MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Learning-Based Iterative Modular Adaptive Control for Nonlinear Systems

Benosman, Mouhacine; Farahmand, Amir-massoud; Xia, Meng

TR2018-108 August 17, 2018

Abstract

In this paper we study the problem of adaptive trajectory tracking control for a class of nonlinear systems with structured parametric uncertainties. We propose to use an iterative modular approach: we first design a robust nonlinear state feedback that renders the closed-loop input-to-state stable ISS). Here, the input is considered to be the estimation error of the uncertain parameters, and the state is considered to be the closed loop output tracking error. Next, we propose an iterative adaptive algorithm, where we augment this robust ISS controller with an iterative data-driven learning algorithm to estimate online the parametric uncertainties of the model. We implement this method with two different learning approaches. The first one is a datadriven multi-parametric extremum seeking (MES) method, which guarantees local convergence results, and the second is a Bayesian optimization-based method called Gaussian Process Upper Confidence Bound (GPUCB), which guarantees global results in a compact search set. The combination of the ISS feedback and the data-driven learning algorithms gives a learning-based modular indirect adaptive controller. We show the efficiency of this approach on a two-link robot manipulator numerical example.

International Journal of adaptive control and signal processing

INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING
Int. J. Adapt. Control Signal Process. 0000; **00**:1–30
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/acs

Learning-Based Iterative Modular Adaptive Control for Nonlinear Systems

Mouhacine Benosman[1], Amir-massoud Farahmand[1], Meng Xia[2]

[1]*Mitsubishi Electric Research Laboratories, 201 Broadway Street, Cambridge, MA 02139, USA (Email: m_benosman@ieee.org),*[2]*Mathworks, USA.*

SUMMARY

In this paper we study the problem of adaptive trajectory tracking control for a class of nonlinear systems with structured parametric uncertainties. We propose to use *an iterative modular approach*: we first design a robust nonlinear state feedback that renders the closed-loop input-to-state stable (ISS). Here, the input is considered to be the estimation error of the uncertain parameters, and the state is considered to be the closed-loop output tracking error. Next, we propose an iterative adaptive algorithm, where we augment this robust ISS controller with an iterative data-driven learning algorithm to *estimate online the parametric uncertainties of the model*. We implement this method with two different learning approaches. The first one is a data-driven multi-parametric extremum seeking (MES) method, which guarantees local convergence results, and the second is a Bayesian optimization-based method called Gaussian Process Upper Confidence Bound (GP-UCB), which guarantees global results in a compact search set. The combination of the ISS feedback and the data-driven learning algorithms gives a *learning-based modular indirect adaptive controller*. We show the efficiency of this approach on a two-link robot manipulator numerical example. Copyright © 0000 John Wiley & Sons, Ltd.

Received . . .

1. INTRODUCTION

Classical adaptive methods can be classified into two main approaches: 'direct' approaches, where the controller is updated to adapt to the process, and 'indirect' approaches, where the model is updated to better reflect the actual process. Many adaptive methods have been proposed over the years for linear and nonlinear systems; we cannot possibly cite them all. Instead we refer the reader to e.g., [1, 2, 6, 7] and the references therein for more detail. Of particular interest to us is the indirect modular approach to adaptive nonlinear control, e.g., [6]. In this approach, first the controller is designed by assuming that all the parameters are known and then an identifier is used to guarantee

boundedness or asymptotic convergence of the estimation error. When the identifier is based on a data-driven learning algorithm, which is independent of the designed controller, the approach is called 'learning-based', e.g. [3]. In this line of research in adaptive control we can cite the following references [3, 4, 5],[8] to [44].

For example, in the neural network (NN)-based modular adaptive control design, the idea is to write the model of the system as a combination of a known part and an unknown part (a.k.a. the disturbance part). The NN is then used to approximate the unknown part of the model. Finally, a controller based on both the known and the NN-estimate of the unknown part is determined to realize some desired regulation or tracking performance, e.g., [8, 10, 13].

In this work, we build upon this type of modular learning-based adaptive design and provide a framework that combines iterative data-driven learning methods and robust model-based nonlinear control. We propose an iterative learning-based modular indirect adaptive controller, in which iterative data-driven learning algorithms are used *to estimate, in closed-loop, the uncertain parameters of the model*. Here, we focus on the class of nonlinear systems affine in the control, and propose to use two different data-driven learning algorithms: The first one is a data-driven multi-parametric extremum seeking (MES) method, which guarantees local convergence results, and the second is a Bayesian optimization-based method called Gaussian Process Upper Confidence Bound (GP-UCB), which guarantees global results in a compact search set.

We want to underline that the main difference with the existing model-based indirect adaptive control methods is the fact that we do not use the model to design the uncertainty parameters estimation filters. Indeed, model-based indirect adaptive controllers are based on parameters estimators designed using the system's model, e.g., the X-swapping methods presented in [6], where gradient descent filters obtained using the systems dynamics are designed to estimate the uncertain parameters. We argue that because we do not use the system's dynamics to design uncertainties estimation filters we have less restrictions on the type of uncertainties that we can estimate, e.g., uncertainties appearing nonlinearly can be estimated with the proposed approach, see [25, 33] for some earlier results on a mechatronics application. We also show (cf. Section 5) that with the proposed approach we can estimate at the same time a vector of linearly dependent uncertainties, a case which cannot be straightforwardly solved using model-based filters, e.g., refer to [34] where it is shown that the X-swapping model-based method fails to estimate a vector of linearly dependent model coefficients.

MES is a data-driven control approach with well-known convergence properties, and has been analyzed in many textbooks and papers, e.g., [35, 36, 37], [38] to [42], and references therein. This makes MES a good candidate for the data-driven estimation part of our modular adaptive controller, as already shown in some of our preliminary results in [30, 31, 32]. However, one of the main limitations with dither-based MES is the convergence to local minima. To improve

Prepared using acsauth.cls
DOI: 10.1002/acs

this part of the controller, we introduce another data-driven learning algorithm in the estimation part of the adaptive controller. We propose in this paper to use the GP-UCB learning algorithm, a Bayesian optimization method [46]. These methods solve the exploration-exploitation problem in the continuous armed bandit problem, thus they can be classified as a non-associative reinforcement learning (RL) algorithm, e.g., [47]. Contrary to the MES algorithm, GP-UCB is guaranteed to reach the global minima under certain mild assumptions.

One point worth mentioning at this stage is that comparing to 'pure' data-driven controllers, e.g., pure MES or data-driven RL algorithms, the proposed control has a different goal. The available data-driven controllers are meant for output or state regulation, i.e., solving a static optimization problem. In contrast, we propose to use data-driven learning to complement a model-based nonlinear control *to estimate the unknown parameters of the model*, which means that the control goal, i.e., state or output trajectory tracking is handled by the model-based controller. The learning algorithm is used to improve the tracking performance of the model-based controller, and once the learning algorithm has converged, one can carry on using the nonlinear model-based feedback controller alone, i.e., without the need of the learning algorithm. Furthermore, due to the fact that we are merging together a model-based control with a data-driven learning algorithm, we believe that this type of controller can converge faster to an optimal performance, comparatively to the pure data-driven controller, since by 'partly' using a model-based controller, we are taking advantage of the partial information given by the physics of the system, whereas the pure data-driven algorithms assume no knowledge about the system, and thus start the search for an optimal control signal from scratch.

Similar ideas of merging model-based control and MES has been proposed in [21, 22, 24, 25, 26, 33, 30, 31, 32]. For instance, extremum seeking is used to complement a model-based controller, under the linearity of the model assumption in [21] (in the direct adaptive control setting, where the controllers gains are estimated), or in the indirect adaptive control setting, under the assumption of linear parametrization of the control in terms of the uncertainties in [22]. The modular design idea of using a model-based controller with ISS guarantee, complemented with an MES-based module can be found in [25, 26, 30, 31, 32], where the MES was used to estimate the model parameters and in [24, 48], where feedback gains were tuned using MES algorithms. The work of this paper falls in this class of ISS-based modular indirect adaptive controllers. The difference with other MES-based adaptive controllers is that, due to the ISS modular design we can use any data-driven learning algorithm to estimate the model uncertainties, not necessarily extremum seeking-based. To emphasize this we show here the performance of the controller when using a type of RL-based learning algorithm, namely, GP-UCB algorithms.

The rest of the paper is organized as follows. In Section 2, we present some notations, and fundamental definitions that will be needed in the sequel. In Section 3, we formulate the problem,

Prepared using acsauth.cls
DOI: 10.1002/acs

and introduce the class of systems that we are studying in this work. The nominal controller design is presented in Section 4. In Section 4.2, a robust controller is designed which guarantees ISS from the estimation error input to the tracking error state. In Section 4.3, the ISS controller is complemented with an MES algorithm to estimate the model parametric uncertainties. In Section 4.4, we introduce the RL GP-UCB algorithm as a data-driven learning to complement the ISS controller. Section 5 is dedicated to an application example, and a conclusion is given in Section 6.

2. PRELIMINARIES AND DEFINITIONS

Throughout the paper, we use $\|\cdot\|$ to denote the Euclidean norm; i.e., for a vector $x \in \mathbb{R}^n$, we have $\|x\| \triangleq \|x\|_2 = \sqrt{x^T x}$, where x^T denotes the transpose of the vector x. We denote by Card(S) the size of a finite set S. The Frobenius norm of a matrix $A \in \mathbb{R}^{m\times n}$, with elements a_{ij}, is defined as $\|A\|_F \triangleq \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}|a_{ij}|^2}$. Given $x \in \mathbb{R}^m$, the signum function is defined as $\text{sign}(x) \triangleq [\text{sign}(x_1),\ \text{sign}(x_2),\ \cdots,\ \text{sign}(x_m)]^T$, where sign(.) denotes the classical signum function. We use $\dot{f}$ to denote the time derivative of f and $f^{(r)}(t)$ for the r-th derivative of $f(t)$, i.e., $f^{(r)} \triangleq \frac{d^r f}{dt^r}$. We denote by $\mathbb{C}^k$, functions that are k times differentiable and by $\mathbb{C}^\infty$, a smooth function. A continuous function $\alpha : [0, a) \to [0, \infty)$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$. It is said to belong to class $\mathcal{K}_\infty$ if $a = \infty$ and $\alpha(r) \to \infty$ as $r \to \infty$ [49]. A continuous function $\beta : [0, a) \times [0, \infty) \to [0, \infty)$ is said to belong to class $\mathcal{KL}$ if, for a fixed s, the mapping $\beta(r, s)$ belongs to class $\mathcal{K}$ with respect to r and, for each fixed r, the mapping $\beta(r, s)$ is decreasing with respect to s and $\beta(r, s) \to 0$ as $s \to \infty$ [49].

Next, we introduce some definitions that will be used in the sequel, e.g. [49]: consider the system

$$\dot{x} \ = \ f(t, x, u), \tag{1}$$

where $f : [0, \infty) \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is piecewise continuous in t and locally Lipschitz in x and u, uniformly in t. The input $u(t)$ is piecewise continuous, bounded function of t for all $t \geq 0$.

Definition 1 ([49, 50])

The system (1) is said to be *input-to-sate stable* (ISS) if there exist a class $\mathcal{KL}$ function β and a class $\mathcal{K}$ function γ such that for any initial state $x(t_0)$ and any bounded input $u(t)$, the solution $x(t)$ exists for all $t \geq t_0$ and satisfies

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t - t_0) + \gamma(\sup_{t_0\leq\tau\leq t}\|u(\tau)\|).$$

Theorem 1 ([49, 50])

Let $V : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function such that

$$\begin{aligned} \alpha_1(\|x\|) \leq & V(t, x) \leq \alpha_2(\|x\|), \\ \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x, u) \leq & -W(x), \quad \forall \|x\| \geq \rho(\|u\|) > 0, \end{aligned} \tag{2}$$

Prepared using acsauth.cls
DOI: 10.1002/acs

for all $(t, x, u) \in [0, \infty) \times \mathbb{R}^n \times \mathbb{R}^m$, where α_1, α_2 are class $\mathcal{K}_\infty$ functions, ρ is a class $\mathcal{K}$ function, and $W(x)$ is a continuous positive definite function on $\mathbb{R}^n$. Then, the system (1) is input-to-state stable (ISS).

