2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

Learning-based Balancing of Model-based and Feedback Control for Second-order Mechanical Systems

Thomas Beckers[1], Leonardo J. Colombo[2], Manfred Morari[1], and George J. Pappas[1]

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9992751

Abstract— **High-performance tracking control of mechanical systems typically requires model-based control as it enables to counteract undesirable dynamics in a timely fashion. The quality of the compensation depends on the accuracy of the underlying model. However, the dynamics are often (partially) infeasible for complex systems in a-priori unknown environments. Due to the favorable properties of model-based control, the goal is to apply feedback control for error compensation only when necessary. In this paper, we present an online learning-based balancing (OLBB) framework to adapt the feedback gains such that the controlled system with state-dependent uncertainties satisfies given performance specifications. For this purpose, an oracle predicts the unknown dynamics and an error model adapts the feedback gains. The framework can be easily applied to existing control approaches to improve the safety and performance of the closed-loop.**

I. Introduction

Trajectory tracking is essential for many modern applications such as robotic manipulators or unmanned vehicles. Many different types of control methods have been proposed to achieve an accurate trajectory tracking for these mostly nonlinear mechanical systems, mostly based on the idea of feedback linearization, computed torque control or backstepping methods [1]. The controller is in general separated into a model-based and a feedback part. A precise model of the system is necessary to compensate its dynamics to achieve a low gain feedback term. This is beneficial in many ways: it avoids the saturation of actuators, avoids large errors in the presence of noise, and enhances safety in applications such as human-robot interaction due to reduced stiffness [2].

As the performance of the compensation depends on the quality of the model, all dynamics must be modeled as accurately as possible. However, a precise model of the system and the environment is often very time-consuming and hard to obtain by using first principles. Especially in modern applications including for instance service robotics, the interaction with unstructured and a priori unknown environments introduces unstructured, state-dependent uncertainties. Typical examples for these uncertainties are position dependent wind fields in drone control or velocity dependent friction in robotics.

This work was supported by the AFOSR under grant FA9550-19-1-0265 (Assured Autonomy in Contested Environments) and a 2020 Leonardo Grant for Researchers and Cultural Creators.

[1] are with the Department of Electrical and Systems Engineering, University of Pennsylvania, 19104 Philadelphia, USA (e-mail: [tbeckers, morari, pappasg]@seas.upenn.edu).

[2] is with the Centre for Automation and Robotics (CSIC-UPM), Ctra. M300 Campo Real, Km 0,200, Arganda del Rey - 28500 Madrid, Spain (e-mail: leonardo.colombo@car.upm-csic.es)

A common approach is to derive a dynamic model from first principles [3] and to increase the feedback gains to compensate any force-like uncertainties until a desired tracking performance is achieved [4]. However, as explained above, high-gain control is undesired, while deriving a more accurate model of the system is often difficult, e.g. in soft robotics [5]. Further, the stability of the closed-loop system might not be guaranteed. Adaptive control is capable of adapting the feedback gains based on the system behavior but it typically requires a parametric model of the system. A convenient approach to avoid the modeling process is provided by learning-based oracles such as neural networks or Gaussian processes (GPs). These data-driven models have shown remarkable results in many different control applications with a minimum of engineering effort, see [3].

The ultimate goal is to satisfy performance specifications for the tracking of a mechanical system with unknown, state-dependent forces while keeping the feedback gains as low as possible. Learning-based approaches for safe tracking control have drawn attention recently. For instance, tracking control of Euler-Lagrange systems with stability guarantees is presented in [6], [7]. The problem of tracking control of aerial/underwater vehicles with uncertainties has been addressed in [8], [9], though these approaches are restricted to structured uncertainties or use high feedback gains for compensation. Safe feedback linearization based on GPs are introduced in [10], [11]. However, all these approaches do not balance the model-based and feedback control such that the systems are unnecessarily stiff and hence, a safety issue in case of potential collisions. The balancing task for the tracking control for a class of systems is addressed in [6], [12], [13] but it neither allows online learning nor shows how to determine the feedback gains. Thus, the balancing of model-based and feedback control for safe, high-performance tracking control is still an open problem.

Contribution: The contribution of this article is an online learning-based balancing (OLBB) framework for tracking control of second-order mechanical systems with unknown state-dependent disturbances. Given a performance specification, our approach minimizes the feedback induced stiffness such that the system remains as compliant as possible while satisfying the specifications. Specially, an online updated oracle predicts the unmodeled system behavior so that the minimum necessary feedback gains are obtained.

The remaining article is structured as follows: After the preliminaries in Section II, the learning-based balancing framework is presented in Section III. Finally, a numerical example is given in Section IV.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:39:08 UTC from IEEE Xplore. Restrictions apply.

II. PRELIMINARIES

We consider a second-order mechanical system given by[1]

$$\dot{\boldsymbol{x}}_1 = \boldsymbol{x}_2,\ \boldsymbol{\tau} = \underbrace{M(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2 + C(\boldsymbol{x})\boldsymbol{x}_2 + G(\boldsymbol{x}_1)}_{\text{known}} + \underbrace{\boldsymbol{f}(\boldsymbol{x})}_{\text{unknown}} \quad (1)$$

with the state vector $\boldsymbol{x} = [\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top]^\top \in \mathcal{X} \subseteq \mathbb{R}^{2n}$, generalized forces $\boldsymbol{\tau} \in \mathbb{R}^n$, positive definite inertia matrix $M(\boldsymbol{x}_1) \in \mathbb{R}^{n\times n}$, Coriolis matrix $C(\boldsymbol{x}) \in \mathbb{R}^{n\times n}$ and gravity vector $G(\boldsymbol{x}_1) \in \mathbb{R}^n$. This class of systems includes but is not limited to fully actuated Euler-Lagrange systems as many types of industrial robots [14] and rigidity-based formation control [15]. The function $\boldsymbol{f}\colon \mathcal{X} \to \mathbb{R}^n$ is assumed to be a state-dependent, unknown part of the dynamics. Typical sources of $\boldsymbol{f}$ are friction in mechanical systems, wind disturbances for drones or unmodeled dynamics. Further, we assume that the system (1) has a unique solution. In the following, a desired trajectory is given by $\boldsymbol{x}_d \in \mathcal{C}^2, \boldsymbol{x}_d\colon \mathbb{R}_+ \to \mathcal{X}$, where $\boldsymbol{x}_d(t) = [x_{d,1}, \dots, x_{d,n}, \dot{x}_{d,1}, \dots, \dot{x}_{d,n}]^\top$. The tracking error is denoted by $\boldsymbol{e}(t) = \boldsymbol{x}(t) - \boldsymbol{x}_d(t)$. We consider the existence of a nominal control law

$$\boldsymbol{\tau} = \boldsymbol{h}_{\text{mb}}(\boldsymbol{x}, \boldsymbol{x}_d) + \boldsymbol{h}_{\text{fb}}(\boldsymbol{x})H\boldsymbol{e} \quad (2)$$

with model-based term $\boldsymbol{h}_{\text{mb}}\colon \mathcal{X}\times\mathcal{X} \to \mathbb{R}^n$ and feedback $\boldsymbol{h}_{\text{fb}}\colon \mathcal{X} \to \mathbb{R}^{n\times n}$ with a designable feedback gain $H \in \mathcal{H} \subseteq \mathbb{R}^{n\times 2n}$. The model-based term $\boldsymbol{h}_{\text{mb}}$ is usually derived based on M, C and G of the mechanical system (1). Further, we consider the nominal control law (2) to stabilize the known dynamics of (1) while satisfying the following property.

