B P_n \dot{z}_0 + \frac{d}{dt} \left[\frac{\partial \hat{\boldsymbol{f}}_n}{\partial \boldsymbol{x}} \dot{\hat{\boldsymbol{x}}} \right] \right)$$

and we construct a desired value of $\ddot{\boldsymbol{g}}$ with $\boldsymbol{g}_{\ddot{d}}$ given by (12). Then, it is substituted into $\dot{V}_n$ to obtain

$$\dot{V}_n = -z_0^\top Q_n z_0 - z_1^\top G_{z_1} z_1 - z_2^\top G_{z_2} z_2 + (z_0^\top P_n + z_1^\top D(\boldsymbol{x}) + z_2^\top E(\boldsymbol{x})) B \boldsymbol{\rho}_n(\boldsymbol{x}) + z_2^\top (\ddot{\boldsymbol{g}} - \boldsymbol{g}_{\ddot{d}}) \quad (20)$$

$$D(\boldsymbol{x}) := \frac{\partial \hat{\boldsymbol{f}}_n}{\partial \boldsymbol{x}} + G_n \quad (21)$$

$$E(\boldsymbol{x}) := BP + G_{z_1} D + G_n \frac{\partial \dot{\hat{\boldsymbol{x}}}}{\partial \boldsymbol{x}} + \frac{\partial}{\partial \boldsymbol{x}} \left(\frac{\partial \hat{\boldsymbol{f}}_n}{\partial \boldsymbol{x}} \dot{\hat{\boldsymbol{x}}} \right). \quad (22)$$

To eliminate the last summand in (20) except for the estimation error $\boldsymbol{\rho}_\omega$, we note that

$$\ddot{\boldsymbol{g}}(R, u) = R(\check{\boldsymbol{\omega}}^2 \boldsymbol{e} u + 2\check{\boldsymbol{\omega}} \dot{u} + \dot{\boldsymbol{\omega}} \times \boldsymbol{e} u + \boldsymbol{e} \ddot{u}), \quad (23)$$

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

$$\dot{\boldsymbol{\omega}} \times \boldsymbol{e} u + \boldsymbol{e} \ddot{u} = R^\top \boldsymbol{g}_{\ddot{d}} - \check{\boldsymbol{\omega}}^2 \boldsymbol{e} u - 2\check{\boldsymbol{\omega}} \dot{u}. \quad (24)$$

Using (24) and Assumption 1, the evolution of the Lyapunov function V can be upper bounded by

$$\mathrm{P}\{\dot{V}_n \leq -z_0^\top Q_n z_0 - z_1^\top G_{z_1} z_1 - z_2^\top G_{z_2} z_2 + \left\| (z_0^\top P_n + z_1^\top \bar{D} + z_2^\top \bar{E}) B + \bar{c} z_2 \right\| \bar{\rho}_n(\boldsymbol{s}) \} \geq \delta, \quad (25)$$

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 $z = [z_1^\top, z_2^\top, z_3^\top]^\top$ outside a ball

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

where a maximum of $\bar{\rho}_n$ exists regarding to Assumption 1. Finally, the Lyapunov function (19) is lower and upper bounded by $\alpha_1(\|z\|) \leq V_n(z) \leq \alpha_2(\|z\|)$, where $\alpha_1(r) = 0.5 \min\{\mathrm{eig}(P_n), 1\} r^2$ and $\alpha_2(r) = 0.5 \max\{\mathrm{eig}(P_n), 1\} r^2$. Thus, we can compute the radius $b_n \in \mathbb{R}_{\geq 0}$ of the bound by

$$b_n = \sqrt{\max\{\mathrm{eig}(P_n), 1\} / \min\{\mathrm{eig}(P_n), 1\}}. \quad (26)$$

Since Assumption 2 only allows a finite number of switches, there exists a time $T \in \mathbb{R}_{\geq 0}$ such that $n(t) = n_{\mathrm{end}}\ \forall t \geq T$. Thus, $\mathrm{P}\{\|z_0(t)\| \leq \max_{\boldsymbol{s} \in S_{\mathcal{X}}} \bar{\rho}_{n_{\mathrm{end}}}(\boldsymbol{s}) b_{n_{\mathrm{end}}}, \forall t \geq T\} \geq \delta$. ■

Remark 6: An extension of Theorem 1 to include the rotation $\boldsymbol{\omega}$ in the bound (14) is easy to achieve by additional terms in the Lyapunov function as given in [4], [6].