Remark 1. Note that other equivalent definitions for ISS have been given in [50, pp. 1974-1975]. For instance, Theorem 1 holds if inequality (2) is replaced by

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x, u) \leq -\mu(\|x\|) + \Omega(\|u\|),$$

where $\mu \in \mathcal{K}_\infty \bigcap C^1$ and $\Omega \in \mathcal{K}_\infty$.

3. PROBLEM FORMULATION

3.1. Nonlinear system model

We consider here affine uncertain nonlinear systems of the form

$$\begin{aligned} \dot{x} &= f(x) + \Delta f(t, x) + g(x)u,\ x(0) = x_0, \\ y &= h(x), \end{aligned} \tag{3}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, $y \in \mathbb{R}^m$ ($p \geq m$), represent the state, the input, and the controlled output vectors, respectively. $\Delta f(t, x)$ is a vector field representing additive model uncertainties. The vector fields f, Δf, columns of g and function h satisfy the following standard assumptions.

Assumption A1 The function $f : \mathbb{R}^n \to \mathbb{R}^n$ and the columns of $g : \mathbb{R}^n \to \mathbb{R}^p$ are $\mathbb{C}^\infty$ vector fields on a bounded set X of $\mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}^m$ is a $\mathbb{C}^\infty$ vector on X. The vector field $\Delta f(x)$ is $\mathbb{C}^1$ on X.

Assumption A2 System (3) has a well-defined (vector) relative degree $\{r_1,\ r_2,\ \cdots,\ r_m\}$ at each point $x^0 \in X$, and the system is linearizable, i.e., $\sum_{i=1}^{m} r_i = n$.

Assumption A3 The desired output trajectories y_{id} ($1 \leq i \leq m$) are smooth functions of time, relating desired initial points $y_{id}(0)$ at $t = 0$ to desired final points $y_{id}(t_f)$ at $t = t_f$.

3.2. Control objectives

Our objective is to design a learning-based state feedback adaptive iterative controller such that the output tracking error remains bounded over the learning iterations, whereas the tracking error upper-bound is a function of the uncertain parameters estimation error, which can be decreased by the data-driven learning iterations. We stress that the goal of learning algorithm is not stabilization but rather performance optimization, i.e., the learning improves the parameters estimation error, which in turn improves the output tracking error. To achieve this control objective, we proceed as

Prepared using acsauth.cls
DOI: 10.1002/acs

follows: first, we design a robust controller which can guarantee input-to-state stability (ISS) of the tracking error dynamics w.r.t. the estimation errors input. More formally, we want to design a state-feedback controller $u(t,x)$, such that the solution of the feedback dynamics satisfies the ISS condition

$$\|e_y(t)\| \leq \beta(\|e_y(t_0)\|, t-t_0) + \gamma(\sup_{t_0 \leq \tau \leq t} \|e_\Delta(\tau)\|),$$

where $e_y,\ e_\Delta$ denote the output tracking error, and the uncertainties estimation error, respectively. Then, we combine this controller with a data-driven learning algorithm to *iteratively estimate the uncertain parameters*, by optimizing online a desired learning cost function, i.e, we want to design a learning algorithm such that $e_\Delta(I)$ decreases with the number of learning iterations I, which implies by the ISS condition that e_y will decrease with I, as well.

4. ADAPTIVE CONTROLLER DESIGN

4.1. Nominal Controller

Let us first consider the system under nominal conditions, i.e., when $\Delta f(t,x) = 0$. In this case, it is well known, e.g., [49], that system (3) can be written as

$$y^{(r)}(t) \ = \ b(\xi(t)) + A(\xi(t))u(t), \tag{4}$$

where

$$\begin{aligned} y^{(r)}(t) \ &= \ [y_1^{(r_1)}(t),\ y_2^{(r_2)}(t),\ \cdots,\ y_m^{(r_m)}(t)]^T, \\ \xi(t) \ &= \ [\xi^1(t),\ \cdots,\ \xi^m(t)]^T, \\ \xi^i(t) \ &= \ [y_i(t),\ \cdots,\ y_i^{(r_i-1)}(t)]. \quad 1 \leq i \leq m \end{aligned} \tag{5}$$

The functions $b(\xi)$, $A(\xi)$ can be written as functions of f, g and h, and $A(\xi)$ is non-singular in $\tilde{X}$, where $\tilde{X}$ is the image of the set of X by the diffeomorphism $x \mapsto \xi$ between the states of system (3) and the linearized model (4). Now, to deal with the uncertain model, we first need to introduce one more assumption on system (3).

Assumption A4 The additive uncertainties $\Delta f(t,x)$ in (3) appear as additive uncertainties in the input-output linearized model (4)-(5) as follows (see also [51])

$$y^{(r)}(t) \ = \ b(\xi(t)) + A(\xi(t))u(t) + \Delta b(t,\xi(t)), \tag{6}$$

where $\Delta b(t,\xi)$ is $\mathbb{C}^1$ w.r.t. the state vector $\xi \in \tilde{X}$.

Remark 2. Assumption A4 can be ensured under the matching conditions, e.g., [52].

It is well known that the nominal model (4) can be easily transformed into a linear input-output mapping. Indeed, we can first define a virtual input vector $v(t)$ as

$$v(t) \ = \ b(\xi(t)) + A(\xi(t))u(t). \tag{7}$$

Prepared using acsauth.cls
DOI: 10.1002/acs

Combining (4) and (7), we can obtain the following input-output mapping

$$y^{(r)}(t) \;=\; v(t). \tag{8}$$

Based on the linear system (8), it is straightforward to design a stabilizing controller for the nominal system (4) as*

$$u_n = A^{-1}(\xi)\left[v_s(t,\xi) - b(\xi)\right], \tag{9}$$

where v_s is a $m \times 1$ vector and the i-th ($1 \leq i \leq m$) element v_{si} is given by

$$v_{si} = y_{id}^{(r_i)} - K_{r_i}^i(y_i^{(r_i-1)} - y_{id}^{(r_i-1)}) - \cdots - K_1^i(y_i - y_{id}). \tag{10}$$

If we denote the tracking error as $e_i(t) \;\triangleq\; y_i(t) - y_{id}(t)$, we obtain the following tracking error dynamics

$$e_i^{(r_i)}(t) + K_{r_i}^i e^{(r_i-1)}(t) + \cdots + K_1^i e_i(t) = 0, \tag{11}$$

where $i \in \{1,\ 2,\ \cdots,\ m\}$. By properly selecting the gains K_j^i where $i \in \{1,\ 2,\ \cdots,\ m\}$ and $j \in \{1,\ 2,\ \cdots,\ r_i\}$, we can obtain global asymptotic stability of the tracking errors $e_i(t)$. To formalize this condition, we add the following assumption.

Assumption A5 There exists a non-empty set $\mathcal{A}$ where $K_j^i \in \mathcal{A}$ such that the polynomials in (11) are Hurwitz, where $i \in \{1,\ 2,\ \cdots,\ m\}$ and $j \in \{1,\ 2,\ \cdots,\ r_i\}$.

To this end, we define $z = [z^1,\ z^2,\ \cdots,\ z^m]^T$, where $z^i = [e_i,\ \dot{e}_i,\ \cdots,\ e_i^{(r_i-1)}]$ and $i \in \{1,\ 2, \cdots,\ m\}$. Then, from (11), we can obtain

$$\dot{z} \;=\; \tilde{A}z,$$

where $\tilde{A} \in \mathbb{R}^{n\times n}$ is a diagonal block matrix given by

$$\tilde{A} \;=\; \mathrm{diag}\{\tilde{A}_1,\ \tilde{A}_2,\ \cdots,\ \tilde{A}_m\}, \tag{12}$$

and $\tilde{A}_i$ ($1 \leq i \leq m$) is a $r_i \times r_i$ matrix given by

$$\tilde{A}_i \;=\; \begin{bmatrix} 0 & 1 & & & \\ 0 & & 1 & & \\ 0 & & & \ddots & \\ \vdots & & & & 1 \\ -K_1^i & -K_2^i & \cdots & \cdots & -K_{r_i}^i \end{bmatrix}.$$

*The inverse of A is to be understood in the sense of Moore-Penrose pseudo-inverse which is guaranteed to exist by the relative degree Assumption A2.

Prepared using acsauth.cls
DOI: 10.1002/acs

As discussed above, the gains K_j^i can be chosen such that the matrix $\tilde{A}$ is Hurwitz. Thus, there exists a positive definite matrix $P > 0$ such that (see e.g. [49])

$$\tilde{A}^T P + P\tilde{A} = -I. \tag{13}$$

In the next section, we build upon the nominal controller (9) to write a robust ISS controller.

4.2. Lyapunov reconstruction-based ISS Controller

We now consider the uncertain model (3), i.e., when $\Delta f(t,x) \neq 0$. The corresponding exact linearized model is given by (6) where $\Delta b(t,\xi(t)) \neq 0$. The global asymptotic stability of the error dynamics (11) cannot be guaranteed anymore due to the additive uncertainty $\Delta b(t,\xi(t))$. We use Lyapunov reconstruction techniques to design a new controller so that the tracking error is guaranteed to be bounded given that the estimate error of $\Delta b(t,\xi(t))$ is bounded. The new controller for the uncertain model (6) is defined as

$$u_f = u_n + u_r, \tag{14}$$

where the nominal controller u_n is given by (9) and the robust controller u_r will be given later. By using controller (14), and (6) we obtain

$$\begin{aligned} y^{(r)}(t) &= b(\xi(t)) + A(\xi(t))u_f + \Delta b(t,\xi(t)), \\ &= b(\xi(t)) + A(\xi(t))u_n + A(\xi(t))u_r + \Delta b(t,\xi(t)), \\ &= v_s(t,\xi) + A(\xi(t))u_r + \Delta b(t,\xi(t)), \end{aligned} \tag{15}$$

where (15) holds from (9). This leads to the following error dynamics

$$\dot{z} = \tilde{A}z + \tilde{B}\delta, \tag{16}$$

where $\tilde{A}$ is defined in (12), δ is a $m \times 1$ vector given by

$$\delta = A(\xi(t))u_r + \Delta b(t,\xi(t)), \tag{17}$$

and the matrix $\tilde{B} \in \mathbb{R}^{n\times m}$ is given by

$$\tilde{B} = \left[\tilde{B}_1^T, \tilde{B}_2^T, \ldots, \tilde{B}_m^T \right]^T, \tag{18}$$

where each $\tilde{B}_i$ $(1 \leq i \leq m)$ is given by a $r_i \times m$ matrix such that

$$\tilde{B}_i(l,q) = \begin{cases} 1 & \text{for } l = r_i,\, q = i, \\ 0 & \text{otherwise.} \end{cases}$$

Prepared using acsauth.cls
DOI: 10.1002/acs

If we choose $V(z) = z^T P z$ as a Lyapunov function for the dynamics (16), where P is the solution of the Lyapunov equation (13), we obtain

$$\begin{aligned} \dot{V}(t) &= \frac{\partial V}{\partial z}\dot{z}, \\ &= z^T(\tilde{A}^T P + P\tilde{A})z + 2z^T P\tilde{B}\delta, \\ &= -\|z\|^2 + 2z^T P\tilde{B}\delta, \end{aligned} \tag{19}$$

where δ given by (17) depends on the robust controller u_r.