Assumption 1 *The control law* (2) *is designed such that for the closed-loop with the* ***known dynamics***

$$\begin{aligned} \dot{\boldsymbol{x}}_1 &= \boldsymbol{x}_2 \quad (3)\\ \boldsymbol{h}_{\text{mb}}(\boldsymbol{x}, \boldsymbol{x}_d) + \boldsymbol{h}_{\text{fb}}(\boldsymbol{x})H\boldsymbol{e} &= M(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2 + C(\boldsymbol{x})\boldsymbol{x}_2 + G(\boldsymbol{x}_1), \end{aligned}$$

there exists a Lyapunov function $V(\boldsymbol{e}) = \boldsymbol{e}^\top P(\boldsymbol{x}, H)\boldsymbol{e}$ *with positive definite, symmetric matrix* $P\colon \mathcal{X}\times\mathcal{H} \to \mathbb{R}^{2n\times 2n}, \forall H \in \mathcal{H}$, *so that* $\dot{V}(\boldsymbol{e}) < 0, \forall \boldsymbol{e} \neq 0$ *and* $\dot{V}(\boldsymbol{0}) = 0$.

Assumption 1 is fulfilled when we know a control law which asymptotically stabilizes the tracking error of the *known dynamics* of the system (1). Control approaches to ensure Assumption 1 include computed torque control for robotic manipulators [4], feedback linearization [16] or backstepping, among others.

The application of the control law (2) to the actual dynamics (1) leads to an unpredictable system behavior in general and manifests as a gap between theory and practice. A solution seems to be given by increasing feedback gains to penalize the tracking error. However, high feedback gains are often undesirable as it comes with an increase of the stiffness among other unwanted effects, see [13]. The stiffness can be defined locally by the generalized forces of (1) for a deflection $\Delta\boldsymbol{x}_1$ at some state $\boldsymbol{x}_s \in \mathcal{X}$. Let $T(\boldsymbol{x}, \boldsymbol{x}_d)\colon \mathcal{X}\times\mathcal{X} \to \mathbb{R}^n$ be the generalized forces of the closed loop system (3), i.e. $T(\boldsymbol{x}, \boldsymbol{x}_d) = M(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2 + C(\boldsymbol{x})\boldsymbol{x}_2 + G(\boldsymbol{x}_1) - \boldsymbol{h}_{\text{mb}}(\boldsymbol{x}, \boldsymbol{x}_d) - \boldsymbol{h}_{\text{fb}}(\boldsymbol{x})H\boldsymbol{e}$. Then, we can separate the stiffness at $\boldsymbol{x} = \boldsymbol{x}_s$ in two parts given by

$$\left\| \frac{\partial T(\boldsymbol{x}, \boldsymbol{x}_d)}{\partial \boldsymbol{x}_1} \right\|_{\boldsymbol{x}=\boldsymbol{x}_s} = \left\| t(\boldsymbol{x}, \boldsymbol{x}_d) - s(H, \boldsymbol{x}, \boldsymbol{e}) \right\|_{\boldsymbol{x}=\boldsymbol{x}_s} \quad (4)$$

with $t(\boldsymbol{x}, \boldsymbol{x}_d) = \partial(M(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2 + C(\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{x}_2 + G(\boldsymbol{x}_1) - \boldsymbol{h}_{\text{mb}}(\boldsymbol{x}, \boldsymbol{x}_d))/(\partial \boldsymbol{x}_1)$ and the feedback induced stiffness $s\colon \mathcal{H}\times\mathcal{X}\times\mathbb{R}^{2n} \to \mathbb{R}^n$ given by

$$s(H, \boldsymbol{x}, \boldsymbol{e}) = \frac{\partial \boldsymbol{h}_{\text{fb}}(\boldsymbol{x})H\boldsymbol{e}}{\partial \boldsymbol{x}_1}. \quad (5)$$

This term shows the relation between high feedback gains and an unintended increase of the stiffness in the system. An illustrative example can be found in the area of soft robotics, where a high-gain feedback controller would artificially increase the stiffness of the supposed to be soft structure. We focus here on the feedback induced stiffness $s(H, \boldsymbol{x}, \boldsymbol{e})$ as the model-based term $\boldsymbol{h}_{\text{mb}}$ is typically designed in a way to fulfill stability requirements as in Assumption 1.

A. Problem Setting

The objective of this paper is to satisfy performance specification and to reduce the feedback induced stiffness (5) imposed by the control law (2) by minimizing the feedback gain H. More detailed, we consider the system (1)

$$\dot{\boldsymbol{x}}_1 = \boldsymbol{x}_2,\ \dot{\boldsymbol{x}}_2 = \boldsymbol{f}_{\text{dyn}}(\boldsymbol{x}) + M^{-1}(\boldsymbol{x}_1)[\boldsymbol{\tau} + \boldsymbol{f}(\boldsymbol{x})], \quad (6)$$

where $\boldsymbol{f}_{\text{dyn}}(\boldsymbol{x}) = M^{-1}(\boldsymbol{x}_1)[C(\boldsymbol{x})\boldsymbol{x}_2 + G(\boldsymbol{x}_1)]$. Let $\hat{\boldsymbol{f}}\colon \mathcal{X} \to \mathbb{R}^n$ be an estimate of $\boldsymbol{f}$ and $\boldsymbol{\tau} = \boldsymbol{h}_{\text{mb}}(\boldsymbol{x}, \boldsymbol{x}_d) - \hat{\boldsymbol{f}}(\boldsymbol{x}) + \boldsymbol{h}_{\text{fb}}(\boldsymbol{x})H\boldsymbol{e}$. The control terms $\boldsymbol{h}_{\text{mb}}$ and $\boldsymbol{h}_{\text{fb}}$ are designed based on the known dynamics of (1) such that Assumption 1 is fulfilled. Then, the problem is to find a feedback $H \in \mathcal{H}$ such that for a desired maximum tracking error $\varepsilon \in \mathbb{R}_{>0}$, there exist constants $\gamma_1, \gamma_2 \in \mathbb{R}_{>0}$ and

$$\begin{aligned} &\min_{H\in\mathcal{H}} \|s(H, \boldsymbol{x}, \boldsymbol{e})\| \\ &\text{s.t. } \|\boldsymbol{e}(t)\| \leq \gamma_1 \exp^{-\gamma_2(t-t_0)} \|\boldsymbol{e}(0)\| + \varepsilon, \end{aligned} \quad (7)$$

where s is the feedback induced stiffness (5).

III. OLBB FRAMEWORK

We address the problem with an online learning-based balancing (OLBB) control law which adapts the model-based and feedback part such that the desired control performance is guaranteed with high probability. We start with an overview of the architecture of the framework followed by a detailed description of the involved parts.