Remark 7: The torque control law of (11) has a singularity at $u = 0$ as without control force u no tracking control is possible in general. To overcome the singularity, a reasonable trajectory planning can be performed, see [6] or the control torques are set to zero at this point. In practice, this leads to chattering that can be alleviated by a slight modification of the control law to remove the singularity, see [21].

Remark 8: The proposed approach allows multiple ways of data collection and adaptation of the feedback matrix G_n. A possible strategy can be time-triggered where new data points are recurrently attached to the data set $\mathcal{D}_n$ to improve the prediction accuracy of the oracle and the magnitude of G_n is decreased over time. More advanced strategies can be based on the model uncertainty or the tracking error, see [8].

The proof shows that the bound of the tracking error (26) depends on the prediction error $\bar{\rho}_n$ of the oracle.

IV. Numerical Example

In this section, we present a numerical example of a quadrocopter within an a-priori unknown wind field. The dynamics of the quadrocopter are described by (1) with mass $m = 1kg$, inertia $J = \mathrm{diag}(2, 2, 1) kgm^2$ and the direction $\boldsymbol{e} = [0, 0, 1]^\top$ of the force input u. As unknown dynamics $\boldsymbol{f}$ and $\boldsymbol{f}_\omega$, we consider an arbitrarily chosen wind field and the gravity force given by

$$\boldsymbol{f}(\boldsymbol{x}) = [0, 0, 2\sin(x_1) + \exp(-5x_2^2) - 9.81]^\top \quad (27)$$

$$\boldsymbol{f}_\omega(\boldsymbol{s}) = [2\exp(-x_1^2 - x_2^2) + \omega_1 \cos(x_2)^2, 0, 0]^\top. \quad (28)$$

The posterior mean $\boldsymbol{\mu}(\boldsymbol{y}|\boldsymbol{s}, \mathcal{D}_n)$ of a GP model is used as oracle to predict the z-component of $\boldsymbol{f}(\boldsymbol{x})$ and the x-component of $\boldsymbol{f}_\omega(\boldsymbol{s})$ with the squared exponential kernel, see [13]. Thus, the partial derivative $\frac{\partial \hat{\boldsymbol{f}}}{\partial \boldsymbol{x}}$ in (22) is the derivative of the posterior mean (8). The prior knowledge about the existing gravity in $\boldsymbol{f}(\boldsymbol{x})$ is considered as estimate in the mean function of the GP with $m_3(\boldsymbol{s}) = -10$. At starting time $t = 0$, the data set $\mathcal{D}_n$ is empty such that the prediction is solely based on the mean function. The initial position of the quadrocopter is $\boldsymbol{p}(0) = [0.1, -0.1, 0]^\top$ whereas the desired trajectory starts at $\boldsymbol{p}_d(0) = [0, 0, 0]^\top$ due to an assumed position measurement error. We employ an online learning approach which collects a new data point every $0.1s$ such that the total number of data points is $N = 5n$.

In Fig. 2, the first $3s$ of the desired (dashed) and the actual trajectory (solid) is shown. The crosses denote the collected training data. Each training point consists of the actual state $\boldsymbol{s}$ and $\boldsymbol{y}$ as given by (5). Since the training point depends on the typically noisy measurement of the accelerations $\ddot{\boldsymbol{p}}$ and $\dot{\boldsymbol{\omega}}$, Gaussian distributed noise $\mathcal{N}(0, 0.08^2 I_3)$ is added to the measurement. The GP model is updated every second until $t = 12s$, where the last 10 collected training points are appended to the set $\mathcal{D}_n$ and the hyperparameters are optimized by means of the likelihood function, see [14].