Next, we design the controller u_r based on the form of the uncertainties $\Delta b(t, \xi(t))$. More specifically, we consider the case when $\Delta b(t, \xi(t))$ is of the following form

$$\Delta b(t, \xi(t)) = E\, Q(\xi, t), \tag{20}$$

where $E \in \mathbb{R}^{m \times m}$ is a matrix of unknown constant parameters, and $Q(\xi, t) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ is a known bounded function of states and time variables. For notational convenience, we denote by $\hat{E}(t)$ the estimate of E, and by $e_E = E - \hat{E}$, the estimate error. We define the unknown parameter vector $\Delta = [E(1,1), ..., E(m,m)]^T \in \mathbb{R}^{m^2}$, i.e., concatenation of all elements of E, its estimate is denoted by $\widehat{\Delta}(t) = [\hat{E}(1,1), ..., \hat{E}(m,m)]^T$, and the estimation error vector is given by $e_\Delta(t) = \Delta - \widehat{\Delta}(t)$.

Next, we propose the following robust controller

$$u_r = -A^{-1}(\xi)[\tilde{B}^T P z \|Q(\xi, t)\|^2 + \widehat{E}(t)Q(\xi, t)]. \tag{21}$$

The closed-loop error dynamics can be written as

$$\dot{z} = \tilde{f}(t, z, e_\Delta), \tag{22}$$

where $e_\Delta(t)$ is considered to be an input to the system (22).

Theorem 2

Consider the system (3), under Assumptions A1-A5, where $\Delta b(t, \xi(t))$ satisfies (20). If we apply to (3) the feedback controller (14), where u_n is given by (9) and u_r is given by (21), then the closed-loop system (22) is ISS from the estimation errors input $e_\Delta(t) \in \mathbb{R}^{m^2}$ to the tracking errors state $z(t) \in \mathbb{R}^n$.

Proof: By substitution (21) into (17), we obtain

$$\begin{aligned} \delta &= -\tilde{B}^T P z \|Q(\xi, t)\|^2 - \widehat{E}(t)\, Q(\xi, t) + \Delta b(t, \xi(t)), \\ &= -\tilde{B}^T P z \|Q(\xi, t)\|^2 - \widehat{E}(t)\, Q(\xi, t) + E\, Q(\xi, t), \end{aligned}$$

If we consider $V(z) = z^T P z$ as a Lyapunov function for the error dynamics (16). Then, from (19), we obtain

$$\dot{V} \leq -\|z\|^2 + 2z^T P\tilde{B}E\, Q(\xi, t) - 2z^T P\tilde{B}\widehat{E}(t)\, Q(\xi, t) - 2\|z^T P\tilde{B}\|^2\|Q(\xi, t)\|^2,$$

Prepared using acsauth.cls
DOI: 10.1002/acs

which leads to

$$\dot{V} \leq \ -\|z\|^2 + 2z^T P\tilde{B}e_E Q(\xi,t) - 2\|z^T P\tilde{B}\|^2\|Q(\xi,t)\|^2.$$

Since $z^T P\tilde{B}e_E Q(\xi) \leq \|z^T P\tilde{B}e_E Q(\xi)\| \leq \|z^T P\tilde{B}\|\|e_E\|_F\|Q(\xi)\| = \|z^T P\tilde{B}\|\|e_\Delta\|\|Q(\xi)\|$, we obtain

$$\begin{aligned}\dot{V} \leq \ & -\|z\|^2 + 2\|z^T P\tilde{B}\|\|e_\Delta\|\|Q(\xi,t)\| - 2\|z^T P\tilde{B}\|^2\|Q(\xi,t)\|^2,\\ \leq \ & -\|z\|^2 - 2(\|z^T P\tilde{B}\|\|Q(\xi,t)\| - \frac{1}{2}\|e_\Delta\|)^2 + \frac{1}{2}\|e_\Delta\|^2,\\ \leq \ & -\|z\|^2 + \frac{1}{2}\|e_\Delta\|^2.\end{aligned}$$

Thus, we have the following relation

$$\dot{V} \leq -\frac{1}{2}\|z\|^2, \quad \forall\|z\| \geq \|e_\Delta\| > 0,$$

Then from the Lyapunov direct Theorem 1, we obtain that system (22) is ISS from input e_Δ to state z. □

Remark 3. In the case of constant uncertainty vector, i.e., $\Delta b = \Delta = cte \in \mathbb{R}^m$, the controller (21) boils down to the simple feedback

$$u_r = -A^{-1}(\xi)[\tilde{B}^T Pz + \widehat{\Delta}(t)]. \tag{23}$$

Remark 4. We have not explicitly mentioned in Theorem 2 the case of measurement noise. However, in the case of bounded additive measurement noise $d(t) \in \mathbb{R}^p$ which appears as an additive disturbance to Δb, i.e., the new uncertainty term writes as $\Delta b + A.d(t)$, we can easily show, following the same steps as in the proof of Theorem 2, that the ISS result holds from the extended input $(e_\Delta^T, \tilde{d}^T)^T,\ \tilde{d} = A.d$, to the tracking error z. This means that the controller (9), (14), and (21) is robust w.r.t. this type of measurement noise, and the results of Theorem 2 remains valid in this case.

4.3. Iterative MES-based parametric uncertainties estimation

Let us define now the following cost function

$$J(\widehat{\Delta}) \ = \ F(z(\widehat{\Delta})), \tag{24}$$

where $F : \mathbb{R}^n \to \mathbb{R}$, $F(\mathbf{0}) = 0$, $F(z) > 0$ for $z \in \mathbb{R}^n - \{\mathbf{0}\}$. We need the following assumptions on J.

Assumption A6 The cost function J has a local minimum at $\widehat{\Delta}^* \ = \ \Delta$, i.e., $J(\widehat{\Delta}) > J(\Delta),\ \forall\widehat{\Delta} \in \mathcal{V}(\Delta)$, where $\mathcal{V}(\Delta)$ denotes a compact neighborhood of Δ.

Prepared using acsauth.cls
DOI: 10.1002/acs

Algorithm 1 MES-based Learning Adaptive Controller

– Initialize: $I = 1,\ x(0) = x_0,\ J_{th} > 0,\ \hat{\Delta} = \Delta_{nominal},\ K_1^i, ..., K_{r_i}^i, i = 1, ..., m.$

– Solve (13).

– Apply the controller (9), (14), and (21), to (3), (20).

(Loop) – Evaluate the learning cost J by (24).

– IF $J \leq J_{th} \rightarrow$ Exit Loop, IF not:

– I=I+1.

– Estimate $\hat{\Delta}$ by (25).

– Reset $t \in [(I-1)t_f,\ It_f]$, $x((I-1)t_f) = x_0$, then, apply the controller (9), (14), and (21), to (3), (20).

– Go to (Loop).

Assumption A7 The initial error $e_\Delta(t_0)$ is sufficiently small, i.e., the original parameter estimate vector $\widehat{\Delta}$ is in the compact neighborhood $\mathcal{V}(\Delta)$ as defined in Assumption A6.

Assumption A8 The cost function J is analytic and its variation with respect to the uncertain parameters is bounded in the neighborhood of $\widehat{\Delta}^*$, i.e., $\|\frac{\partial J}{\partial \tilde{\Delta}}(\tilde{\Delta})\| \leq \xi_2$, $\xi_2 > 0$, $\tilde{\Delta} \in \mathcal{V}(\widehat{\Delta}^*)$, where $\mathcal{V}(\widehat{\Delta}^*)$ denotes a compact neighborhood of $\widehat{\Delta}^*$.

Remark 5. Assumption A6 simply states that the cost function J has at least a local minimum at the true values of the uncertain parameters.

Remark 6. Assumption A7 indicates that our results are of local nature, i.e., our analysis holds in a small neighborhood of the actual values of the uncertain parameters. This makes the results of the MES-based controller valid only for small uncertainties. This can be a limitation in some practical applications. We will address this problem in the Section 4.4, where we introduce another learning algorithm with more global convergence results.

We can now present the stability analysis of the MES-based controller (Algorithm 1).

Lemma 3

Consider the system (3), under Assumptions A1-A8, where the uncertainty is given by (20). If we apply to (3) the feedback controller (14), (9), and (21), where the state vector is reset following the resetting law $x(It_f) = x_0,\ I \in \{1, 2, ...\}$, the desired trajectory vector is reset following $\hat{y}_{id}(t) = y_{id}(t - (I-1)t_f),\ (I-1)t_f \leq t \leq It_f,\ I \in \{1, 2, ...\}$, the cost function is given by (24), and the elements of the vector $\widehat{\Delta}(t)$ are estimated through the iterative MES algorithm

$$\begin{aligned}
&\dot{\tilde{x}}_i = a_i \sin(\omega_i t + \tfrac{\pi}{2}) J(\widehat{\Delta}),\ a_i > 0,\\
&\hat{\delta}\Delta_i(t) = \tilde{x}_i + a_i \sin(\omega_i t - \tfrac{\pi}{2}),\\
&\widehat{\Delta}_i(t) = \widehat{\Delta}_{i-nominal} + \delta\Delta_i(t),\\
&\delta\Delta_i(t) = \hat{\delta}\Delta_i((I-1)t_f),\ (I-1)t_f \leq t \leq It_f,\ I = 1, 2, 3, ..., i \in \{1, 2, \ldots, m^2\},
\end{aligned} \tag{25}$$

Prepared using acsauth.cls
DOI: 10.1002/acs

with $\omega_i \neq \omega_j$, $\omega_i + \omega_j \neq \omega_k$, $i, j, k \in \{1, 2, \ldots, m^2\}$, and $\omega_i > \omega^*$, $\forall\ i \in \{1, 2, \ldots, m^2\}$, with ω^* large enough. Then, the obtained closed-loop impulsive time-dependent dynamic system is well posed. The norm of the error vector $z(t)$ satisfies an ISS condition, with respect to the input $e_\Delta(t) = \Delta - \hat{\Delta}(t)$. Furthermore, $e_{\Delta(t)}$ remains bounded over the learning iterations, and satisfies $\sup_{(I-1)t_f \leq \tau \leq It_f} \|e_\Delta(\tau)\| = \frac{\xi_1}{\omega_0} + \sqrt{\sum_{i=1}^{m^2} a_i^2},\ \xi_1 > 0,\ I \to \infty,$ with $\omega_0 = \max_{i \in \{1,2,\ldots,m^2\}} \omega_i$, $\beta \in \mathcal{KL}$, $\tilde{\beta} \in \mathcal{KL}$ and $\gamma \in \mathcal{K}$.

Proof: First, the closed-loop system (3), (20), (14), (9), (21), (24), and (25) can be viewed as an impulsive time-dependent dynamical system ([53], pp. 18-19), with the trivial resetting law $\Delta x(t) = x_0,$ for $t = It_f,\ I \in \{1, 2, ...\}$. In this case the resetting times given by $It_f,\ t_f > 0\ I \in \{1, 2, ...\}$, are well defined and distinct. Furthermore, due to Assumption 1 and the smoothness of the control (9), (14), and (21) (within each iteration), this impulsive dynamic system admits a unique solution in forward time, for any initial condition $x_0 \in \mathbb{R}^n$ ([53], p. 12). Finally, the fact that $t_f \neq 0$ excludes a Zeno behavior over a finite time interval (only a finite number of resets are possible over a finite time interval).