A. Architecture

In Fig. 1, a block diagram of the proposed framework is shown. The standard approach is denoted in black whereas the blocks of our framework are colored in red. As introduced in (1), the system comprises a known and an unknown

[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 and $\|A\|$ denotes the Euclidean-induced matrix norm. The expression $\mathcal{N}(\mu, \Sigma)$ describes a normal distribution with mean μ and covariance Σ. The probability function is denoted by $\Pr$. The set $\mathbb{R}_{>0}$ denotes the positive real numbers and $\mathcal{C}^i$ the set of i-th times continuously differentiable functions.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:39:08 UTC from IEEE Xplore. Restrictions apply.

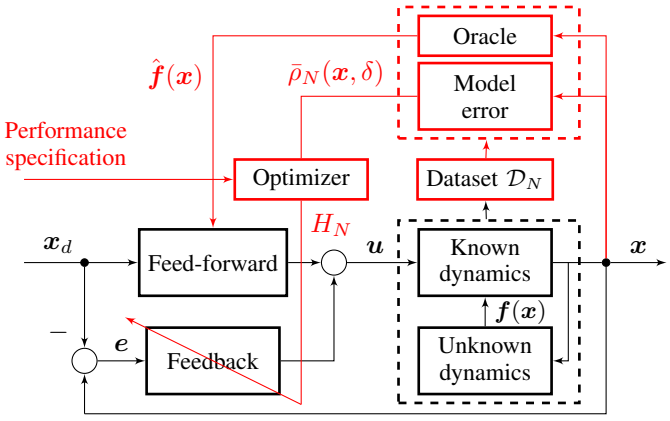


Fig. 1. Block diagram of the online learning-based balancing (OLBB) framework (red) on top of a nominal control approach (black)

dynamics part (black dashed box). Due to the unknown dynamics, the state $\boldsymbol{x}$ might not be following the desired trajectory using the nominal controller derived based on the known dynamics only. The key idea of the OLBB framework is to improve the model-based model by using an online learning-based oracle to predict the unknown dynamics and an adaptation of the feedback gains based on the model error. If the oracle predicts precisely, the feedback gains can be reduced to decrease the stiffness of the system. In contrast, a high model error leads to an increase of the feedback gains to satisfy the performance constraints. The quality of the oracle depends on a dataset which is created by online collected data. We introduce a novel balancing algorithm which allows to compute the necessary feedback gains online.

B. Oracle and Model Error

For the learning of the unknown dynamics of (1), we consider an oracle which predicts the values of $\boldsymbol{f}$ for a given state $\boldsymbol{x}$. For this purpose, the oracle collects $m \in \mathbb{N}$ training points of the system (1) such that a dataset

$$\mathcal{D}(t) = \{\boldsymbol{x}^{\{i\}}, \boldsymbol{y}^{\{i\}}\}_{i=1}^{m(t)}, \tag{8}$$

where $\boldsymbol{y} = \boldsymbol{\tau} - M(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2 - C(\boldsymbol{x})\boldsymbol{x}_2 - G(\boldsymbol{x}_1)$, is created. The dataset $\mathcal{D}(t)$ with $m\colon \mathbb{R}_+ \to \mathbb{N}$ can change over time t, such that the oracle allows online learning. For notational simplicity, we define $\mathcal{D}_N$ as the training set for the time interval $t = [t_N, t_{N+1})$ with m_N training points where $0 = t_0 < t_N < t_{N+1}, \ldots$. For example, the dataset $\mathcal{D}_N$ could contain all accumulated recorded data pairs $\{\boldsymbol{x}, \boldsymbol{y}\}$ up to time t_N, i.e. $\mathcal{D}_0 \subset \ldots \subset \mathcal{D}_N$. However, the dataset could also be disjoint if each set consists of new recorded data pairs. The time-dependent estimate of the oracle at time t_N is denoted by $\hat{\boldsymbol{f}}_N\colon \mathcal{X} \to \mathbb{R}^n$ to highlight the dependency on the corresponding dataset $\mathcal{D}_N$ such that

$$\hat{\boldsymbol{f}}_N(\boldsymbol{x}) = \hat{\boldsymbol{y}}|\boldsymbol{x}, \mathcal{D}_N, \tag{9}$$

where $\hat{\boldsymbol{y}} \in \mathbb{R}^n$ denotes the estimate of $\boldsymbol{y}$. Note that this structure also allows offline learning, e.g. the prediction of the oracle depends on previous collected data only and is likewise suitable for any hybrid online/offline approach. In order to quantify the error of the prediction (9), we introduce the following assumptions.

Assumption 2 *Consider an oracle with the predictions $\hat{\boldsymbol{f}}_N \in \mathcal{C}^0$ based on the dataset $\mathcal{D}_N$ (8). Let $\mathcal{X} \subset \mathbb{R}^{2n}$ be a compact set. There exists a finite function $\bar{\rho}_N\colon \mathcal{X} \times (0,1] \to \mathbb{R}_{\geq 0}$ such that the prediction error is bounded by*

$$\Pr\{\|\boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}_N(\boldsymbol{x})\| \leq \bar{\rho}_N(\boldsymbol{x}, \delta)\} \geq \delta \text{ with} \tag{10}$$

probability $\delta \in (0,1]$, $\forall \boldsymbol{x} \in \mathcal{X}$ and $\forall N \in \{1, \ldots, N_{end}\}$.

Assumption 3 *The number of datasets $\mathcal{D}_N$ is finite and there is a finite time between the change of datasets $\mathcal{D}_N, \mathcal{D}_{N+1}$, such that there exists a $T_{end} \in \mathbb{R}_{\geq 0}$ where $\mathcal{D}_N = \mathcal{D}_{N_{end}}, \forall t \geq T_{end}$.*

Assumption 2 is fulfilled if the model error is bounded in probability. For instance, this is true for bounded oracle predictions and bounded unknown disturbances such as wind fields or general unbounded forces such as friction over a closed set. However, globally unbounded forces cannot be considered due to the need of a bounded model error. Using expressive oracles such as Gaussian process models allow us to fulfill this assumption for any continuous $\boldsymbol{f}$ as shown in the next section. Assumption 3 is mildly restrictive since the number of sets is often naturally bounded due to finite computational power or memory limitations. Since the unknown function $\boldsymbol{f}$ in (1) is not time-dependent, long-life learning is typically not required. Next, we present Gaussian process models as a possible oracle.