Thus, the function n is the integer part of $2t$ up to $t = 12s$ given by $n(t) = \min(12s, \lfloor 2t \rfloor)$. The initial feedback gain matrix is set to $G_{n=0} = [\mathrm{diag}(10, 10, 40), \mathrm{diag}(10, 10, 10)]$ and $G_{z_1} = G_{z_2} = 2I_3$. In this example, we adapt the feedback

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:54:51 UTC from IEEE Xplore. Restrictions apply.

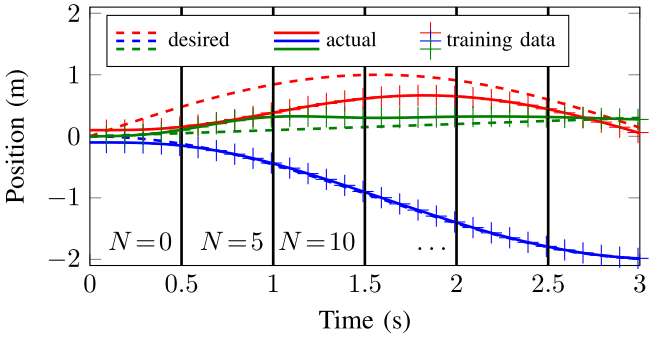


Fig. 2. A segment of the desired (dashed) and actual trajectory (solid). Every 0.1s a training point (cross) is recorded. Every 0.5s the oracle is updated based on all collected training points N. The additional training data allows to refine the model such that the tracking error is decreasing.

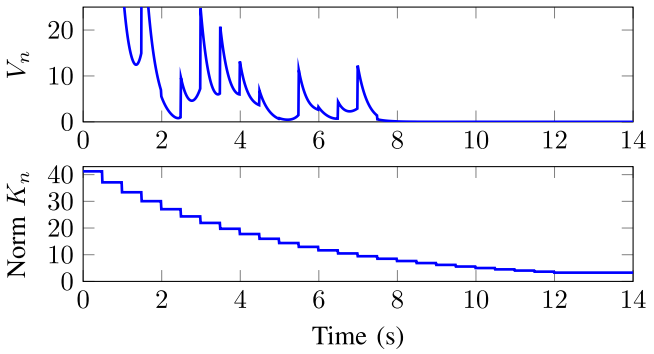


Fig. 3. Top: Lyapunov function converges to a tight set around zero. The jumps occur when the oracle is updated. Bottom: Norm of the feedback gain matrix is decreasing due to improved accuracy of the oracle.

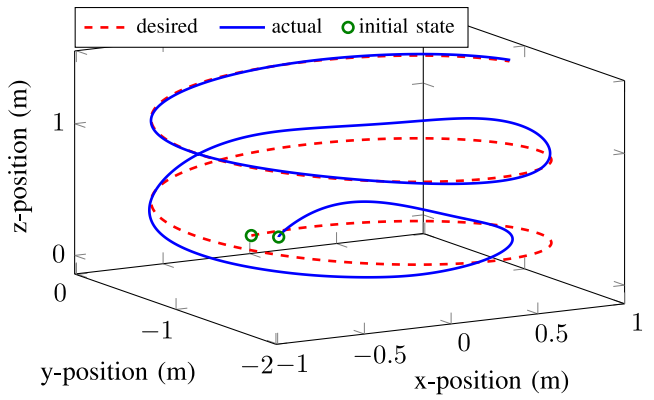


Fig. 4. Actual trajectory converges to desired trajectory.

gain matrix based on the number of training points. When the GP model is updated with new training data, the feedback gains are decreased by $G_n = 0.9^n G_{n=0}$. Thus, after the first update, the feedback gains are 90% of the initial gains, see Fig. 3. The simulation time is 14s.