Next, based on Theorem 2, we know that the tracking error dynamics (22) is ISS from the input $e_\Delta(t)$ to the state $z(t)$. Thus, by Definition 1 in Section 2, there exist a class $\mathcal{KL}$ function β and a class $\mathcal{K}$ function γ such that for each iteration I, and the associated time interval $(I-1)t_f \leq t < It_f$, for any initial state $z((I-1)t_f)$, any bounded input $e_\Delta(t)$, we can write that

$$\|z(t)\| \leq \beta(\|z((I-1)t_f)\|, t) + \gamma(\sup_{(I-1)t_f \leq \tau \leq It_f} \|e_\Delta(\tau)\|). \tag{26}$$

Now, we need to evaluate the bound on the estimation vector $\hat{\delta}\Delta = (\hat{\delta}\Delta_1, ..., \hat{\delta}\Delta_{m^2})^T$, i.e., bound on $\widehat{\Delta} = (\widehat{\Delta}_1, ..., \widehat{\Delta}_{m^2})^T$, to do so we use some of the results presented in [45]. First, based on Assumptions A6, A7 and A8, the MES nonlinear dynamics (25) can be approximated by a linear averaged dynamic (using averaging approximation over time ([45], p 435, Definition 1)). Furthermore, $\exists \xi_1,\ \omega^*$, such that for all $\omega_0 = \max(\omega_1, ..., \omega_{m^2}) > \omega^*$, the solution of the averaged model $\hat{\delta}\Delta_{aver}(t)$ is locally close to the solution of the original ES dynamics, and satisfies ([45], p. 436)

$$\|\hat{\delta}\Delta(t) - d_{vec}(t) - \hat{\delta}\Delta_{aver}(t)\| \leq \tfrac{\xi_1}{\omega_0},\ \xi_1 > 0,\ \forall t \geq 0,$$

with $d_{vec}(t) = (a_1 \sin(\omega_1 t - \frac{\pi}{2}), ..., a_{rm} \sin(\omega_{rm} t - \frac{\pi}{2}))^T$. Moreover, since J is analytic it can be approximated locally in $\mathcal{V}(\hat{\delta}\Delta^*)$ with a quadratic function, e.g., Taylor series up to second order. This together with the proper choice of the dither signals as in (25), and the dither frequencies satisfying $\omega_i \neq \omega_j,\ \omega_i + \omega_j \neq \omega_k,\ i, j, k \in \{1, 2, \ldots, m^2\},$ with $\omega_i > \omega^*,\ \forall\ i \in \{1, 2, \ldots, m^2\}$, allows us to prove that $\hat{\delta}\Delta_{aver}$ satisfies ([45], p. 437)

$$\lim_{t \to \infty} \hat{\delta}\Delta_{aver}(t) = \hat{\delta}\Delta^*,$$

DOI: 10.1002/acs

which together with the previous inequality leads to

$$\|\hat{\delta}\Delta(t) - \hat{\delta}\Delta^*\| - \|d(t)\| \leq \|\hat{\delta}\Delta(t) - \hat{\delta}\Delta^* - d(t)\| \leq \frac{\xi_1}{\omega_0},\ \xi_1 > 0,\ t \to \infty,$$
$$\Rightarrow \|\hat{\delta}\Delta(t) - \hat{\delta}\Delta^*\| \leq \frac{\xi_1}{\omega_0} + \|d(t)\|,\ t \to \infty.$$

This finally implies that

$$\|\hat{\delta}\Delta(t) - \hat{\delta}\Delta^*\| \leq \frac{\xi_1}{\omega_0} + \sqrt{\sum_{i=1,...,m^2} {a_i}^2},\ \xi_1 > 0,\ t \to \infty,$$
$$\Rightarrow \|\hat{\delta}\Delta(It_f) - \hat{\delta}\Delta^*\| \leq \frac{\xi_1}{\omega_0} + \sqrt{\sum_{i=1,...,m^2} {a_i}^2} \equiv \sup_{(I-1)t_f \leq \tau \leq It_f} \|e_\Delta(\tau)\|,\ \xi_1 > 0,\ I \to \infty.$$

Next, based on Assumption A8, the cost function is locally Lipschitz, with the Lipschitz constant $\max_{\hat{\delta}\Delta \in \mathcal{V}(\hat{\delta}\Delta^*)} \|\frac{\partial J}{\partial \hat{\delta}\Delta}\| = \xi_2$, i.e., $\|J(\hat{\delta}\Delta_1) - J(\hat{\delta}\Delta_2)\| \leq \xi_2 \|\hat{\delta}\Delta_1 - \hat{\delta}\Delta_2\|,\ \forall \hat{\delta}\Delta_1, \hat{\delta}\Delta_2 \in \mathcal{V}(\hat{\delta}\Delta^*)$, which together with the previous inequality leads to

$$\|J(\hat{\delta}\Delta(It_f)) - J(\hat{\delta}\Delta^*)\| \leq \xi_2 (\frac{\xi_1}{\omega_0} + \sqrt{\sum_{i=1,...,m^2} a_j^2}),\ \xi_1,\ \xi_2 > 0,\ I \to \infty.$$

Finally, we show that the MES algorithm (25) is a gradient-based algorithm, as follows: from the first two equations in (25), if we denote $X = (\tilde{x}_1, ..., \tilde{x}_{m^2})^T$, we can write

$$\dot{X} = (a_1\omega_1 sin(\omega_1 t - \frac{\pi}{2}), ..., a_1\omega_{m^2} sin(\omega_{m^2} t - \frac{\pi}{2}))^T J(\hat{\delta}\Delta). \tag{27}$$

Based on Assumption A8, the cost function can be locally approximated with its first order Taylor development in $\mathcal{V}(\hat{\delta}\Delta^*)$, which leads to

$$\dot{X} \simeq \tilde{d}_{vec}\big(J(\tilde{\hat{\delta}}\Delta) + \overline{d}_{vec}^T \frac{\partial J}{\partial \hat{\delta}\Delta}(\tilde{\hat{\delta}}\Delta)\big),\ \tilde{\hat{\delta}}\Delta \in \mathcal{V}(\hat{\delta}\Delta^*), \tag{28}$$

where $\tilde{d}_{vec} = (a_1\omega_1 sin(\omega_1 t - \frac{\pi}{2}), ..., a_{m^2}\omega_{m^2} sin(\omega_{m^2} t - \frac{\pi}{2}))^T$, and $\overline{d}_{vec} = (a_1 sin(\omega_1 t - \frac{\pi}{2}), ..., a_{m^2} sin(\omega_{m^2} t - \frac{\pi}{2}))^T$.

Next, by integrating (28), over $[t,\ t + t_f]$ and neglecting the terms inversely proportional to the high frequencies, i.e., terms on $\frac{1}{\omega_i}$'s (high frequencies filtered by the integral operator), we obtain

$$X(t + t_f) - X(t) \simeq -t_f R \frac{\partial J}{\partial \hat{\delta}\Delta}(\tilde{\hat{\delta}}\Delta), \tag{29}$$

with $R = 0.5\ \text{diag}\{\omega_1 a_1^2, ..., \omega_{m^2} a_{m^2}^2\}$.

Next, from (25), we can write $\|\widehat{\Delta}(t + t_f) - \widehat{\Delta}(t)\| \leq \|X(t + t_f) - X(t)\| + \|\overline{d}_{vec}(t + t_f) - \overline{d}_{vec}(t)\|$, which together with (29), with the bound $\|\frac{\overline{d}_{vec}(t+t_f) - \overline{d}_{vec}(t)}{t_f}\| \leq \|\dot{\overline{d}}_{vec}\| \leq \omega_0 \sqrt{\sum_{i=1,...,m^2} a_i^2}$, and Assumption A8, leads to the inequality

$$\|\widehat{\Delta}((I+1)t_f) - \widehat{\Delta}(It_f)\| \leq 0.5 t_f \max(\omega_1 {a_1}^2, ..., \omega_{m^2} {a_{m^2}}^2)\xi_2 + t_f \omega_0 \sqrt{\sum_{i=1,...,m^2} {a_i}^2},\ I \in \{1, 2, ...\},$$

which shows the boundedness of e_Δ over the learning iterations. □

Prepared using acsauth.cls
DOI: 10.1002/acs

Remark 7. The batch Algorithm 1 resembles classical ILC in its resetting of initial conditions. This type of batch algorithms are suitable in applications where the initial conditions can be controlled, for example, in rigid manipulators in factory automation with repetitive tasks, or in electro-magnetic brakes where the moving part is periodically reset to the same initial conditions, etc., see [64] for more applications. However, similar to the recent results in the ILC literature where the resetting condition has been relaxed, e.g., [65] and references therein, in our method we can also relax this condition if we consider the case where the effect of the initial conditions on the cost function dissipates rapidly in the transient. In other words, if the transient time of the dynamical system is shorter than the iteration length of the learning algorithm, such that in each iteration we reach the same value of the cost function for the same value of the estimated parameters, regardless of the states' initial conditions. This is indeed, a classical assumption in extremum seekers algorithms, e.g., Assumption 1, 2 in [54].

Remark 8. The results of Lemma 3 can be summarized as follows: The controller (14), (9), (21), and (25) guarantees that the tracking error satisfies and ISS inequality as defined in Definition 1, where the input is the estimation error e_Δ. This means that the tracking error is bounded as long as the estimation error is bounded, and that the tracking error decreases with the decrease of the estimation error. Furthermore, the estimation error is bounded, decreases with the learning iterations $I = 1, 2, ...$, and converges asymptotically to an error bounded by $\frac{\xi_1}{\omega_0} + \sqrt{\sum_{i=1}^{m^2} a_i^2}$, $\xi_1 > 0$. This estimation error upper-bound can then be made as small as needed by the proper choice of the MES dither signals' amplitudes a_is.

Remark 9. The iterative adaptive controller of Lemma 3 uses the MES algorithm (25) to estimate the model parametric uncertainties. One might ask the question: where is the famous persistence of excitation (PE) condition on the MES filter ? The answer can be found in the examination of equation (25). Indeed, the MES algorithm uses as 'input' the sinusoidal signals $a_i \sin(\omega_i t + \frac{\pi}{2})$ which clearly satisfy the PE condition. The main difference with classical adaptive control result is that these excitation signals are not entering the system dynamics directly, but instead are applied as inputs to the MES algorithm, reflected on the MES estimations outputs and thus transmitted to the system through the feedback loop.

Remark 10. As explained earlier, when introducing the objectives of the controller, the data-driven learning part of the controller is not aiming at stabilizing the feedback-loop, instead, its goal is to estimate online the parametric uncertainties of the model. During the estimation phase, the stability, in the sense of boundedness, is ensured by the model-based part of the controller, which imposes ISS on the closed-loop. However, we notice that the upper-bound of the tracking error norm is function of the upper-bound of the estimation error norm, which by decreasing, over the learning iterations, implies an improvement of the tracking performance over the learning iterations.

Prepared using acsauth.cls
DOI: 10.1002/acs

Remark 11. The upper-bound of the estimation error norm obtained in Lemma 3 is correlated to the choice of the simple dither-based MES algorithm. However, due to the modular design, this bound can be improved, for instance by using other types of MES algorithms, e.g., [55], where different type of dither signals have been explored, or [56], where the authors proposed an MES algorithm without steady-state residual oscillations. Furthermore, we want to emphasize here that the convergence results of the estimation error in Lemma 3 are of local nature, but they can easily be extended to semi-global convergence results by substituting the MES algorithm with an MES which guarantees semi-global convergence, e.g., [54].

As we mentioned earlier, the dither-based MES suffers from the problem of local minima, which forced us to constrain our analysis, of the iterative adaptive controller, to a neighborhood of the true parameters. To address this point in the next section we propose to use GP-UCB as the data-driven learning algorithm for global model uncertainties estimation in any compact search set D.

4.4. Iterative GP-UCB based parametric uncertainties estimation

In this section we propose to use Gaussian Process Upper Confidence Bound (GP-UCB) algorithm to estimate the uncertain parameter vector Δ. GP-UCB is a Bayesian optimization algorithm for stochastic optimization, i.e., the task of finding the global optimum of an unknown function when the evaluations are potentially contaminated with noise, e.g., [46, 57]. The underlying working assumption for Bayesian optimization algorithms, including GP-UCB, is that the function evaluation is costly, so we would like to minimize the number of evaluations while having as accurate estimate of the minimizer (or maximizer) as possible [58]. For GP-UCB, this goal is guaranteed by having an upper bound on the regret of the algorithm – to be defined precisely later.