C. Gaussian Process as Oracle

Gaussian process models have been proven as very powerful oracle for nonlinear function regression. For the prediction, we concatenate the m_N training points of $\mathcal{D}_N$ in an input matrix $X = [\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^{m_N}]$ and a matrix of outputs $Y^\top = [\tilde{\boldsymbol{y}}^1, \tilde{\boldsymbol{y}}^2, \ldots, \tilde{\boldsymbol{y}}^{m_N}]$, where the measurements $\tilde{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{\eta}$ are corrupted by Gaussian noise $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma I)$. A prediction $\boldsymbol{y}^* \in \mathbb{R}^n$ for the output at $\boldsymbol{x}^* \in \mathbb{R}^{2n}$ is given by

$$\begin{aligned} \mu_i(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}_N) &= \boldsymbol{k}(\boldsymbol{x}^*, X)^\top K^{-1} Y_{:,i} \\ \mathrm{var}_i(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}_N) &= k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{k}(\boldsymbol{x}^*, X)^\top K^{-1} \boldsymbol{k}(\boldsymbol{x}^*, X). \end{aligned} \tag{11}$$

for all $i \in \{1, \ldots, n\}$, where $Y_{:,i}$ denotes the i-th column of the matrix of outputs Y. The kernel $k\colon \mathbb{R}^{2n} \times \mathbb{R}^{2n} \to \mathbb{R}_+$ is a measure for the correlation of two states $(\boldsymbol{x}, \boldsymbol{x}')$. The function $K\colon \mathbb{R}^{2n \times m_N} \times \mathbb{R}^{2n \times m_N} \to \mathbb{R}^{m_N \times m_N}$ 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 $j', j \in \{1, \ldots, m_N\}$ with the delta function $\delta(j,j') = 1$ for $j = j'$ and zero, otherwise. The vector-valued function $\boldsymbol{k}\colon \mathbb{R}^{2n} \times \mathbb{R}^{2n \times m_N} \to \mathbb{R}^{m_N}$, with the elements $k_j = k(\boldsymbol{x}^*, X_{:,j})$ for all $j \in \{1, \ldots, m_N\}$, expresses the covariance between $\boldsymbol{x}^*$ and the input training data X. Based on (11), the normal distributed components $y_i^*|\boldsymbol{x}^*, \mathcal{D}_N$ are combined into a multi-variable Gaussian distribution with $\boldsymbol{\mu}(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}_N) = [\mu_1(\cdot), \ldots, \mu_n(\cdot)]^\top$ and $\Sigma(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}_N) = \mathrm{diag}\,[\mathrm{var}_1(\cdot), \ldots, \mathrm{var}_n(\cdot)]$. For simplicity, we consider identical kernels for each output dimension. However, the GP model can be easily adapted to different

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:39:08 UTC from IEEE Xplore. Restrictions apply.

kernels for each output dimension. With the introduced GP model, we are now addressing Assumption 2. To satisfy Assumption 2, additional assumptions on the unknown parts of the dynamics (1) must be introduced.

Assumption 4 *The kernel k is selected such that $\boldsymbol{f}$ has a bounded reproducing kernel Hilbert space (RKHS) norm on $\mathcal{X}$, i.e. $\|f_i\|_k < \infty$ for all $i = 1, \dots, n$.*

The norm of a function in a RKHS is a smoothness measure with respect to the metric of the kernel k, see [17]. Assumption 4 requires that the kernel must be selected in such a way that the functions $\boldsymbol{f}$ are elements of the associated RKHS. This first sounds paradoxical as the function $\boldsymbol{f}$ is unknown. However, there exist kernels, namely universal kernels, which can approximate any continuous function arbitrarily precisely on a compact set [18]. In summary, Assumptions 1 to 4 require that we know a stabilizing controller for the nominal system and use an oracle with bounded model error on a closed set for any finite number of data points.

Lemma 1 (adapted from [6]) *Consider the unknown functions $\boldsymbol{f}$ and a GP model satisfying Assumption 4. The model error is bounded by*

$$\Pr\{\| \boldsymbol{\mu}(\hat{\boldsymbol{f}}_N|\boldsymbol{x}, \mathcal{D}_N) - \boldsymbol{f}(\boldsymbol{x})\| \leq \|\boldsymbol{\beta}^\top \Sigma^{\frac{1}{2}}(\hat{\boldsymbol{f}}_N|\boldsymbol{x}, \mathcal{D}_N)\|\} \geq \delta$$

for $\boldsymbol{x} \in \mathcal{X}, \delta \in (0,1)$ with constant $\boldsymbol{\beta} \in \mathbb{R}^n$ given by $\beta_j = \sqrt{2\|f_j\|_k^2 + 300\gamma_j \ln^3\left(\frac{m_N+1}{1-\delta^{1/n}}\right)}$ and the information gain $\gamma_j = \max_{\boldsymbol{p}^{\{1\}},\dots,\boldsymbol{p}^{\{m_N+1\}} \in \mathcal{X}} \frac{1}{2}\log|I + \sigma_j^{-2}K(\boldsymbol{x}, \boldsymbol{x}')|$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \{\boldsymbol{p}^{\{1\}}, \dots, \boldsymbol{p}^{\{m_N+1\}}\}$.

Proof: It is a direct implication of [6, Lemma 1]. ■

With Assumption 4 and the fact, that universals kernels exist which generate bounded predictions, see [19], GP models can be used as oracle to fulfill Assumption 2. In this case, the prediction error bound is given by $\bar{\rho}_N(\boldsymbol{x}, \delta) := \|\boldsymbol{\beta}^\top \Sigma^{\frac{1}{2}}(\hat{\boldsymbol{f}}_N(\boldsymbol{x})|\boldsymbol{x}, \mathcal{D}_N)\|$ as introduced in Lemma 1.

D. Learning-based Control Law

In this section, we propose how the framework modifies the nominal control law to track the desired trajectory. First, we show the adaptation of the model-based part, followed by the tuning rule for the feedback gains. The model-based part $\boldsymbol{h}_{\text{mb}}(\boldsymbol{x}, \boldsymbol{x}_d)$ of the nominal control law is based on the known dynamics of the system (1). For low feedback gains, the model of the system needs to be as accurate as possible. Therefore, we use the prediction of the oracle (9) to improve the nominal model-based controller such that

$$\tilde{\boldsymbol{h}}_{\text{mb}}(\boldsymbol{x}, \boldsymbol{x}_d, t) = \boldsymbol{h}_{\text{mb}}(\boldsymbol{x}, \boldsymbol{x}_d) - \hat{\boldsymbol{f}}_N(\boldsymbol{x}), \tag{12}$$

where $\tilde{\boldsymbol{h}}_{\text{mb}}(\boldsymbol{x}, \boldsymbol{x}_d, t)\colon \mathcal{X} \times \mathcal{X} \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ describes the model-based control part of the proposed framework. The reason for the time-dependency of $\tilde{\boldsymbol{h}}_{\text{mb}}$ are the online updates of the oracle. As soon as the oracle is retrained at time t_N based on new data, the model-based (12) uses the updated prediction until the next update at time t_{N+1}. Note, that the time steps t_N for all $N \in \{1, \dots, N_{\text{end}}\}$ can be chosen arbitrarily and do not need to be defined beforehand. The feedback gains are adapted based on the accuracy of the model as defined in Assumption 2. If the prediction of the oracle is precise, the feedback gains can be lowered without increasing the tracking error. For this purpose, we introduce an adaptive feedback gain matrix $H_N \in \mathcal{H}$ depending on the model error $\bar{\rho}_N$, probability δ and dataset $\mathcal{D}_N$ for time t_N to t_{N+1}. Then, the feedback law is designed as

$$\tilde{\boldsymbol{h}}_{\text{fb}}(\boldsymbol{x}, \boldsymbol{x}_d, t) = \boldsymbol{h}_{\text{fb}}(\boldsymbol{x}) H_N \boldsymbol{e} \tag{13}$$

with $\boldsymbol{h}_{\text{fb}}$ of Assumption 1, the tracking error $\boldsymbol{e}$, and the feedback gain matrix H_N given by