Figure 4 visualizes that the actual position (solid) of the quadrocopter converges to a very tight set around the desired position (dashed). The effects of the switching to the updated GP model are more noticeable in the evolution of the Lyapunov function in Fig. 3. The function might increase after an update of the GP model due to the change of G_n and the new prediction accuracy of the GP model. However, the function converges to a bounded set as proposed in Theorem 1 after the finite number of switching events.

V. Conclusion

We present an online 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. The online fashion of the proposed approach allows to improve the quality of the oracle over time and, thus, to improve the tracking performance. Even though no particular oracle is assumed, we show that Gaussian processes fulfill all requirements as oracle in the proposed control scheme.

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 Trans. Autom. Control*, vol. 44, no. 9, pp. 1663–1671, Sep. 1999.

[2] D. Lee, H. J. Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *Int. J. Control Autom. Syst.*, vol. 7, no. 3, pp. 419–428, 2009.

[3] 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. Amer. Control Conf.*, 2008, pp. 3356–3361.

[4] E. Frazzoli, M. A. Dahleh, and E. Feron, "Trajectory tracking control design for autonomous helicopters using a backstepping algorithm," in *Proc. Amer. Control Conf.*, 2000, pp. 4102–4107.

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

[6] R. Mahony and T. Hamel, "Robust trajectory tracking for a scale model autonomous helicopter," *Int. J. Robust Nonlinear Control*, vol. 14, no. 12, pp. 1035–1059, Aug. 2004.

[7] M. Kobilarov, "Trajectory tracking of a class of underactuated systems with external disturbances," in *Proc. Amer. Control Conf.*, 2013, pp. 1044–1049.

[8] J. Umlauft and S. Hirche, "Feedback linearization based on Gaussian processes with event-triggered online learning," *IEEE Trans. Autom. Control*, vol. 65, no. 10, pp. 4154–4169, Oct. 2020.

[9] M. Greeff and A. P. Schoellig, "Exploiting differential flatness for robust learning-based tracking control using Gaussian processes," *IEEE Control Syst. Lett.*, vol. 5, no. 4, pp. 1121–1126, Oct. 2021.

[10] T. Beckers, D. Kulić, and S. Hirche, "Stable Gaussian process based tracking control of Euler-Lagrange systems," *Automatica*, no. 103, pp. 390–397, May 2019.

[11] M. K. Helwa, A. Heins, and A. P. Schoellig, "Provably robust learning-based approach for high-accuracy tracking control of Lagrangian systems," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1587–1594, Apr. 2019.

[12] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with Gaussian processes," in *Proc. Int. Conf. Robot. Autom.*, 2016, pp. 491–496.

[13] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2006.

[14] I. Steinwart and A. Christmann, *Support Vector Machines*. Dordrecht, The Netherlands: Springer, 2008, doi: 10.1007/978-0-387-77242-4.

[15] K. J. Åström and P. Eykhoff, "System identification—A survey," *Automatica*, vol. 7, no. 2, pp. 123–162, 1971.

[16] T. Beckers and S. Hirche, "Stability of Gaussian process state space models," in *Proc. Eur. Control Conf.*, 2016, pp. 2275–2281.

[17] D. H. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural Comput.*, vol. 8, no. 7, pp. 1341–1390, 1996.

[18] G. Wahba, *Spline Models for Observational Data*. Philadelphia, PA, USA: SIAM, 1990.

[19] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, "Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes," in *Proc. IEEE Conf. Decis. Control*, 2016, pp. 4661–4666.

[20] D. Liberzon and A. S. Morse, "Basic problems in stability and design of switched systems," *IEEE Control Syst. Mag.*, vol. 19, no. 5, pp. 59–70, Oct. 1999.

[21] H. K. Khalil, *Noninear Systems*, Hoboken, NJ, USA: Prentice-Hall, 1996.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:54:51 UTC from IEEE Xplore. Restrictions apply.