One difficulty of stochastic optimization is that since we only observe noisy samples from the function, we cannot really be sure about the exact value of the function at any given point. One may try to query a single point many times in order to have an accurate estimate of the function. This, however, may lead to excessive number of samples, and can be wasteful way of assigning samples when the true value of the function at that point is actually far from optimal. The Upper Confidence Bound (UCB) family of algorithms provides a principled approach to guide the search [59]. These algorithms, which are not necessarily formulated in a Bayesian framework, automatically balance the exploration (i.e., finding the regions of the parameter space that *might* be promising) and the exploitation (i.e., focusing on the regions that are known to be the best based on the *current* available knowledge) using the principle of optimism in the face of uncertainty. These algorithms often come with strong theoretical guarantee about their performance. For more information about the UCB class of algorithms, refer to [60, 61, 62]. GP-UCB is a particular UCB algorithms that is suitable to deal with continuous domains. It uses a Gaussian Process (GP) to maintain the mean and confidence information about the unknown function.

Prepared using acsauth.cls
DOI: 10.1002/acs

We briefly discuss GP-UCB in our context following the discussion of the original papers [46, 57].

Consider the learning cost function $J : D \to \mathbb{R}$ to be minimized, which can be defined by (24). This function depends on the states of the closed-loop system, which depend on the parameters $\widehat{\Delta}$ used in the controller design. Thus, we can consider it as an unknown function of $\widehat{\Delta}$, where $D \subset \mathbb{R}^{m^2}$.

In this context of GP-UCB, to be able to capture the fact that J has a global minimum in the compact search set D, we need to extend Assumption A6, as follows:

Assumption A9 The cost function J has a global minimum at $\widehat{\Delta}^* = \Delta$, in the compact search set D, i.e., $J(\widehat{\Delta}) > J(\Delta),\ \forall \widehat{\Delta} \in D$.

Let us assume that $\tilde{J}$ is a function sampled from a Gaussian Process (GP). Recall that a GP is a stochastic process indexed by the set D that has the property that for any finite subset of the evaluation points, that is $\{\widehat{\Delta}_1, \widehat{\Delta}_2, \ldots, \widehat{\Delta}_I\} \subset D$, the joint distribution of $\left(\tilde{J}(\widehat{\Delta}_i)\right)_{i=1}^{I}$ is a multivariate Gaussian distribution.

We recall that GP is defined by a mean function

$$\mu(\widehat{\Delta}) = \mathbb{E}\left[\tilde{J}(\widehat{\Delta})\right], \tag{30}$$

and its covariance function (or kernel)

$$\kappa(\widehat{\Delta}, \widehat{\Delta}') = \mathrm{Cov}(\tilde{J}(\widehat{\Delta}), \tilde{J}(\widehat{\Delta}')) = \mathbb{E}\left[\left(\tilde{J}(\widehat{\Delta}) - \mu(\widehat{\Delta})\right)\left(\tilde{J}(\widehat{\Delta}') - \mu(\widehat{\Delta}')\right)^{\top}\right]. \tag{31}$$

The kernel κ of a GP determines the behavior of a typical function sampled from the GP. For instance, if we choose

$$\kappa(\widehat{\Delta}, \widehat{\Delta}') = \exp\left(-\frac{\|\widehat{\Delta} - \widehat{\Delta}'\|^2}{2l^2}\right), \tag{32}$$

the squared exponential kernel with length scale $l > 0$, it implies that the GP is mean square differentiable of all orders.

Let us first briefly describe how we can find the posterior distribution of a GP$(0, \kappa)$, i.e., a GP with zero prior mean. Suppose that for $\underline{\widehat{\Delta}}_{I-1} \triangleq \{\widehat{\Delta}_1, \widehat{\Delta}_2, \ldots, \widehat{\Delta}_{I-1}\} \subset D$, we have observed the noisy evaluation $y_i = \tilde{J}(\widehat{\Delta}_i) = J(\widehat{\Delta}_i) + \eta_i$ with $\eta_i \sim N(0, \sigma^2)$ being i.i.d. Gaussian noise. We can find the posterior mean and variance for a new point $\widehat{\Delta}^* \in D$ as follows: Denote the vector of observed values by $\mathbf{y}_{I-1} = [y_1, \ldots, y_{I-1}]^{\top} \in \mathbb{R}^{I-1}$, and define the Grammian matrix $K \in \mathbb{R}^{I-1 \times I-1}$ with $[K]_{i,j} = \kappa(\widehat{\Delta}_i, \widehat{\Delta}_j)$, and the vector $\kappa_* = [\kappa(\underline{\widehat{\Delta}}_1, \widehat{\Delta}^*), \ldots, \kappa(\underline{\widehat{\Delta}}_{I-1}, \widehat{\Delta}^*)]$. The expected mean $\mu_I(\widehat{\Delta}^*)$ and the variance $\sigma_I(\widehat{\Delta}^*)$ of the posterior of the GP evaluated at $\widehat{\Delta}^*$ are (cf. Section 2.2 of [63])

$$\mu_I(\widehat{\Delta}^*) = \kappa_* \left[K + \sigma^2 \mathbf{I}\right]^{-1} \mathbf{y}_{I-1}, \tag{33}$$

$$\sigma_I^2(\widehat{\Delta}^*) = \kappa(\widehat{\Delta}^*, \widehat{\Delta}^*) - \kappa_*^T \left[K + \sigma^2 \mathbf{I}\right]^{-1} \kappa_*. \tag{34}$$

Prepared using acsauth.cls
DOI: 10.1002/acs

Algorithm 2 GP-UCB-based Learning Adaptive Controller

– Initialize: $I = 1,\ x(0) = x_0,\ J_{th} > 0,\ \hat{\Delta} = \Delta_{nominal}$.

– Apply the controller (9), (14), and (21), to (3), (20).

(Loop) – Evaluate the learning cost J by (24).

– IF $J \leq J_{th} \rightarrow$ Exit Loop, IF not:

– I=I+1.

– Estimate $\hat{\Delta}$ by (32), (33), (34), (35), and (36).

– Reset $t \in [(I-1)t_f,\ t_f]$, $x((I-1)t_f) = x_0$, then, apply the controller (9), (14), and (21), to (3), (20).

– Go to (Loop).

At iteration I, the GP-UCB algorithm selects the next query point $\widehat{\Delta}_I$ by solving the following optimization problem:

$$\widehat{\Delta}_I \leftarrow \underset{\widehat{\Delta} \in D}{\operatorname{argmin}}\, \mu_{I-1}(\widehat{\Delta}) - \beta_I^{1/2} \sigma_{I-1}(\widehat{\Delta}). \tag{35}$$

We select β_I as[†]

$$\beta_I = 2\,\|J\|_{\mathcal{H}_\kappa} + 300\gamma_I \log^3(I/\delta), \tag{36}$$

where $\delta \in (0,1)$, represents the confidence parameter, and $\gamma_I = \log(I)^c,\ c > 0$.

Remark 12. The optimization problem (35) is often nonlinear and non-convex. Nonetheless solving it only requires querying the GP, which in general is much faster than querying the original dynamical system. This is important when the dynamical system is a real system and we would like to minimize the number of interactions with it before finding a $\hat{\Delta}$ with small $J(\hat{\Delta})$. One practical way to approximately solve (35) is to restrict the search to a finite subset D' of D. The finite subset can be a uniform grid structure over D, or it might consist of randomly selected members of D.

Next, the theoretical guarantees for GP-UCB given below are in the form of regret upper bound. To recall the definition of cumulative regret, let us define $\hat{\Delta}^* \leftarrow \operatorname{argmin}_{\hat{\Delta} \in D} J(\hat{\Delta})$, the global minimizer of the objective function. The regret at time $t = jt_f$ is defined by $r_j = J(\widehat{\Delta}_j) - J(\hat{\Delta}^*)$. This is a measure of sub-optimality of the choice of $\widehat{\Delta}_j$ according to the cost function J. The cumulative regret at time $T = Nt_f$ is defined as $R_N = \sum_{i=1}^{i=N} r_i$.

We now summarize the convergence properties of the GP-based iterative adaptive controller (Algorithm 2), in the following Lemma.

[†] $\|.\|_{\mathcal{H}_\kappa}$ denotes the norm associated with the reproducing kernel Hilbert space (RKHS), e.g. [63].

Prepared using acsauth.cls
DOI: 10.1002/acs

Lemma 4

Consider the system (3), under Assumptions A1-A5, and A9, where the uncertainty is given by (20). If we apply to (3) the feedback controller (14), (9), and (21), where the state vector is reset following the resetting law $x(It_f) = x_0,\ I \in \{1, 2, ...\}$, the desired trajectory vector is reset following $\hat{y}_{id}(t) = y_{id}(t - (I-1)t_f),\ (I-1)t_f \leq t < It_f,\ I \in \{1, 2, ...\}$, the cost function is given by (24), and the elements of the vector $\widehat{\Delta}(t)$ are estimated through the iterative GP-UCB algorithm

$$\hat{\Delta}(t) = \Delta_{nominal} + \delta\Delta(t), \tag{37}$$

where $\Delta_{nominal}$ is the nominal value of Δ, $\delta\Delta = [\delta\Delta_1, ..., \delta\Delta_{m^2}]^T$ is computed using the GP-UCB algorithm (32), (33), (34), (35), and (36). Then, the obtained closed-loop impulsive time-dependent dynamic system is well posed. The norm of the error vector $z(t)$ is bounded over the learning iterations, such that, $\|z(t)\| \leq \beta(\|z((I-1)t_f)\|, t) + \gamma(\sup_{(I-1)t_f \leq \tau \leq It_f} \|e_\Delta(\tau)\|),\ (I-1)t_f \leq t \leq It_f$. Furthermore, the parameters vector Δ is estimated, over $[0, It_f]$, with a cumulative regret $R_I = \sum_{j=0}^{I} r_t,\ r_j = J(\hat{\Delta}_j) - J(\Delta)$, which admits the following upper-bound

$$R_I \leq \sqrt{\frac{8I\beta_I\gamma_I}{1 + \log(1 + \sigma^{-2})}},$$

with probability at least $1 - \delta$, where $\delta > 0$ is the confidence parameter.

Proof: The well-posedness of the closed-loop system, and the boundedness of the tracking error z over the learning iterations, are concluded from similar arguments as in the proof of Lemma 3. Next, to provide an upper bound on the regret of the method, we follow (Theorem 3, in [46]), and fix the confidence parameter $\delta > 0$. If the reproducing kernel Hilbert space (RKHS) $\mathcal{H}_\kappa$ defined by the kernel κ is such that $\|J\|_{\mathcal{H}_\kappa} < \infty$, we can choose

$$\beta_I = 2\|J\|_{\mathcal{H}_\kappa} + 300\gamma_I \log^3(It_f/\delta),$$

in which γ_I should depend on kernel κ. For the exponential kernel and $D \subset \mathbb{R}^d$, we have $\gamma_I = O((\log(I))^{d+1})$.

We define $\hat{\Delta}^* \leftarrow \text{argmin}_{\hat{\Delta} \in D}\, J(\hat{\Delta})$, the global minimizer of the objective function. We define the regret at time $t = jt_f$ by $r_j = J(\hat{\Delta}_j) - J(\Delta)$, which is a measure of sub-optimality of the choice of $\hat{\Delta}_j$ according the cost function J. The cumulative regret over $[0, It_f]$ is defined as $R_I = \sum_{i=1}^{i=I} r_i$. If we choose $\hat{\Delta}_j$ according to (35) with the aforementioned parameters, we can write the cumulative regret's upper-bound [46]

$$R_I \leq \sqrt{\frac{8I\beta_I\gamma_I}{1 + \log(1 + \sigma^{-2})}},$$

with probability at least $1 - \delta$. □

Remark 13. Lemma 4 states that the cumulative regret bound is sub-linear in the learning iterations I, which makes the average regret for each learning iteration, defined as the cumulative regret

Prepared using acsauth.cls
DOI: 10.1002/acs

divided by I, inversely proportional to $\sqrt{I}$, and thus decreasing with the increase of the learning iterations. Furthermore, due to Assumption A9, the decrease in the cost function over the learning iterations, implies a decrease of the estimation error over the learning iterations.