$$\begin{aligned} &\min_{H_N \in \mathcal{H}} \|H_N\| \text{ for } N \in \{1, \dots, N_{\text{end}}\} \\ &\text{s.t. } \max_{\boldsymbol{x} \in \mathcal{X}} \frac{\|P_{\text{r}}(\boldsymbol{x}, H_N)\|}{\underline{\lambda}[B(\boldsymbol{x}, H_N)]} \kappa[P(\boldsymbol{x}, H_N)]^{\frac{1}{2}} \bar{\rho}_N(\boldsymbol{x}, \delta) \leq \varepsilon, \end{aligned} \tag{14}$$

where $\kappa(\cdot)$ denotes the condition number of the matrix and $\underline{\lambda}(\cdot), \bar{\lambda}(\cdot)$ the smallest/largest eigenvalue, respectively. The matrix P is defined in Assumption 1, $P_{\text{r}} \in \mathbb{R}^{2n \times n}$ denotes the right submatrix of P defined by $P = [P_{\text{l}}, P_{\text{r}}]$ and $B\colon \mathcal{X} \times \mathcal{H} \to \mathbb{R}^{2n \times 2n}$ is given by the Lyapunov drift $\dot{V}_N(\boldsymbol{e}) = -\boldsymbol{e}^\top B(\boldsymbol{x}, H_N)\boldsymbol{e}$ of the nominal system. The matrix B is positive definite and exists due to the asymptotic stability of the nominal system as defined in Assumption 1. Note that B_N depends on the known dynamics only. The constraint (14) contains three separated factors. The fraction is the ratio between the term P_r, that creates the drift $\dot{V}_N$ based on the model error, and $\underline{\lambda}(B)$ that indicates the decay rate of the Lyapunov function. As the drift increases linearly and the decay rate quadratically with the feedback gain H_N, the term tends to zero. The condition number of P typically increase with the feedback gain as it is the connection between the ellipse-like level sets of the Lyapunov function and the ball of the maximum desired tracking error (7). Even though we can not guarantee the feasibility of (14) in general, we show later in Lemma 2 that for commonly-used control approaches for mechanical systems a solution for (14) always exists. Finally, the last factor $\bar{\rho}_N(\boldsymbol{x}, \delta)$ depends on the model error and decreases with more data in general. Taken together, the proposed control law with improved model (12) and adaptive feedback (13) is condensed to

$$\tilde{\boldsymbol{\tau}}(t) = \tilde{\boldsymbol{h}}_{\text{mb}}(\boldsymbol{x}, \boldsymbol{x}_d, t) + \tilde{\boldsymbol{h}}_{\text{fb}}(\boldsymbol{x}, \boldsymbol{x}_d, t). \tag{15}$$

E. OLBB Algorithm

After the detailed presentation of the steps above, we summarize the entire procedure in Algorithm 1. At the beginning, the maximum desired tracking error ε and the maximum number N_{end} of datasets used by the oracle are set. This choice typically depends on how many data points can be collected which in turn is bounded due to memory and computational limitations. For example, if m_N data points are collected between t_N and t_{N+1}, the maximum number N_{end} of datasets is set to the greatest integer less than or equal to $\bar{m}/m_N$, where $\bar{m}$ is the upper limit of data points due to memory and computational limitations.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:39:08 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Online learning-based balancing

input: max. tracking error ε, max. datasets N_{end}
initialize: $N = 0, \mathcal{D}_0$
$H_0 \leftarrow$ Compute feedback gain based on $\varepsilon, \bar{\rho}_0, \delta$ (13)
initialize: Control law $\tilde{\boldsymbol{\tau}}(t)$ with $\mathcal{D}_0, \hat{\boldsymbol{f}}_0$ (15)
while $N < N_{\text{end}}$ **do**
  $N \leftarrow N + 1$
  $\mathcal{D}_N \leftarrow$ Collect m_N data points (8)
  Train oracle with $\mathcal{D}_N$ (9)
  $\bar{\rho}_N \leftarrow$ Get model error (10)
  $H_N \leftarrow$ Compute feedback gain based on $\varepsilon, \bar{\rho}_N, \delta$ (14)
  Use control law $\tilde{\boldsymbol{\tau}}(t)$ with $\mathcal{D}_N, \hat{\boldsymbol{f}}_N$ (15)
end while

The oracle is initialized with a dataset that can originate from an offline identification of the system or simulation data. However, without preliminary data, the dataset $\mathcal{D}_0$ is empty and the initial estimate $\hat{\boldsymbol{f}}_0 = 0$. Then, the initial feedback gains are computed based on (13). The control law (15) based on $\hat{\boldsymbol{f}}_0$ is used to control the system. A number of m_N data points from the actual system (1) as defined in (8) are collected and stored in the dataset $\mathcal{D}_N$. Then, the oracle is retrained on the dataset $\mathcal{D}_N$ and the feedback gains H_N are recomputed. At arbitrary time t_N, the control law is updated by using the prediction $\hat{\boldsymbol{f}}_N$ and feedback gains H_N.

Note that until time t_N, the control law makes use of $\hat{\boldsymbol{f}}_{N-1}$ and H_{N-1} such that there are no requirements on the computation time for the retraining of the oracle and the computation of the feedback gains. This process is repeated until the maximum number N_{end} of datasets is reached.

F. Theoretical Analysis

In the following, we introduce the formal results for the stability and performance guarantees of the proposed framework. First, we provide a proof of the boundedness of the tracking error. Then, the solution of an optimization problem is proven to be sufficient to guarantee the specified performance of the closed-loop.

Theorem 1 *Consider a system given by* (1) *with unknown dynamics $\boldsymbol{f}$ and the existence of an oracle satisfying Assumptions 2 and 3. Let $H_N \in \mathcal{H}$ be a matrix based on the dataset $\mathcal{D}_N$. With Assumption 1, the control law*

$$\boldsymbol{\tau} = \underbrace{\boldsymbol{h}_{\text{mb}}(\boldsymbol{x}, \boldsymbol{x}_d) - \hat{\boldsymbol{f}}_N(\boldsymbol{x})}_{\tilde{\boldsymbol{h}}_{\text{mb}}(\boldsymbol{x}, \boldsymbol{x}_d, t)} + \underbrace{\boldsymbol{h}_{\text{fb}}(\boldsymbol{x}) H_N \boldsymbol{e}}_{\tilde{\boldsymbol{h}}_{\text{fb}}(\boldsymbol{x}, \boldsymbol{x}_d, t)} \tag{16}$$

guarantees that the tracking error is uniformly ultimately bounded in probability for arbitrary large $\delta \in (0, 1)$ by $\Pr\{\|\boldsymbol{e}(t)\| \leq b_{\text{end}} \max_{\boldsymbol{x} \in \mathcal{X}} \bar{\rho}_{\text{end}}(\boldsymbol{x}, \delta), \forall t \geq T_\varepsilon\} \geq \delta$ on $\mathcal{X}$ with $b_{\text{end}}, T_\varepsilon \in \mathbb{R}_+$.

We prove the stability of the closed-loop with the proposed control law with multiple Lyapunov functions, where the N-th function is active when the oracle predicts based on $\mathcal{D}_N$.