Remark 14. Here again, one needs to think about what is the equivalent PE condition, which ensures that the estimation filter keeps looking for the true values of the uncertain parameters, without 'settling'. To understand this, we have to remember that the GP-UCB algorithm is a type of reinforcement algorithms, which are well known to combine exploration and exploitation phases. Indeed, as we explained earlier, here the main idea is to approximate the learning cost function with a GP process, and look for the optimal argument of the cost function based on this approximation. However, to refine the approximation, we saw that we needed to sample the cost function at several sampling points $\hat{\Delta}_j,\ j \in \{1, ..., N, ...\}$. This sampling of the compact set D constitutes the exploration phase of the GP-UCB algorithm. The way the next sampling point is selected is based on solving the optimization problem (35), over the entire compact set D, then the cost function is measured at this sampling point, and its value is added to the vector of measurements $\mathbf{y}_I$, to refine the GP approximation of the cost function, in terms of its mean and variance functions (33), (34). This exploration process, ensures that the entire compact set D is sampled, and thus can be seen as satisfying an equivalent PE condition, in this context of RL algorithms.

Remark 15. In Lemma 4, the controller given by (9), (14), and (21) is assumed to be measurement noise-free, whereas the learning cost function is assumed to be corrupted with measurement noise. First, this assumptions can correspond to the case of applications where the feedback control law is based on precise measurements with negligible measurement noise, and where the learning cost function is computed based on noisy observations. Second, we have explained in Remark 4 that the controller (9), (14), and (21) is robust to bounded additive measurement noise, in which case the boundedness of the error vector, shown in Lemma 4, is still maintained in the presence of bounded measurement noise added to the feedback control signal.

5. TWO-LINK MANIPULATOR EXAMPLE

We consider here a two-link robot manipulator, with the following dynamics

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \ = \ \tau, \tag{38}$$

Prepared using acsauth.cls
DOI: 10.1002/acs

where $q \triangleq [q_1, q_2]^T$ denotes the two joint angles and $\tau \triangleq [\tau_1, \tau_2]^T$ denotes the two joint torques. The matrix $H \in \mathbb{R}^{4\times 4}$ is assumed to be non-singular and its elements are given by

$$\begin{aligned} H_{11} &= m_1\ell_{c_1}^2 + I_1 + m_2[\ell_1^2 + \ell_{c_2}^2 + 2\ell_1\ell_{c_2}\cos(q_2)] + I_2, \\ H_{12} &= m_2\ell_1\ell_{c_2}\cos(q_2) + m_2\ell_{c_2}^2 + I_2, \\ H_{21} &= H_{12}, \\ H_{22} &= m_2\ell_{c_2}^2 + I_2. \end{aligned} \tag{39}$$

The matrix $C(q, \dot{q})$ is given by

$$C(q, \dot{q}) \triangleq \begin{bmatrix} -h\dot{q}_2 & -h\dot{q}_1 - h\dot{q}_2 \\ h\dot{q}_1 & 0 \end{bmatrix},$$

where $h = m_2\ell_1\ell_{c_2}\sin(q_2)$. The vector $G = [G_1, G_2]^T$ is given by

$$\begin{aligned} G_1 &= m_1\ell_{c_1}g\cos(q_1) + m_2g[\ell_2\cos(q_1+q_2) + \ell_1\cos(q_1)], \\ G_2 &= m_2\ell_{c_2}g\cos(q_1+q_2), \end{aligned} \tag{40}$$

where, $\ell_1,\ \ell_2$ are the lengths of the first and second link, respectively, $\ell_{c_1},\ \ell_{c_2}$ are the distances between the rotation center and the center of mass of the first and second link respectively. $m_1,\ m_2$ are the masses of the first and second link, respectively, I_1 is the moment of inertia of the first link and I_2 the moment of inertia of the second link, respectively, and g denotes the Earth gravitational constant.

In our simulations, we assume that the parameters take the following values: $I_2 = \frac{5.5}{12}\ kg\cdot m^2$, $m_1 = 10.5\ kg$, $m_2 = 5.5\ kg$, $\ell_1 = 1.1\ m$, $\ell_2 = 1.1\ m$, $\ell_{c_1} = 0.5\ m$, $\ell_{c_2} = 0.5\ m$, $I_1 = \frac{11}{12}\ kg\cdot m^2$, $g = 9.8\ m/s^2$. The system dynamics (38) can be rewritten as

$$\ddot{q} = H^{-1}(q)\tau - H^{-1}(q)\left[C(q, \dot{q})\dot{q} + G(q)\right]. \tag{41}$$

Thus, the nominal controller is given by

$$\tau_n = \left[C(q, \dot{q})\dot{q} + G(q)\right] + H(q)\left[\ddot{q}_d - K_d(\dot{q} - \dot{q}_d) - K_p(q - q_d)\right], \tag{42}$$

where $q_d = [q_{1d}, q_{2d}]^T$, denotes the desired trajectory and the diagonal gain matrices $K_p > 0$, $K_d > 0$, are chosen such that the linear error dynamics (as in (11)) are asymptotically stable. We choose as output references the $5th$ order polynomials $q_{1ref}(t) = q_{2ref}(t) = \sum_{i=0}^{5} a_i(t/t_f)^i$, where the a_i's have been computed to satisfy the boundary constraints $q_{iref}(0) = 0, q_{iref}(t_f) = q_f, \dot{q}_{iref}(0) = \dot{q}_{iref}(t_f) = 0, \ddot{q}_{iref}(0) = \ddot{q}_{iref}(t_f) = 0,\ i = 1, 2,$ with $t_f = 2\ sec,\ q_f = 1.5\ rad.$ In these tests, we assume that the nonlinear model (38) is uncertain. In particular, we assume that there exist additive uncertainties in the model (41), i.e.,

$$\ddot{q} = H^{-1}(q)\tau - H^{-1}(q)\left[C(q, \dot{q})\dot{q} + G(q)\right] - E\ G(q), \tag{43}$$

DOI: 10.1002/acs

where E is a matrix of constant uncertain parameters. Following (21), the robust-part of the control writes as

$$\tau_r = -H(\tilde{B}^T P z \|G\|^2 - \hat{E}\, G(q)), \tag{44}$$

where

$$\tilde{B}^T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

P is solution of the Lyapunov equation (13), with

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -K_p^1 & -K_d^1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -K_p^2 & -K_d^2 \end{bmatrix},$$

$z = [q_1 - q_{1d}, \dot{q}_1 - \dot{q}_{1d}, q_2 - q_{2d}, \dot{q}_2 - \dot{q}_{2d}]^T$, and $\hat{E}$ is the matrix of the parameters' estimates. Eventually, the final feedback controller writes as

$$\tau = \tau_n + \tau_r. \tag{45}$$

We consider the challenging case where the uncertain parameters are linearly dependent. In this case the uncertainties' 'effect' is not observable from the measured output. Indeed, in the case where the uncertainties enter the model in a linearly dependent function, e.g. when the matrix Δ has only one non-zero line, some of the classical available modular model-based adaptive controllers, for instance X-swapping controllers, cannot be used to estimate all the uncertain parameters simultaneously. For example, it has been shown in [34], that the model-based gradient descent filters failed to estimate simultaneously multiple parameters in the case of the electromagnetic actuators example. For instance, in comparison with the MES-based indirect adaptive controller of [22], the modular approach does not rely on the parameters mutual exhaustive assumption, i.e., each element of the control vector needs to be linearly dependent on at least one element of the uncertainties vector. More specifically, we consider here the following case: $\Delta(1,1) = 1.5,\ \Delta(1,2) = 1$, and $\Delta(2,i) = 0,\ i = 1,2$. In this case, the uncertainties' effect on the acceleration $\ddot{q}_1$ cannot be differentiated, and thus the application of the model-based X-swapping method to estimate the actual values of both uncertainties at the same time is challenging. Similarly, the method of [22], cannot be readily applied because the second control τ_2 is not linearly depend on the uncertainties, which only affects τ_1. However, we show next that, by using the modular ISS-based controller, we manage to estimate the actual values of the uncertainties simultaneously and improve the tracking performance. Moreover, to make the simulations more realistic, we introduce uniformly distributed additive noises on the angular measurements with a maximal noise excursion of $0.1\ rad$.

DOI: 10.1002/acs

5.1. Iterative MES-based uncertainties estimation

The estimates of the two parameters $\widehat{\Delta}_i$ $(i = 1, 2)$ are computed using a discrete version of (25), given by

$$\begin{aligned} x_i(k+1) &= x_i(k) + a_i t_f \sin(\omega_i t_f k + \frac{\pi}{2}) J(\widehat{\Delta}), \\ \widehat{\Delta}_i(k+1) &= x_i(k+1) + a_i \sin(\omega_i t_f k - \frac{\pi}{2}), \quad i = 1, 2 \end{aligned} \tag{46}$$

where, $k \in \mathbb{N}$ denotes the iteration index, $x_i(0) = \widehat{\Delta}_i(0) = 0$. We choose the following learning cost function

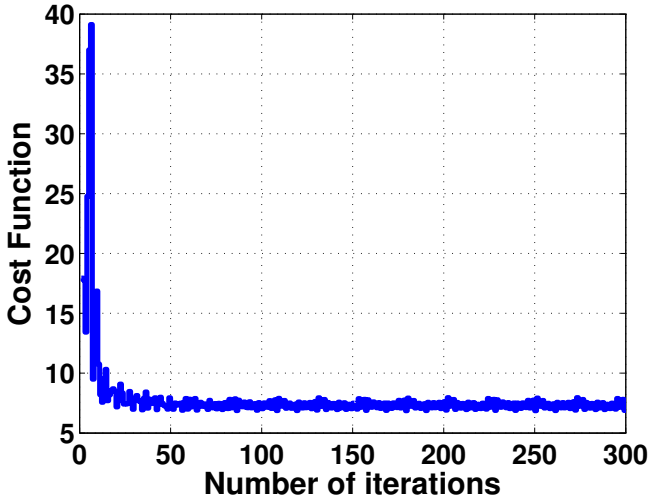
$$J(\widehat{\Delta}) = \int_0^{t_f} (q(\widehat{\Delta}) - q_d(t))^T Q_1 (q(\widehat{\Delta}) - q_d(t)) \mathrm{d}t + \int_0^{t_f} (\dot{q}(\widehat{\Delta}) - \dot{q}_d(t))^T Q_2 (\dot{q}(\widehat{\Delta}) - \dot{q}_d(t)) \mathrm{d}t, \tag{47}$$

where $Q_1 > 0$ and $Q_2 > 0$ denote the weight matrices. We implement the learning algorithm with the parameters: $a_1 = 0.03$, $a_2 = 0.02$, $\omega_1 = 10\ rad/sec$, $\omega_2 = 15\ rad/sec$, $Q_1 = \begin{bmatrix} 500 & 0 \\ 0 & 500 \end{bmatrix}$, $Q_2 = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$. The obtained learning cost function is displayed on Figure 1(a), where we see that the performance improves over the learning iterations. The corresponding parameters estimation profiles are reported in Figures 1(b), and 1(c), which show a quick convergence, with less than 20 iterations, of the first estimates $\hat{\Delta}_1$ to a neighborhood of the actual value. The convergence of the second estimates $\hat{\Delta}_2$ is slower, with 100 iterations, which is expected from the MES algorithms when multiple parameters are estimated at the same time. One has to underline here, however, that the MES setteles at the optimal values of the parameters, but with some residual oscillations around the optimal values. This is a well know property of the dither-based MES algorithm (46), which is also amplified by the measurements noise explicitly introduced in the simulations to make the tests more realistic. This last point motivates the use of the GP-UCB algorithms which are designed specifically for stochastic systems, and will lead to less noisy estimation, as we will see in the next section. Finally, The tracking performance is shown in Figures 2(a), 2(b), where we can see that, after learning the actual values of the uncertainties, the tracking of the desired trajectories is recovered. We only show the first angular trajectories here, because the uncertainties affect directly only the acceleration $\ddot{q}_1$, and their effect on the tracking for the second angular variable is negligible.

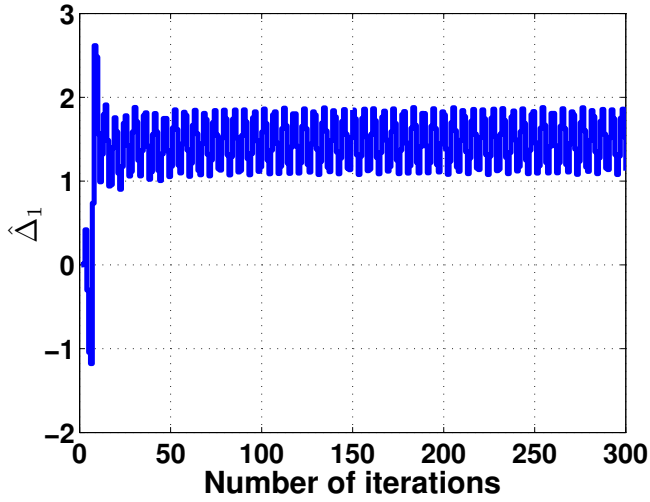
5.2. Iterative GP-UCB-based uncertainties estimation

To show that the modular ISS-based controller is independent of the choice of the learning algorithm, we apply the GP-UCB learning algorithm-based estimator to the same two-links manipulator example. We apply the Algorithm 2, with the following parameters: $\sigma = 0.1$, $l = 0.2$, and $\delta = 0.05$. We test the GP-UCB algorithm under the uncertainties conditions stated above. The obtained learning cost, and estimated parameters are reported in Figures 3(a), 3(b), 3(c). We

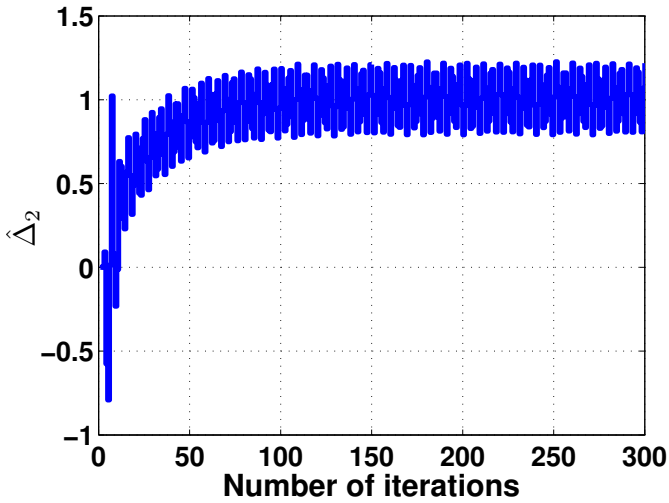
Prepared using acsauth.cls
DOI: 10.1002/acs



(a) Cost function over the learning iterations (MES)



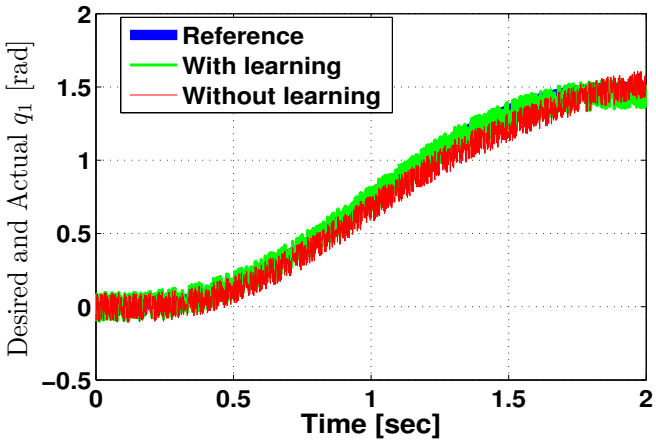
(b) Estimate of Δ_1 over the learning iterations (MES)



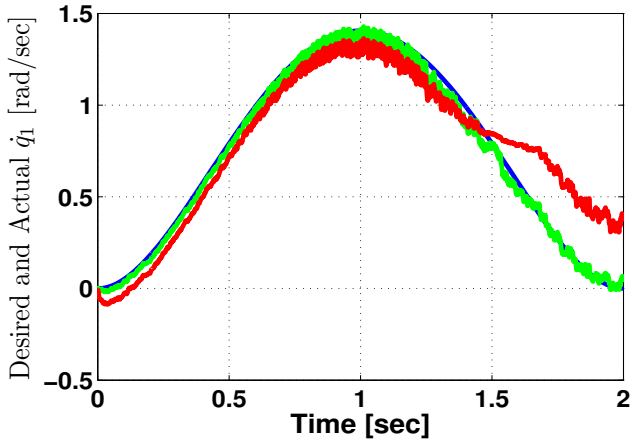
(c) Estimate of Δ_2 over the learning iterations (MES)

Figure 1: Cost function and uncertainties estimates- MES algorithm

DOI: 10.1002/acs



(a) Obtained vs. desired first angular trajectory (MES)



(b) Obtained vs. desired first angular velocity trajectory (MES)

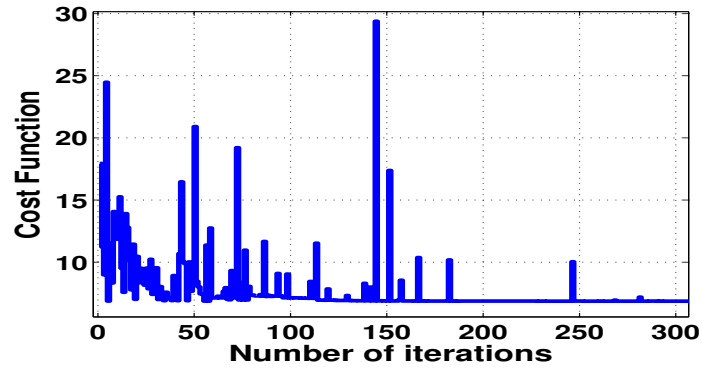
Figure 2: Obtained vs. desired angular trajectories (MES)

can see on these figures that the uncertainties are well estimated. One could argue that they are better estimated with the GP-UCB than with MES algorithm, because there is no permanent dither signal, which leads to permanent oscillations in the MES-based learning, which is also amplified by the measurement noise in the case of the MES algorithm, whereas the GP-UCB is designed for stochastic dynamics and handles measurement noise better. The tracking performance improved as well due to the precise estimation of the parameters, as seen in Figure 4.

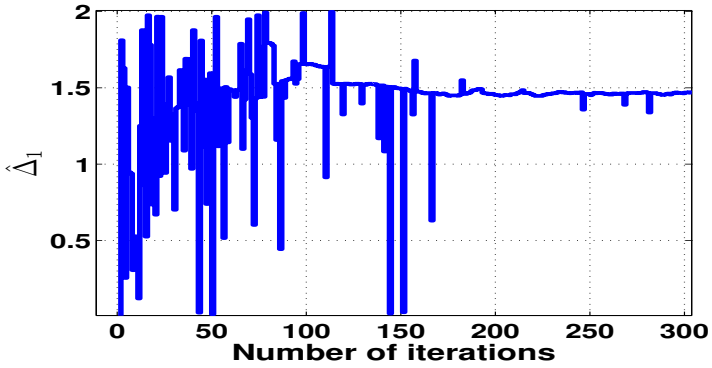
6. CONCLUSION

We have studied the problem of iterative adaptive control for nonlinear systems which are affine in the control. For this class of systems, we have proposed the following controller: we use a

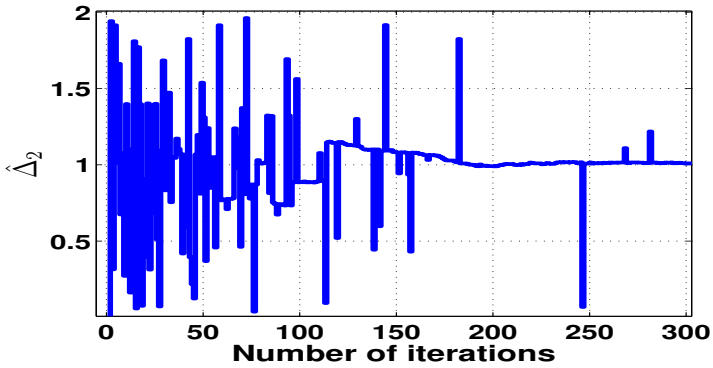
Prepared using acsauth.cls
DOI: 10.1002/acs



(a) Cost function over the learning iterations (GP-UCB)



(b) Estimate of Δ_1 over the learning iterations (GP-UCB)

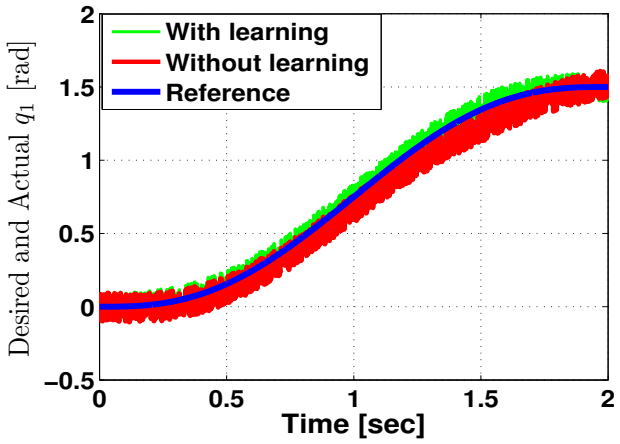


(c) Estimate of Δ_2 over the learning iterations (GP-UCB)

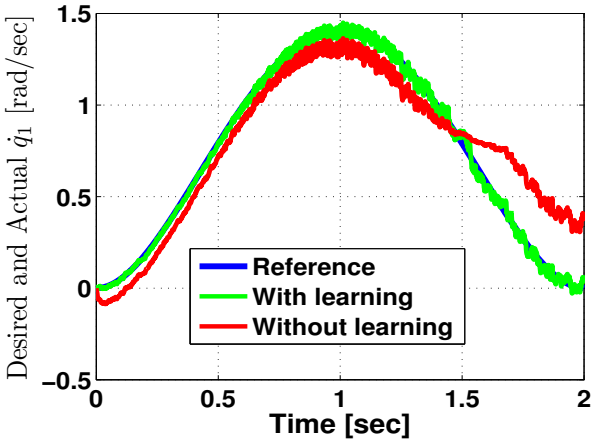
Figure 3: Cost function and uncertainties estimates- (GP-UCB) algorithm

modular approach, where we first design a robust nonlinear controller, based on the model (assuming knowledge of the uncertain parameters), and then complement this controller with an iterative estimation module to estimate the actual values of the uncertain parameters. The novelty is that the proposed estimation module is based on data-driven learning algorithms. Indeed, we propose to

Prepared using acsauth.cls
DOI: 10.1002/acs



(a) Obtained vs. desired first angular trajectory (GP-UCB)



(b) Obtained vs. desired first angular velocity trajectory (GPU-CB)

Figure 4: Obtained vs. desired angular trajectories (GPU-CB)

use two learning algorithms, namely, a multi-parametric extremum seeking algorithm, and a GP-UCB algorithm, to learn in real-time the uncertainties of the model. We call the learning approach 'data-driven' because it only requires to measure an output signal from the system and compare it to a desired reference signal (independent of the model), to iteratively learn the best estimate of the model uncertainties. We have guaranteed the stability (while learning) of the proposed approach, by ensuring that the model-based robust controller leads to an ISS result. Which in turn guarantees boundedness of the states of the closed-loop system, even during the learning phase. The ISS result together with a convergent learning algorithm eventually leads to a bounded output tracking error, which decreases with the decrease of the estimation error.