Proof: Using the proposed control law for the dynamics (1) leads to tracking error dynamics given by $\dot{\boldsymbol{e}}_2 = \boldsymbol{f}_{\text{dyn}}(\boldsymbol{x}) + M^1(\boldsymbol{x}_1)(\boldsymbol{h}_{\text{mb}}(\boldsymbol{x}, \boldsymbol{x}_d) + \boldsymbol{h}_{\text{fb}}(\boldsymbol{x}) H_N \boldsymbol{e}) + \boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}_N(\boldsymbol{x}) - \boldsymbol{x}_{d,2}$. The Lyapunov function of Assumption 1 is used with the time-dependent feedback gain matrix H_N such that $V_N(\boldsymbol{e}) = \boldsymbol{e}^\top P(\boldsymbol{x}, H_N)\boldsymbol{e}$. Note that this leads to a switching Lyapunov function but due to a finite number of switching events, the switching between stable systems cannot lead to an unbounded trajectory. The time-derivative of the Lyapunov function $\dot{V}_N(\boldsymbol{e}) = \boldsymbol{e}^\top P(\boldsymbol{x}, H_N)\boldsymbol{e}$ is given by

$$\begin{aligned}\dot{V}_N(\boldsymbol{e}) = {} & \boldsymbol{e}^\top P(\boldsymbol{x}, H_N)\dot{\boldsymbol{e}} + \boldsymbol{e}^\top \dot{P}(\boldsymbol{x}, H_N)\boldsymbol{e} \\ & + \boldsymbol{e}^\top P_{\text{r}}(\boldsymbol{x}, H_N)[\boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}_N(\boldsymbol{x})],\end{aligned} \tag{17}$$

where $P_{\text{r}} \in \mathbb{R}^{2n \times n}$ denotes the right submatrix of P. With dynamics (1) and Assumption 1, there exists a positive definite matrix function $B \colon \mathcal{X} \times \mathcal{H} \to \mathbb{R}^{2n \times 2n}$ such that equation (17) can be rewritten as $\dot{V}_N(\boldsymbol{e}) = -\boldsymbol{e}^\top B(\boldsymbol{x}, H_N)\boldsymbol{e} + \boldsymbol{e}^\top P_{\text{r}}(\boldsymbol{x}, H_N)[\boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}_N(\boldsymbol{x})]$. Using Assumption 2, the Lyapunov derivative is upper bounded by

$$\Pr\{\dot{V}_N \leq -\boldsymbol{e}^\top B(\boldsymbol{x}, H_N)\boldsymbol{e} + \|\boldsymbol{e}^\top P_{\text{r}}(\boldsymbol{x}, H_N)\| \bar{\rho}_N(\boldsymbol{x}, \delta)\} \geq \delta$$

analogously to [6], [10]. Due to the dominance of the quadratic part, the derivative (17) is negative outside a neighborhood of zero. Thus, the tracking error is uniformly ultimately bounded in probability. Since Assumption 3 only allows a finite number of switches, there exists a time $T_{\text{end}} \in \mathbb{R}_{\geq 0}$ such that $\mathcal{D}_N = \mathcal{D}_{N_{\text{end}}}$ for all $t \geq T_{\text{end}}$. Thus, there exists a time $T_\varepsilon \geq T_{\text{end}}$ such that $\Pr\{\|\boldsymbol{e}(t)\| \leq \infty, \forall t \geq T_\varepsilon\} \geq \delta$. ■

At this point, we have shown that the tracking error $\boldsymbol{e}(t)$ of the actual system (1) is bounded in probability. The exact time T_ε can be computed based on the initial tracking error $\boldsymbol{e}(0)$, see [20]. Next, the minimum feedback gain is computed such that the tracking error $\boldsymbol{e}(t)$ fulfills the performance specifications, i.e. the tracking error $\boldsymbol{e}$ converges exponentially fast to the desired tracking error ε.

Theorem 2 *Consider the system given by* (1) *with unknown dynamics $\boldsymbol{f}$ and the existence of an oracle satisfying Assumptions 2 and 3. For any desired maximum tracking error $\varepsilon > 0$, the control law* (16) *with gain H_N satisfying*

$$\max_{\boldsymbol{x} \in \mathcal{X}} \frac{\|P_r(\boldsymbol{x}, H_N)\|}{\underline{\lambda}[B(\boldsymbol{x}, H_N)]} \kappa[P(\boldsymbol{x}, H_N)]^{\frac{1}{2}} \bar{\rho}_N(\boldsymbol{x}, \delta) \leq \varepsilon \tag{18}$$

for all $N \in \{1, \ldots, N_{end}\}$ guarantees that the tracking error is bounded in probability and exponentially converges such that $\Pr\{\|\boldsymbol{e}(t)\| \leq \gamma_1 \exp^{-\gamma_2 (t - t_0)} \|\boldsymbol{e}(0)\| + \varepsilon\} \geq \delta$.

Proof: The quadratic-like Lyapunov function, as required by Assumption 1, $V_N(\boldsymbol{e}) = \boldsymbol{e}^\top P(\boldsymbol{x}, H_N)\boldsymbol{e}$ is lower and upper bounded by $\alpha_1(\|\boldsymbol{e}\|) \leq V_N(\boldsymbol{e}) \leq \alpha_2(\|\boldsymbol{e}\|)$, where $\alpha_1(r) = \underline{\lambda}[P(\boldsymbol{x}, H_N)]r^2$ and $\alpha_2(r) = \bar{\lambda}[P(\boldsymbol{x}, H_N)]r^2$. Thus, the maximum tracking error $b_N \in \mathbb{R}_{\geq_0}$ such that $\Pr\{\|\boldsymbol{e}(t)\| \leq b_N, \forall t \geq T_\varepsilon\} \geq \delta$ is given by

$$b_N = c \max_{\boldsymbol{x} \in \mathcal{X}} \sqrt{\frac{\bar{\lambda}[P(\boldsymbol{x}, H_N)]}{\underline{\lambda}[P(\boldsymbol{x}, H_N)]}} = c \max_{\boldsymbol{x} \in \mathcal{X}} \kappa[P(\boldsymbol{x}, H_N)]^{\frac{1}{2}}, \tag{19}$$

where $\dot{V}_N < 0$ for all $\|\boldsymbol{x}\| > c \in \mathbb{R}$ and $\kappa(\cdot)$ denotes the condition number of the matrix. Using Theorem 1,

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:39:08 UTC from IEEE Xplore. Restrictions apply.

the evolution of the Lyapunov function is negative with probability δ for all $\|\boldsymbol{e}\| > \max_{\boldsymbol{x}\in\mathcal{X}} \frac{\|P_\mathrm{r}(\boldsymbol{x},H_N)\|}{\underline{\lambda}[B(\boldsymbol{x},H_N)]}\bar{\rho}_N(\boldsymbol{x},\delta)$, where $P_\mathrm{r} \in \mathbb{R}^{2n\times n}$ denotes the right submatrix of P. As a consequence, the tracking error exponentially converges such that $\Pr\{\|\boldsymbol{e}(t)\| \leq \gamma_1 \exp^{-\gamma_2(t-t_0)}\|\boldsymbol{e}(0)\| + \varepsilon\} \geq \delta$ with

$$\gamma_1 \leq \max_{\boldsymbol{x}\in\mathcal{X}} \kappa[P(\boldsymbol{x},H_N)]^{\frac{1}{2}}, \gamma_2 \geq \frac{\|P_\mathrm{r}(\boldsymbol{x},H_N)\|}{2\underline{\lambda}[B(\boldsymbol{x},H_N)]} \tag{20}$$

for all $H_N \in \mathcal{H}$. That completes the proof. ∎

Intuitively, the radius of the bound depends on the model error $\bar{\rho}_N$. If a perfect oracle was available, the tracking error would converge to zero for any feedback gain H_N. In case of model error, the feedback gains need to be increased such that (18) is fulfilled to guarantee a maximum tracking error. Next, we exemplary show the existence of a solution for (18) and how to find it in case of linear *nominal* closed-loop systems and diagonal feedback gains. Note that a linear nominal closed-loop does not imply any further assumptions on the unknown force $\boldsymbol{f}$, i.e. it can still be nonlinear.