We believe that one of the main advantages of the proposed controller, compared to the existing model-based adaptive controllers, is that we can learn (estimate) multiple uncertainties at the same

Prepared using acsauth.cls
DOI: 10.1002/acs

time even if they appear in the model equation in a challenging way, e.g., linearly dependent uncertainties affecting only one output, or uncertainties appearing in a nonlinear term of the model, which are well-known limitations of the model-based estimation approaches. Another advantage of the proposed approach is that due to its modular design, one could easily change the learning algorithm without having to change the model-based part of the controller. Indeed, as long as the first part of the controller, i.e., the model-based part, has been designed with a proper ISS property, one can 'plug into it' any convergent learning data-driven algorithm, as demonstrated here by using two different learning approaches. We reported here results about using MES and GP-UCB in this setting of iterative modular adaptive control, in the case of constant parametric uncertainties. In future work, we will focus on the case of time-varying uncertainties, as well as experimental validation of the proposed controllers.

REFERENCES

1. P. Ioannou, J. Sun, *Robust Adaptive Control*. Dover Publications, 2012.
2. I. D. Landau, T.-B. Airimiţoaie, A. Castellanos-Silva, A. Constantinescu. Adaptive and Robust Active Vibration Control: Methodology and Tests. Advances in Industrial Control. Springer–Verlag, 2017.
3. M. Benosman, *Learning-Based Adaptive Control: An Extremum Seeking Approach - Theory and Applications*. Cambridge, MA: Butterworth-Heinemann, 2016.
4. M. Benosman, A.-M. Farahmand, M. Xia, Learning-based modular indirect adaptive control for a class of nonlinear systems. IEEE, American Control Conference, pp. 733–738, 2016.
5. M. Benosman, A.-M. Farahmand, Bayesian Optimization-based Modular Indirect Adaptive Control for a Class of Nonlinear Systems. In: 12th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing. Eindhoven, The Netherlands, pp. 253–258, 2016.
6. M. Krstic, I. Kanellakopoulos, P. Kokotovic, *Nonlinear and Adpative Control Design*. New York: Wiley, 1995.
7. G. Tao, Mutlivariable adaptive control: A survey. Automatica 50, pp. 2737–2764, 2014.
8. C. Wang, D. J. Hill, Deterministic Learning Theory for Identification, Recognition, and Control. CRC Press, 2010.
9. Z. Hou, S. Jin, A Novel Data-Driven Control Approach for a Class of Discrete-Time Nonlinear Systems. IEEE, Transactions on Control Systems Technology 19 (6), pp. 1549–1558, 2011.
10. J. T. Spooner, M. Maggiore, R. Ordonez, K. M. Passino, Stable adaptive control and estimation for nonlinear systems. Wiley– Interscience, New York, 2002.
11. C. Zhang, R. Ordonez, Extremum–Seeking Control and Applications: A Numerical Optimization–Based Approach. Springer, New York, 2012.
12. F. L. Lewis, D. Vrabie, K. G. Vamvoudakis, Reinforcement learning and feedback control:using natural decision methods to design optimal adaptive controllers. IEEE Control Systems Magazine, pp. 76–105, 2012.
13. C. Wang, D. J. Hill, S. S. Ge, G. Chen, An ISS-modular approach for adaptive neural control of pure-feedback systems, *Automatica*, vol. 42, no. 5, pp. 723–731, 2006.
14. M. Guay, T. Zhang, Adaptive extemum seeking control of nonlinear dynamic systems with parametric uncertainties. Automatica (39), pp. 1283–1293, 2003.
15. D. Dehaan, M. Guay, Extremum-seeking control of state-constrained nonlinear systems. Automatica 41 (9), pp. 1567–1574, 2005.
16. M. Guay, D. Dochain, M. Perrier, N. Hudon. Flatness-based Extremum-seeking Control Over Periodic Orbits. IEEE Transactions on Automatic Control, 52 (10), 2005–2012, 2007.

DOI: 10.1002/acs

17. D. Vrabie, K. Vamvoudakis, F. L. Lewis, Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles. IET Digital library, 2013.
18. L. Koszaka, R. Rudek, I. Pozniak-Koszalka, An idea of using reinforcement learning in adaptive control systems, in Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, pp. 190–190, 2006.
19. J.Y. Lee, J. B. Park, Y. H. Choi, Intergral Q-learning and explorized iteration for adaptive optimal control of continuous-time linear systems. Automatica 48, pp. 2850–2859, 2012.
20. P. Frihauf, M. Krstic, T. Basar, Finite-horizon LQ control for unknown discrete-time linear systems via extremum seeking. European Journal of Control 19 (5), pp. 399–407, 2013.
21. P. Haghi, K. Ariyur, On the extremum seeking of model reference adaptive control in higher-dimensional systems, in *American Control Conference*, pp. 1176–1181, 2011.
22. ——, Adpative feedback linearization of nonlinear MIMO systems using ES-MRAC, in *American Control Conference*, pp. 1828–1833, 2013.
23. Z.-P. Jiang, Y. Jiang, Robust adaptive dynamic programming for linear and nonlinear systems: An overview. European Journal of Control 19 (5), pp. 417–425, 2013.
24. M. Benosman, G. Atinc, Multi-parametric extremum seeking-based learning control for electromagnetic actuators, in *American Control Conference*, pp. 1914–1919, 2013.
25. ——, Nonlinear learning-based adaptive control for electromagnetic actuators, in *European Control Conference*, pp. 2904–2909, 2013.
26. G. Atinc, M. Benosman, Nonlinear learning-based adaptive control for electromagnetic actuators with proof of stability, in *IEEE, Conference on Decision and Control*, pp. 1277–1282, 2013.
27. H. Modares, F. Lewis, T. Yucelen, G. Chowdhary, Adaptive optimal control of partially-unknown constrained-input systems using policy iteration with experience replay. In: AIAA Guidance, Navigation, and Control Conference. Boston, Massachusetts, pp. 2013-4519, 2013.
28. M. Benosman, S. Di Cairano, A. Weiss, Extremum seeking–based iterative learning linear MPC. In: IEEE Multi–conference on Systems and Control. pp. 1849–1854, 2014.
29. M. Benosman, Multi-parametric extremum seeking-based auto-tuning for robust input-output linearization control. In: IEEE, Conference on Decision and Control. Los Angeles, CA, pp. 2685–2690, 2014.
30. M. Benosman, Learning-based adaptive control for nonlinear systems, in *IEEE European Control Conference*, pp. 920–925, 2014.
31. ——, Extremum-seeking based adaptive control for nonlinear systems, in *IFAC World Congress*, pp. 401–406, 2014.
32. M. Xia, M. Benosman, Extremum seeking-based indirect adaptive control for nonlinear systems with time-varying uncertainties, in *European Control Conference*, pp. 2780–2785, 2015.
33. M. Benosman, G. Atinc, Nonlinear backstepping learning-based adaptive control of electromagnetic actuators, *International Journal of Control*, vol. 88, no. 3, pp. 517–530, 2014.
34. ——, Non-linear adaptive control for electromagnetic actuators, *IET Control Theory and Applications*, vol. 9, no. 2, pp. 258–269, 2015.
35. K. B. Ariyur, M. Krstic, *Real Time Optimization by Extremum Seeking Control*. New York, NY, USA: John Wiley & Sons, Inc., 2003.
36. A. Scheinker, M. Krstic, Model-Free Stabilization by Extremum Seeking. Springer, 2016.
37. C. Zhang, R. Ordonez. Extremum–Seeking Control and Applications: A Numerical Optimization–Based Approach. Springer, New York, 2012.
38. M., Guay. A Perturbation-Based Proportional Integral Extremum-Seeking Control Approach. IEEE Transactions on Automatic Control, 61 (11), 3370–3381, 2016.
39. M. Guay, D. Dochain. A Minmax Extremum-Seeking Controller Design Technique. IEEE Transactions on Automatic Control, 59 (7), 1874–1886, 2015.

DOI: 10.1002/acs

40. M. Guay, S. Dhaliwal, D. Dochain. A time-varying extremum-seeking control approach. In: IEEE, American Control Conference. pp. 2643–2648, 2013.
41. M. Guay, D. Dochain. A time-varying extremum-seeking control approach. Automatica (51), 356–363, 2015.
42. M. Guay, D. Dochain. A multi-objective extremum-seeking controller design technique. International Journal of Control, 88(1), 38–53, 2015.
43. B. Gruenwald, T. Yucelen, On transient performance improvement of adaptive control architectures. Int. Journal of Control 88 (11), pp. 2305–2315, 2015.
44. A. Subbaraman, M. Benosman, Extremum Seeking-based Iterative Learning Model Predictive Control (ESILC-MPC). 12th IFAC Workshop on Adaptation and Learning in Control and Signal, pp. 193–198, 2016.
45. M. Rotea, Analysis of multivariable extremum seeking algorithms, in *American Control Conference. Proceedings of the 2000*, vol. 1, no. 6, pp. 433–437, 2000.
46. N. Srinivas, A. Krause, S. M. Kakade, M. Seeger, Gaussian process optimization in the bandit setting: No regret and experimental design, in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 1015–1022, 2010.
47. R. S. Sutton, A. G. Barto, *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press, 1998.
48. M. Benosman, Multi-Parametric Extremum Seeking-based Auto-Tuning for Robust Input-Output Linearization Control, *International Journal of Robust and Nonlinear Control*, vol. 26, no. 18, pp. 4035–4055, 2016.
49. H. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.
50. M. Malisoff, F. Mazenc, Further remarks on strict input-to-state stable lyapunov functions for time-varying systems, *Automatica*, vol. 41, no. 11, pp. 1973 – 1978, 2005.
51. M. Benosman, F. Liao, K.-Y. Lum, J. L. Wang, Nonlinear control allocation for non-minimum phase systems, *Control Systems Technology, IEEE Transactions on*, vol. 17, no. 2, pp. 394–404, 2009.
52. H. Elmali, N. Olgac, Robust output tracking control of nonlinear MIMO systems via sliding mode technique, *Automatica*, vol. 28, no. 1, pp. 145–151, 1992.
53. W. M. Haddad, V. Chellaboind, S. G. Nersesov, *Impulsive and Hybrid Dynamical Systems: Stability, Dissipativity, and Control*. Princeton University Press, Princeton, 2006.
54. Y. Tan, D. Nesic, I. Mareels, On non-local stability properties of extremum seeking control, *Automatica*, no. 42, pp. 889–903, 2006.
55. Y. Tan, D. Nesic, I. Mareels, On the dither choice in extremum seeking control, *Automatica*, no. 44, pp. 1446–1450, 2008.
56. L. Wang, S. Chen, K. Ma, On stability and application of extremum seeking control without steady-state oscillation, *Automatica*, no. 68, pp. 18–26, 2016.
57. N. Srinivas, A. Krause, S. M. Kakade, M. Seeger, Information-theoretic regret bounds for gaussian process optimization in the bandit setting, *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.
58. E. Brochu, V. M. Cora, N. de Freitas, A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, *arXiv:1012.2599*, 2010.
59. P. Auer, N. Cesa-Bianchi, P. Fischer, Finite-time analysis of the multiarmed bandit problem, *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.
60. S. Bubeck, R. Munos, G. Stoltz, Cs. Szepesvári, X-armed bandits, *Journal of Machine Learning Research (JMLR)*, vol. 12, pp. 1655–1695, 2011.
61. S. Bubeck, N. Cesa-Bianchi, Regret analysis of stochastic and nonstochastic multi-armed bandit problems, *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.
62. R. Munos, From bandits to Monte-Carlo Tree Search: The optimistic principle applied to optimization and planning, *Foundations and Trends in Machine Learning*, vol. 7(1), pp. 1–130, 2014.
63. C. E. Rasmussen, C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.
64. D. A. Bristow, M. Tharayil, and A. G. Alleyne. A survey of iterative learning control. *Control Systems Magazine*, 26(3):96–114, 2006.

DOI: 10.1002/acs

65. J.-X. Xu and R. Yan. On initial conditions in iterative learning control. *IEEE, Transactions on Automatic Control*, 50(9):96–114, 2005. 1349–1354.
66. C. Zhang and R. Ordez. *Extremum-Seeking Control and Applications*. Springer-Verlag, 2012.

DOI: 10.1002/acs