Lemma 2 *Let $\mathcal{H} = \{H = [h_1 I, h_2 I] | h_1, h_2 > 0\}$ be the set of feedback gains and the* nominal *closed-loop dynamics* (3) *be linear such that*

$$\dot{\boldsymbol{e}} = \begin{bmatrix} 0 & I \\ -h_1 I & -h_2 I \end{bmatrix} \boldsymbol{e} =: A(H)\boldsymbol{e}. \tag{21}$$

Then, for all $\varepsilon > 0$, there exists a $\underline{h} \in \mathbb{R}_{>0}$ such that all $H = [h_1 I, h_2 I]$ with $h_1, h_2 \geq \underline{h}$ fulfill the inequality (18).

Proof: By Assumption 1, the linear system (21) is asymptotically stable such that there exists a matrix $P \in \mathbb{R}^{2n\times 2n}$ which satisfies $A^\top P + PA = -I$. With the simplification $h = h_1 = h_2$, we can compute P by $P = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix}$ with $p_{11} = 1 + 1/(2h), p_{12} = 1/(2h)$ and $p_{22} = (h+1)/(2h^2)$. Calculating the matrix norm and the eigenvalues of P in (18) leads to

$$\frac{\|P_\mathrm{r}(\boldsymbol{x},H)\|}{\underline{\lambda}[B(\boldsymbol{x},H)]}\kappa[P(\boldsymbol{x},H)]^{\frac{1}{2}} = \left((p_{12}^2+p_{22}^2)\frac{\alpha(P)^{1/2}+p_{11}+p_{22}}{\alpha(P)^{-1/2}+p_{11}+p_{22}}\right)^{\frac{1}{2}}$$

with $\alpha(P) = p_{11}^2 - 2p_{11}p_{22} + p_{22}^2 + 4p_{12}^2$, which converges to zero for $h \to \infty$. With the bounded model error $\bar{\rho}(\boldsymbol{x})$ of Assumption 2, there exists a $\underline{h} \in \mathbb{R}_{>0}$ such that all $H = [h_1 I, h_2 I]$ with $h \geq \underline{h}$ fulfill the inequality (18). ∎

Thus, by increasing the feedback gains, a solution of (18) in case of a linear nominal closed-loop dynamics exists. Note that in case no solution of (18) for arbitrary nominal closed-loop dynamics might be achieved, the tracking error remains bounded according to Theorem 1.

IV. Case Study

In this case study, we apply the proposed framework to the control of a 2-link robotic manipulator. We assume point masses for the links of $m_{\text{link},1} = m_{\text{link},2} = 1\,\text{kg}$, which are located in the center of each link and the joints are without mass. The length of the links is set to $l_{\text{link},1} = l_{\text{link},2} = 1\,\text{m}$. Gravity is approximated by $g = 10\,\text{ms}^{-2}$. Following [14, Page 164], the system dynamics (1) of the robot with joint angles x_1, x_2 and angular velocities x_3, x_4 are given by

$$\begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \underbrace{M^{-1}(\boldsymbol{x})[C(\boldsymbol{x}) + G(\boldsymbol{x})]}_{\boldsymbol{f}_\text{dyn}} + M^{-1}(\boldsymbol{x})\boldsymbol{\tau} + \boldsymbol{f}(\boldsymbol{x}) \tag{22}$$

with inertia $M \in \mathbb{R}^{2\times 2}$, Coriolis force $C \in \mathbb{R}^2$ and gravity $G \in \mathbb{R}^2$. The unknown dynamics $\boldsymbol{f}(\boldsymbol{x})$ is nonlinear friction $f_\text{fric}(v) = 0.3\tanh(100v)[0.1 + 0.2\exp(-v^2/0.04^2) + 0.1v]$ depending on the velocity $v \in \mathbb{R}$ according to the Stribeck curve, such that $\boldsymbol{f}(\boldsymbol{x}) = [f_\text{fric}(x_3), f_\text{fric}(x_4)]$. Based on the known system dynamics of (22), we can derive a stabilizing control law using computed torque control. Then, the control law is given by

$$\boldsymbol{\tau} = \underbrace{M(\boldsymbol{x})\dot{\boldsymbol{x}}_{d,3:4}^\top + C(\boldsymbol{x}) + G(\boldsymbol{x})}_{\boldsymbol{h}_\text{mb}} \underbrace{- M(\boldsymbol{x})}_{\boldsymbol{h}_\text{fb}} H\boldsymbol{e}, \tag{23}$$

where $\dot{\boldsymbol{x}}_{d,3:4} = [\dot{\boldsymbol{x}}_{d,3}, \dot{\boldsymbol{x}}_{d,4}]^\top$ denotes the desired angular acceleration and $H = [h_1 I, h_2 I] \in \mathbb{R}^{2\times 4}$ the feedback gain matrix. The closed-loop dynamics for the *nominal* system of (22) is the same as in (21), which is an asymptotically stable linear system such that Assumption 1 is satisfied. As oracle, we use a GP with squared exponential kernel as presented in Section III-C and 3-times the posterior standard deviation, i.e. $\boldsymbol{\beta} = [3,3,3,3]^\top$, as model error bound. The maximum number of datasets is set to $N_\text{end} = 20$ and the initial set $\mathcal{D}_0$ is empty. We assume $\varepsilon = 0.1$ as maximum desired tracking error on the closed set $\mathcal{X} = [-\pi/2, \pi/2]^4$. Based on Theorem 2, the initial feedback gains H_0 are computed to $h_1 = 11.2$ and $h_2 = 9.3$ using a sequential quadratic programming optimizer. The initial angles and velocities are set to $\boldsymbol{x}_0 = [0.5, 0.5, 0, 0]^\top$. In Fig. 2, the desired trajectory (dashed) based on an arbitrarily chosen spline curve and actual trajectory (solid) is drawn. Every $0.1\,\text{s}$, a training point as defined in (8) is recorded. The recorded dataset is used to online retrain the GP model and the hyperparameters by means of the likelihood function every $0.5\,\text{s}$. Thus, the empty dataset $\mathcal{D}_0$ is used until $0.5\,\text{s}$ and, afterwards, the dataset $\mathcal{D}_1$ with $m_1 = 10$ training points. As the model error changes for the retrained GP model, the feedback gain matrix H_N is also updated by a sequential quadratic programming optimizer every $0.5\,\text{s}$. The required Lyapunov function $V_N(\boldsymbol{e}) = \boldsymbol{e}^\top P_N \boldsymbol{e}$ is computed by solving the Lyapunov equation for the system matrix (21). The resulting gains are depicted in the top plot of Fig. 3 with the Euclidean-induced norm of H^1 in blue and H^2 in

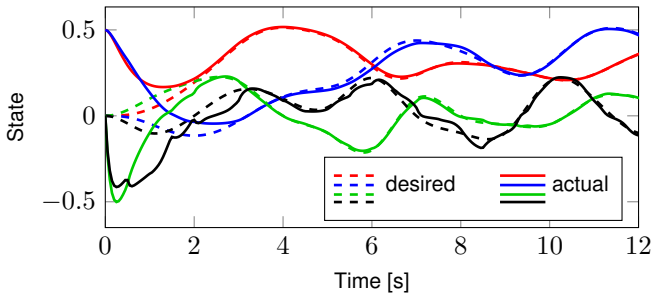


Fig. 2. The actual trajectory (solid) of the system converges to the desired trajectory (dashed).

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:39:08 UTC from IEEE Xplore. Restrictions apply.

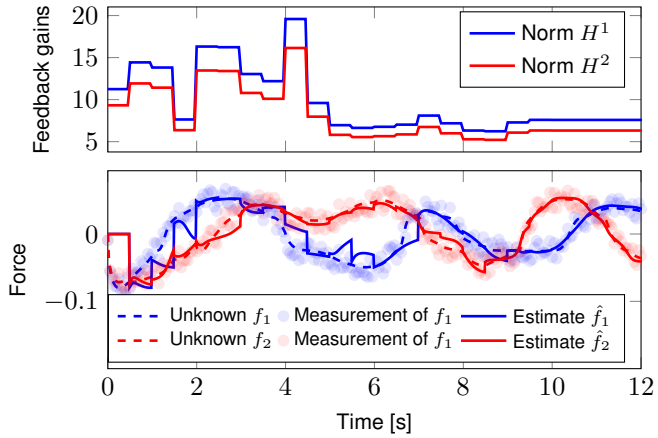


Fig. 3. Top: Adapted feedback gains based on the model error. Bottom: The unknown force $\boldsymbol{f}$ (dashed) and the estimates of the GP model (solid) which are updated with new training data (dots) every $0.5\,\mathrm{s}$.

red. Since additional training data can also lead to worse model accuracy, the gains are not always decreasing over time. However, the GP model becomes more accurate at some point such that the feedback gains can be significantly reduced. At $10\,\mathrm{s}$, the maximum number of datasets is reached and, thus, the model is not further updated and the feedback gains are kept constant. The value for both components of the unknown force $\boldsymbol{f}$ (dashed) and the measurements of $\dot{\boldsymbol{x}}$ (circles), which are corrupted by Gaussian noise with zero mean and a variance of 0.01^2, are visualized in the bottom plot of Fig. 3. The GP model (solid) estimates the unknown force based on the measurements. Note that in time intervals with accurate estimation, the feedback gains are low and vice versa. The resulting tracking error of the proposed method (blue) is visualized in the top plot of Fig. 4. As desired, the tracking error is less than $\varepsilon = 0.1$ for a $t > T_\varepsilon$. The tracking error for a standard computed torque controller with the feedback gains equivalent to the mean of the varying feedback gains shown in Fig. 3 is plotted in red. The proposed method satisfies the desired error (dashed) after $2.1\,\mathrm{s}$. The bottom plot of Fig. 4 shows the stiffness of the robot over time. First, the proposed method (blue) results in a higher stiffness to satisfies the performance constrains due to an inaccurate model. With more training data, the model accuracy improves which allows to reduce the feedback gains. Finally, the stiffness remains significantly lower in comparison to a standard computed torque approach (red).

Conclusion

In this article, we present a framework for online learning-based balancing (OLBB) of model-based and feedback control for high-performance tracking control of mechanical systems with reduced feedback gains. We modify a nominal control law which is derived based on the known dynamics of the system. An oracle improves the predictive model of the control law by online updates with new training data and the model error adapts the feedback gains. The framework allows to compute reduced feedback gains such that a desired performance with exponential convergence of the tracking error is satisfied.

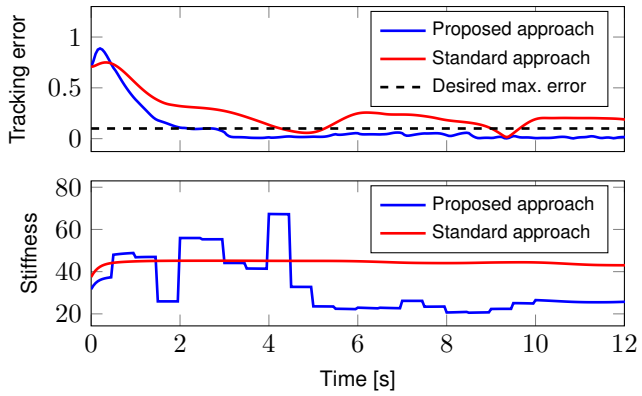


Fig. 4. Tracking error (top) and stiffness (bottom) of the robot with computed torque control (red) and the proposed method (blue). Note that not only the tracking error is reduced but also the stiffness after $4.5\,\mathrm{s}$.

References

[1] R. Ortega, J. A. L. Perez, P. J. Nicklasson, and H. J. Sira-Ramirez, *Passivity-based control of Euler-Lagrange systems: mechanical, electrical and electromechanical applications*. Springer Science & Business Media, 2013.

[2] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical human–robot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, 2008.

[3] 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.

[4] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*, vol. 3. wiley New York, 2006.

[5] A. Amiri Moghadam and K. e. a. Torabi, "Control-oriented modeling of a polymeric soft robot," *Soft Robotics*, vol. 3, no. 2, 2016.

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

[7] 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.

[8] 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.

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

[10] J. Umlauft and S. Hirche, "Feedback linearization based on Gaussian processes with event-triggered online learning," *IEEE Transactions on Automatic Control*, vol. 65(10), p. 4154–4169, 2020.

[11] 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.

[12] N. T. Alberto, M. Mistry, and F. Stulp, "Computed torque control with variable gains through Gaussian process regression," in *Proc. of the International Conference on Humanoid Robots*, pp. 212–217, 2014.

[13] C. Della Santina, M. Bianchi, G. Grioli, F. Angelini, M. Catalano, M. Garabini, and A. Bicchi, "Controlling soft robots: balancing feedback and feedforward elements," *IEEE Robotics & Automation Magazine*, vol. 24, no. 3, pp. 75–83, 2017.

[14] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[15] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[16] B. Charlet, J. Lévine, and R. Marino, "On dynamic feedback linearization," *Systems & Control Letters*, vol. 13, no. 2, pp. 143–151, 1989.

[17] G. Wahba, *Spline models for observational data*. SIAM, 1990.

[18] I. Steinwart and A. Christmann, *Support vector machines*. Springer Science & Business Media, 2008.

[19] T. Beckers and S. Hirche, "Stability of Gaussian process state space models," in *Proc. of the European Control Conference*, pp. 2275–2281, 2016.

[20] H. K. Khalil, *Noninear Systems*. Prentice-Hall, New Jersey, 1996.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:39:08 UTC from IEEE Xplore. Restrictions apply.